

---

Analysis and Design of Security Mechanisms  
in the Context of Advanced Persistent Threats  
Against Critical Infrastructures

---



TESIS DOCTORAL

Juan Enrique Rubio Cortés

Programa de Doctorado en Tecnologías Informáticas  
Departamento de Lenguajes y Ciencias de la Computación  
Escuela Técnica Superior de Ingeniería Informática  
Universidad de Málaga

Febrero de 2022



UNIVERSIDAD  
DE MÁLAGA

AUTOR: Juan Enrique Rubio Cortés

 <https://orcid.org/0000-0002-7338-9390>

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:

<http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Cualquier parte de esta obra se puede reproducir sin autorización pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): [riuma.uma.es](http://riuma.uma.es)





**Analysis and Design of Security Mechanisms  
in the Context of Advanced Persistent Threats  
Against Critical Infrastructures**

por

Juan Enrique Rubio Cortés

Memoria presentada para optar al título de  
Doctor por la Universidad de Málaga

Directores:

Fco. Javier Lopez Muñoz

Catedrático de Universidad

María Cristina Alcaraz Tello

Profesora Titular de Universidad

Febrero de 2022



UNIVERSIDAD  
DE MÁLAGA



## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D./Dña JUAN ENRIQUE RUBIO CORTÉS

Estudiante del programa de doctorado TECNOLOGÍAS INFORMÁTICAS de la Universidad de Málaga, autor/a de la tesis, presentada para la obtención del título de doctor por la Universidad de Málaga, titulada: ANALYSIS AND DESIGN OF SECURITY MECHANISMS IN THE CONTEXT OF ADVANCED PERSISTENT THREATS AGAINST CRITICAL INFRASTRUCTURES

Realizada bajo la tutorización de JAVIER LÓPEZ MUÑOZ y dirección de JAVIER LÓPEZ MUÑOZ Y CRISTINA ALCARAZ TELLO (si tuviera varios directores deberá hacer constar el nombre de todos)

DECLARO QUE:

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo.

Igualmente asumo, ante a la Universidad de Málaga y ante cualquier otra instancia, la responsabilidad que pudiera derivarse en caso de plagio de contenidos en la tesis presentada, conforme al ordenamiento jurídico vigente.

En Málaga, a 27 de FEBRERO de 2022

-	
Fdo.: JUAN ENRIQUE RUBIO CORTÉS Doctorando/a	Fdo.: JAVIER LÓPEZ MUÑOZ Tutor/a
LOPEZ MUÑOZ FRANCISCO	ALCARAZ TELLO MARIA CRISTINA -
Fdo.: JAVIER LÓPEZ MUÑOZ, CRISTINA ALCARAZ TELLO Director/es de tesis	





UNIVERSIDAD  
DE MÁLAGA



D. Fco. Javier López Muñoz, Catedrático de Universidad del área de Ingeniería Telemática del Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, y Dña. María Cristina Alcaraz Tello, Profesora Titular de Universidad del área de Ingeniería Telemática del Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga,

**CERTIFICAN QUE:**

Don Juan Enrique Rubio Cortés, Ingeniero en Informática, ha realizado en el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, bajo nuestra dirección, el trabajo de investigación correspondiente a su Tesis Doctoral, titulada:

**Analysis and Design of Security Mechanisms  
in the Context of Advanced Persistent Threats  
Against Critical Infrastructures**

Revisado el presente trabajo, estimamos que puede ser presentado al tribunal que ha de juzgarlo, y autorizamos la presentación de esta Tesis Doctoral en la Universidad de Málaga.

Málaga, a 27 de febrero de 2022

**ALCARAZ  
TELLO MARIA  
CRISTINA -**

**LOPEZ  
MUÑOZ  
FRANCISCO  
JAVIER -**

Fdo: Dña. María Cristina Alcaraz Tello  
Profesora Titular de Universidad  
Área de Ingeniería Telemática

Fdo: D. Fco. Javier López Muñoz  
Catedrático de Universidad  
Área de Ingeniería Telemática



UNIVERSIDAD  
DE MÁLAGA

*For all those who accompanied me on  
this journey, especially Chelo, David  
and my parents.*



UNIVERSIDAD  
DE MÁLAGA

# Acknowledgements

First and foremost, I would like to express my sincere gratitude to my thesis directors. On the one hand, Javier has become a true reference of friendship and leadership to me. Thank you for believing in me from the very first day and letting me be part of your incredible team. On the other hand, Cristina has been more than a supervisor and a partner, an absolute model of methodology and research success. Thank you for your wisdom, your unlimited support and your solid friendship. I do not have enough words to thank you both for giving shape and sense to my professional career.

This Ph.D. thesis has been quite a challenge me, and I would not have fulfilled it without the help of many other people that made it a bit easier. Starting with the professional side, I must devote the first word to my memories with the awesome guys at NICS lab and their everlasting imprint in my life. They really contributed to make me grow not only as a researcher, but as an adult person. Just to name a few of the incredible people I worked with, I thank Rubén, Rodrigo and Jesús, for their inspiring advice in pretty much every aspect of life, including economics, women and electronics. Not to mention Davide and Martin, my eternal companions on this Ph.D. journey and unconditional allies in bureaucratic annoyances. Or Isaac, for introducing me to the cybersecurity area. I would like to thank David and Lorena, for warmly welcoming me to the group. Also thanks to Carmen and Anto, for their gossip conversations and jokes at lunch. Gerardo and Ana, thank you for your knowledge and guidance. Onieva, thank you for your wisdom on and off the football pitch. Thanks to the junior team, for bringing so many fun moments to the group. Of course, my gratitude to Noelia, for keeping it all standing behind the scenes. And thanks to the latest mates I had the chance to meet, Dani and Sergio, whom I wish all the best in their Ph.D. adventure. You all rock, the NICS legacy is assured with you guys.

After all, I would honestly spend one more thesis mentioning all those coworkers that influenced me throughout these incredible years of doctorate, at the lab and outside. I do not want to forget those professors who kindly helped me during the process of teaching, the own pupils (who ended up turning into friends) and the amazing colleagues I stumbled upon in conferences and meetings of all kinds. In particular, I would like to thank the associates of SealedGRID project, for their support and affection. And I am also deeply grateful to Mark Manulis, for hosting me at the University of Surrey, where I could greatly enhance my work.



---

From the personal side, I thank you, Chelo, for existing and arriving at my life. For holding me at the hardest moments and making me believe it was possible. For walking the wire and dancing together, giving a meaning to this path. You are my fate and I love you to the moon and back. Thank you, mum and dad, for your unconditional love and always being there for everything, for being home and a shoulder to cry. And thanks David for being my soulmate and showing me that the big brother doesn't always lead the way. I must also thank all my beloved friends and relatives that supported me along the way and forgave me for not attending many social events due to this thesis. Thank you for believing in me and being part of my life. I love you all.

Lastly, I am also thankful to some institutions and projects for their funding and support. This thesis has been primarily funded by the Spanish Ministry of Education under the FPU program (FPU15/03213). Special thanks goes to the SADCIP (RTC- 2016-4847-8) and SealedGRID (H2020-MSCA-RISE-2017) projects, because they gave me the opportunity to actively collaborate with other institutions and improve the quality of my research.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Industry 4.0 and Current Issues . . . . .	1
1.1.1	International Initiatives and Consortia . . . . .	3
1.1.2	Review of Innovations and Enabling Technologies . . . . .	6
1.1.3	Hardware and Software in the Industrial Internet of Things . . . . .	11
1.2	Overview of Cybersecurity Challenges in the Industry 4.0 . . . . .	14
1.3	Goals and Contributions of the Thesis . . . . .	18
1.3.1	Thesis Outline . . . . .	20
1.4	Publications and Funding . . . . .	21
<b>2</b>	<b>Cybersecurity in Industrial Ecosystems</b>	<b>25</b>
2.1	Traditional Threats in IS and ICS . . . . .	25
2.2	Landscape of Cybersecurity Threats of Industry 4.0 Enabling Technologies . . . . .	27
2.2.1	Industrial Internet of Things Threats . . . . .	27
2.2.2	Cloud Computing Threats . . . . .	28
2.2.3	Big Data Threats . . . . .	30
2.2.4	Blockchain Threats . . . . .	31
2.2.5	Virtualization Threats . . . . .	32
2.3	Cybersecurity Threats in Industry 4.0 Innovative Services . . . . .	34
2.4	Understanding Advanced Persistent Threats in Industry 4.0 . . . . .	36
2.4.1	Review of Reported Cases . . . . .	38
2.4.2	APTs Phases and Notations . . . . .	41
<b>3</b>	<b>Detection of APTs in Industry 4.0: State of the Art</b>	<b>45</b>
3.1	Classification of Intrusion Detection Systems . . . . .	45
3.2	Academic Research . . . . .	49
3.2.1	Analysis: Detection Mechanisms . . . . .	50
3.2.2	Analysis: Detection Coverage . . . . .	52
3.2.3	Analysis: Protocols Analyzed . . . . .	52
3.3	Industrial IDS Products . . . . .	53

3.3.1	Zone Separation . . . . .	54
3.3.2	Secure Configuration . . . . .	54
3.3.3	Signature-based Solutions . . . . .	55
3.3.4	Context-based Mechanisms . . . . .	56
3.3.5	Honeypot-based Techniques . . . . .	56
3.3.6	Anomaly-based Solutions . . . . .	57
3.4	Current Industry 4.0 Detection and Traceability Solutions . . . . .	58
3.5	Detection and Security Requirements for the Industry 4.0 . . . . .	60
<b>4</b>	<b>Detection and Traceability Solutions based on Distributed Correlation</b>	<b>63</b>
4.1	Modelling Industry 4.0 Networks Using Graph Theory . . . . .	63
4.1.1	Structural Controllability . . . . .	63
4.1.2	Topology Generators . . . . .	67
4.1.3	Representation of APT Attacks and Detection Probabilities . . . . .	70
4.2	APT Traceability Framework for the Industry 4.0 . . . . .	73
4.2.1	Network Architecture and Information Acquisition . . . . .	74
4.2.2	Inputs and Outputs of the Traceability Solution . . . . .	78
4.3	Distributed Correlation Models . . . . .	79
4.3.1	Consensus Model . . . . .	80
4.3.2	Opinion Dynamics . . . . .	82
4.3.3	Clustering Algorithms . . . . .	86
4.3.4	Discussion . . . . .	88
4.4	Adapting the Opinion Dynamics Model . . . . .	89
4.5	Adapting the Clustering Models . . . . .	91
4.6	Common Traceability Features . . . . .	94
4.7	Comparison of Models . . . . .	99
<b>5</b>	<b>Protecting Industry 4.0 Scenarios against APTs and Use Cases</b>	<b>105</b>
5.1	Ensuring the Survivability of the Network . . . . .	105
5.1.1	Threat Model based on Topological Changes . . . . .	106
5.1.2	APT Response using Opinion Dynamics . . . . .	108
5.1.3	Experimental Analysis . . . . .	112
5.2	QoS-Aware Routing protocol based on Opinion Dynamics . . . . .	116
5.2.1	Quality of Service Indicators for Routing Protocols . . . . .	116
5.2.2	QoS-Aware Routing . . . . .	118
5.2.3	Simulation and Evaluation . . . . .	120
5.3	Applicability of Opinion Dynamics in the Industrial Internet of Things . . . . .	126
5.3.1	Data retrieval . . . . .	129
5.3.2	Correlation of Anomalies . . . . .	132



5.3.3	Case Study . . . . .	135
5.4	Applicability of Opinion Dynamics in the Smart Grid . . . . .	138
5.4.1	Resilient Architecture for Fault Detection . . . . .	138
5.4.2	Context-awareness Manager for Authorization Policies . . . . .	152
5.4.3	Readjustment of Intelligent Authorization Policies . . . . .	163
<b>6</b>	<b>Verification and Validation</b>	<b>169</b>
6.1	Clustering-based Detection Approach through Induction . . . . .	169
6.2	Opinion Dynamics-based Traceability through Induction . . . . .	171
6.3	Opinion Dynamics-based Survivability through Induction . . . . .	173
6.4	Opinion Dynamics-based Response through Game Theory . . . . .	174
6.4.1	Proposed Network Architecture . . . . .	176
6.4.2	Rules and Scoring System . . . . .	178
6.4.3	Attack and Defense Models . . . . .	181
6.4.4	Simulations and Results . . . . .	187
6.4.5	Theoretical Demonstration . . . . .	191
6.5	Validation in a Testbed . . . . .	192
6.5.1	I4Testbed: An Industry 4.0 Testbed . . . . .	193
6.5.2	Implementation of the Virtual Agents . . . . .	194
6.5.3	APT Test Case with I4Testbed . . . . .	196
<b>7</b>	<b>Conclusions</b>	<b>201</b>
7.1	Contributions . . . . .	201
7.2	Challenges and Future Work . . . . .	203
<b>A</b>	<b>Resumen en español</b>	<b>207</b>
A.1	Marco de la tesis, objetivos y contribuciones . . . . .	208
A.2	Amenazas de ciberseguridad en la Industria 4.0 . . . . .	214
A.3	Servicios de detección en los sistemas de control modernos . . . . .	221
A.4	Diseño de un marco de trabajo para la detección y trazabilidad de ataques persistentes avanzados . . . . .	226
A.4.1	Modelado de redes y ataques de la Industria 4.0 con teoría de grafos . . . . .	226
A.4.2	Especificación del marco de trabajo para la trazabilidad de APT . . . . .	230
A.4.3	Técnicas de correlación distribuida . . . . .	234
A.4.4	Comparación de soluciones . . . . .	240
A.5	Casos de uso para la protección de la Industria 4.0 . . . . .	241
A.5.1	Protocolo de encaminamiento de mensajes seguro . . . . .	241
A.5.2	Despliegue a un entorno de Internet de las Cosas Industrial . . . . .	244
A.5.3	Aplicabilidad en la Smart Grid . . . . .	245
A.6	Experimentación y validación de las soluciones propuestas . . . . .	248

A.7 Conclusiones y trabajo futuro . . . . .	252
<b>Bibliography</b>	<b>255</b>

# List of Figures

1.1	ISA-95 pyramid and evolution towards Industry 4.0 . . . . .	7
1.2	Overview of the Industry 4.0 infrastructure model and its enabling technologies .	13
1.3	Geographical distribution of cyber attacks against industrial systems up to the first half of 2020 (percentage of resources affected in each country) [1] . . . . .	15
1.4	Average cost generated by the increase in security breaches in 2018 in major international industrial companies [2] . . . . .	15
1.5	Attack vectors in industrial environments in the first half of 2020 [1] . . . . .	16
2.1	Reported vulnerabilities from ICS-CERT . . . . .	37
4.1	Observation rules for the election of the most dominating nodes . . . . .	65
4.2	Architecture of modern industrial organizations . . . . .	67
4.3	Example of PLOD-generated network with 10 nodes, $\alpha = 0.1$ , $\beta = 1.5$ . . . . .	68
4.4	Example of network with 10 nodes generated following the Watts-Strogatz model, with $p = 0.1$ and <i>degree</i> = 4 . . . . .	69
4.5	Example of network with five IT nodes and five OT nodes merged through two firewalls . . . . .	70
4.6	Agent implementations for information acquisition and correlation . . . . .	77
4.7	APT distributed detection and traceability framework . . . . .	79
4.8	Average consensus for three agents . . . . .	82
4.9	Calculus of the Opinion Dynamics for a set of agents . . . . .	86
4.10	Different types of clustering approaches . . . . .	87
4.11	Example of weight calculation by agent C . . . . .	91
4.12	Execution of the Opinion Dynamics after multiple stages of Stuxnet . . . . .	98
4.13	Opinion dynamics after the second stage . . . . .	98
4.14	Evolution of the opinions over time to trace the APT stages . . . . .	99
4.15	Evolution of delta opinions over the network for the Stuxnet attack . . . . .	100
4.16	Network topology used in the test case . . . . .	101
4.17	Purity average for the three test cases . . . . .	102
4.18	Evolution of the Rand Index for 10 APTs and 150 nodes . . . . .	103

5.1	Example of APT with 3 attacks. 1st: Addition of edge from node 4 to node 2. 2nd: Removal of edge from node 3 to node 6. 3rd: Removal of edge from node 6 to node 7.	108
5.2	Secret sharing scheme and shares delivery	112
5.3	Opinion dynamics after 50 attacks	113
5.4	Message loss ratio with the different strategies, 100 messages and 50 attacks over a network of 100, 200 and 300 nodes	114
5.5	Global efficiency with different strategies after 50 attacks	115
5.6	Average compromise level	126
5.7	Average QoS level	126
5.8	Stages of the Opinion Dynamics framework in a IIoT network	127
5.9	Alternatives for the opinion formation in stage 4	134
5.10	Example of network composed by two IIoT cells, using the Watts–Strogatz (WS) and Barabási–Albert (BA) model	136
5.11	Opinion Dynamics clusters after a lateral movement in the IIoT cell	137
5.12	Five subnetworks-based architecture	140
5.13	Hourly load values of Spain in 2015 [3]	143
5.14	Observation rules for the election of the driver nodes	146
5.15	Weekly consumption generated by $F$ function	149
5.16	Forecast after 10 days using ARIMA	150
5.17	Load balancing for the two proposed systems	151
5.18	Opinion dynamics after 50 attacks	152
5.19	Overview of the SealedGRID infrastructure	153
5.20	Hierarchical architecture of the PEP and PDP entities	155
5.21	Architecture of components of the PDP cloud	157
5.22	Sequence diagram for the authorization flow on the PDP-cloud	160
5.23	Network fragmentation due to Opinion Dynamics anomalies	162
5.24	Smart Grid architecture for accomodating DTS in the long term	164
5.25	DT authorization workflow in the long term	167
6.1	Example of network topology used in TI&TO	179
6.2	Test-case 1: Percentage of victories and draws	188
6.3	Test-case 2: Percentage of victories and draws	188
6.4	Test-case 3: Percentage of victories and draws	189
6.5	Test-case 4: Percentage of victories and draws	189
6.6	Example of defender-win after the attacker compromises a honeypot	190
6.7	Percentage of victories for each player in each test case	191
6.8	Overall architecture of the $I_4$ Testbed testbed	194
6.9	Interaction panel GUI on the SCADA system	195
6.10	Components of the Opinion Dynamics System	197

6.11	Evolution of the Opinion Dynamics values over the test case attack stages . . . . .	198
6.12	Evolution of delta opinions over the test case attack stages . . . . .	199
A.1	Evolución de la arquitectura industrial tradicional basada en el estándar ISA-95 y evolución hacia la Industria 4.0 . . . . .	209
A.2	Visión general del modelo de infraestructura de la Industria 4.0 y sus tecnologías asociadas . . . . .	210
A.3	Promedio del coste generado por el incremento de brechas de seguridad en 2018 en 254 empresas consultadas internacionalmente [2] . . . . .	212
A.4	Elección de nodos dominantes en un grafo . . . . .	227
A.5	Ejemplo de red con cinco nodos IT y cinco nodos OT fusionados a través de dos firewalls . . . . .	228
A.6	Implementación de los agentes de detección para la adquisición de información y correlación de anomalías . . . . .	232
A.7	Especificación de entradas y salidas del marco de detección y trazabilidad de APT	234
A.8	Cálculo de las opiniones para un conjunto de 30 nodos sometidos a 10 fases de ataque	236
A.9	Evolución de las opiniones a lo largo de las fases de ataque de Stuxnet . . . . .	237
A.10	Ejecución de Opinion Dynamics tras varias fases de la APT Stuxnet sobre una red sencilla . . . . .	238
A.11	Promedio de la pureza para las técnicas de correlación . . . . .	240
A.12	Protocolo de compartición de secretos para el envío de mensajes . . . . .	242
A.13	Ratio de mensajes perdidos para las tres estrategias de redundancia, con 100 mensajes y 50 ataques sobre una red de 100, 200 y 300 nodos . . . . .	243
A.14	Etapas para aplicar el <i>framework</i> de detección de APT . . . . .	244
A.15	Arquitectura resiliente para la red eléctrica inteligente . . . . .	246
A.16	Fragmentación de la Smart Grid en función de la información sobre amenazas provista por el algoritmo de Opinion Dynamics . . . . .	248
A.17	Porcentaje de victorias para cada jugador en 100 partidas simuladas . . . . .	249
A.18	Diagrama de red del entorno de pruebas industrial <i>I4Testbed</i> . . . . .	250
A.19	Evolución de los valores de Opinion Dynamics en las fases de ataque . . . . .	251



# List of Tables

2.1	Main Cybersecurity threats of Industry 4.0 enabling technologies . . . . .	33
2.2	Main cybersecurity threats of Industry 4.0 innovative services . . . . .	36
2.3	Overview of threats that affect industrial systems . . . . .	43
3.1	Evolution according to detection coverage . . . . .	50
3.2	Evolution according to protocol analyzed . . . . .	50
3.3	Evolution according to detection mechanism . . . . .	50
3.4	Leading companies in the market . . . . .	53
4.1	Summary of structural controllability concepts . . . . .	66
4.2	Map of <i>attackStages</i> to $\Theta$ . . . . .	72
4.3	Detection probability and decay values used in the Stuxnet test case . . . . .	97
4.4	Summary of concepts involved in the APT traceability framework . . . . .	104
5.1	Message loss ratio after 50 attacks, 100 messages and multiple topologies . . . . .	115
5.2	Network parameters collected from the different IIoT cells . . . . .	131
6.1	Map of $V$ to $\Psi$ . . . . .	178
6.2	Map of <i>attackStages</i> to $\Theta$ . . . . .	182
6.3	Summary of movements leveraged by attacker and defender . . . . .	187
6.4	Instances of the $\Psi, \Upsilon, \Theta$ ordered sets used in the simulations . . . . .	187
A.1	Principales amenazas de ciberseguridad de las tecnologías de la Industria 4.0 . . .	214
A.2	Amenazas de ciberseguridad de los servicios de la Industria 4.0 . . . . .	217
A.3	Visión general de las amenazas de los sistemas industriales y relación con las etapas de una APT . . . . .	220
A.4	Evolución de los IDS según su cobertura de detección . . . . .	222
A.5	Evolución de los IDS según el protocolo analizado . . . . .	223
A.6	Evolución de los IDS según su mecanismo de detección . . . . .	223
A.7	Asignación del conjunto <i>attackStages</i> a $\Theta$ . . . . .	230





# Acronyms

**AAA** Authentication, Authorization and Accounting.

**ABAC** Attribute-Based Access Control.

**ACF** Simple Autocorrelation function.

**AIC** Akaike Information Criteria.

**AMETIC** Asociación Multisectorial de Empresas de Tecnologías de la Información, Comunicaciones y Electrónica.

**AMI** Advanced Metering Infrastructure.

**AMQP** Advanced Message Queuing Protocol.

**AP** Access Point.

**APT** Advanced Persistent Threat.

**ARIMA** AutoRegressive Integrated Moving Average.

**BA** Barabási–Albert.

**BC** Betweenness centrality.

**BFS** Breadth-first Search.

**BMWi** German Federal Ministry for Economic Affairs and Energy.

**CAN** Controller Area Network.

**CETC** China Electronics Technology Group Corporation.

**CoAP** Constrained Application Protocol.

**CPS** Cyber-Physical systems.

**CPU** Central Processing Unit.

**CSIC** China Shipbuilding Industry Corporation.

**CVE** Common Vulnerabilities and Exposures.

**DDoS** Distributed Denial of Service.

**DDS** Data Distribution Service.

**DLT** Distributed Ledger Technology.

**DNS** Domain Name System.

**DoS** Denial of Service.

**DPI** Deep Package Inspection.

**DS** Dominating Set.

**DT** Digital Twin.

**DTLS** Datagram Transport Layer Security.

**DVR** Distance Vector Routing.

**ECSCO** European Commission and the European Cybersecurity Organization.

**EFFRA** European Factories of the Future Research Association.

**ENISA** European Union's Network and Information Security Agency.

**ENTSO-E** European Network of Transmission System Operators for Electricity.

**EPoSS** European Technology Platform on Smart Systems Integration.

**ERP** Enterprise Resource Planning.

**EU** European Union.

**FN** False Negative.

**FP** False Positive.

**FW** Firewall.

**GUI** Graphical User Interface.

**HMI** Human-Machine Interfaces.

- HTTP** Hypertext Transfer Protocol.
- ICMP** Internet Control Message Protocol.
- ICS** Industrial Control Systems.
- ICS-CERT** Industrial Control Systems Cyber Emergency Response Team.
- ICT** Information and Communication Technology.
- IDC** International Data Corporation.
- IDS** Intrusion Detection System.
- IEC** International Electrotechnical Commission.
- IEEE** Institute of Electrical and Electronics Engineers.
- IETF** Internet Engineering Task Force.
- IIC** Industrial Internet Consortium.
- IIoT** Industrial Internet of Things.
- IIRA** Internet Reference Architecture.
- IoT** Internet of Things.
- IP** Internet Protocol.
- IPS** Intrusion Prevention System.
- IS** Industrial Systems.
- ISA** International Society of Automation.
- ISACA** Information Systems Audit and Control Association.
- IT** Information Technology.
- LPWAN** Cellular Networks and Low-Power Wide-Area Network.
- LSR** Link-State Routing.
- MDMS** Meter Data Management Systems.
- MEC** Mobile Edge Computing.
- MES** Manufacturing Execution System.

**METI** Japanese Ministry of Economy, Trade and Industry.

**MQTT** Message Queue Telemetry Transport.

**NFV** Network Function Virtualization.

**NIST** National Institute of Standards and Technology.

**OPC UA** Open Platform Communications - Unified Architecture.

**OT** Operational Technology.

**P2P** Peer-to-Peer.

**PaaS** Platform-as-a-Service.

**PACF** Partial Autocorrelation function.

**PC** Personal Computer.

**PDP** Policy Decision Point.

**PDS** Power Dominating Set.

**PEP** Policy Enforcement Point.

**PIP** Policy Information Point.

**PLC** Programmable Logic Controller.

**PoW** Proof of Work.

**PPP** Public-Private-Partnership.

**QoS** Quality of Service.

**RAM** Random Access Memory.

**RAMI4.0** Reference Architectural Model Industrie 4.0.

**RAT** Remote Access Technology.

**RBAC** Role-Based Access Control.

**RTU** Remote Terminal Unit.

**SCADA** Supervisory Control and Data Acquisition.



**SDN** Software Defined Networking.

**SG** Smart Grid.

**SOAP** Simple Object Access Protocol.

**SPIRE** Sustainable Process Industry through Resource and Energy Efficiency.

**SSH** Secure SHell.

**TCP** Transmission Control Protocol.

**TN** True Negative.

**TP** True Positive.

**TSO** Transmission System Operators.

**UDP** User Datagram Protocol.

**URI** Uniform Resource Identifier.

**URL** Uniform Resource Locator.

**VPN** Virtual Private Network.

**WLAN** Wireless Local Area Network.

**WPAN** Wireless Personal Area Network.

**WS** Watts–Strogatz.

**XACML** eXtensible Access Control Markup Language.

**XMPP** Extensible Messaging and Presence Protocol.

**XSS** cross-site scripting.



# Chapter 1

## Introduction

### 1.1 Industry 4.0 and Current Issues

Industry 4.0 refers to the digitization of all components within the industry [4]. It is also called the fourth industrial revolution, in reference to the technological modernization process that is currently taking place in the Industrial Control Systems (ICS) and critical infrastructures. Whereas the first revolution coincided with the introduction of steam engines in the 18th century, the second revolved around the use of electricity in the late 19th century, before electronics were introduced to automate manufacturing processes in 1970s, giving rise to the third industrial revolution. Following with this tendency, the Industry 4.0 concept is not so mature due to a lack of agreement on the set of technologies considered and the different interests of the actors involved (e.g., researchers, standardization committees, governments) [5]. However, it can be defined from a technical perspective as the combination of productive processes with leading technologies of information and communications. This allows all the elements that conform the productive processes (suppliers, plant, distributors, even the product itself) to be digitally connected, providing a highly integrated value chain [4].

In this transition stage, we find that adapting the existing control processes during the third industrial revolution to the new paradigm is one of the main challenges of Industry 4.0. Traditionally, these industrial facilities and critical infrastructures have been governed by SCADA (Supervisory Control and Data Acquisition) systems, which provide real-time data and remote management of the devices that are deployed over the production cycle, such as Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), or field devices (i.e., sensors or actuators). Nevertheless, these systems, henceforth referred to as Operational Technology (OT), are now experiencing a growing interconnection with Information Technology (IT) to share data and uptake new business processes. This is a consequence of the standardization of the software and hardware used in control systems [6], mainly caused by the adoption of Ethernet or TCP/IP networking and wireless technologies in an often critical and until recently isolated environment.

While the integration of the IT and OT worlds has several major benefits, it has also facilitated the emergence of several IT attack vectors in industrial systems (IS) [7]. We refer to attack vectors such as denial of service, presence of malware in the control teams, exploitation of vulnerabilities in communication protocols to intercept traffic, phishing and social engineering, etc. These can be leveraged and combined to perpetrate sophisticated attacks such as an Advanced Persistent Threat (APT) as the case of Stuxnet [8], that ultimately disrupts and damages critical infrastructural operations with a severe impact, ranging from economic costs to pollution or even loss of human lives. These issues have obligated researchers and security officials to seek for advanced defense techniques to tackle complex attacks, which narrows down the focus of this dissertation. Namely, *this doctoral thesis aims to study and design security mechanisms capable of detecting and tracing advanced cybersecurity threats and hence ensure the continuity of the production line at all times*, as it is expected that the number and impact of these cybersecurity threats will increase in future industrial environments.

To address this problem, numerous solutions have emerged from the academic and commercial point of view that are focused on the detection of intrusions in the industrial network, also known as Intrusion Detection Systems (IDSs). Although the analysis of the algorithms underlying these mechanisms has already been approached in depth [9, 10], it is necessary to prospectively analyze the requirements that these systems have to satisfy with the evolution of the control systems in the short, medium and long term, after the integration of the areas of Industry 4.0 and with them the appearance of new threats of cybersecurity. However, and due to the lack of analyses on this subject, this does not only include the analysis of security threats that affect the building blocks of this novel approach in this setting, but also the potential vulnerabilities that might arise due to the creation of novel industrial services in the upcoming years.

Therefore, we position our research in the analysis of advanced detection systems that enable the traceability of APTs in the Industry 4.0 and also ease the deployment of effective countermeasures. In this context, the Industry 4.0 paradigm usually collides with the concept of Industrial Internet of Things (IIoT), which is incidentally addressed in this work and introduced in the following sections. Whereas the latter is envisaged to interconnect industrial assets to the Internet and the cloud with the aim to collect data and optimize processes, Industry 4.0 generally adopts a wider scope by proposing a hyper-connected ecosystem model across several industrial sectors, and it is enabled by the IIoT and other technologies [11]. In other words, Industry 4.0 can be considered as a bigger-picture framework that would not exist without the IIoT from a technical perspective, even though both terms are frequently interchangeable in the literature, as it is an unstable area where the multiple actors involved (international initiatives and consortia) are continuously redefining both concepts and their technologies that comprise them. Likewise, Industry 4.0 is often associated with the manufacturing sector, although both can be applied to a plethora of industrial sectors (e.g., health, manufacturing, transport) that are also described next. In this sense, for the interest of the analysis and to better characterize our research, we will focus on industrial ecosystems for the generic industry, but contextualizing our findings on the



systems in charge of the Smart Grid (SG), as a use case of traditional sector (i.e., the energy) whose critical infrastructures meet the Industry 4.0 paradigm, by providing intelligence to the electric grid in its entire value chain. Therefore, in the remainder of this introductory section, we analyze the evolution of this paradigm and its enabling technologies, by firstly identifying the stakeholders involved in the Industry 4.0, to lay the background of this doctoral thesis. Afterwards, the challenges around the cybersecurity of Industry 4.0 elements and particularly the APTs will be introduced.

### 1.1.1 International Initiatives and Consortia

Currently, a great effort is being made at the European and global level to promote the concept of Industry 4.0. As introduced earlier, it is understood as the integration of cutting-edge information and communication technologies in the industry, accompanied by a particularly intense work on cybersecurity. The most important initiatives and consortia at the international level in this direction are described below.

The industrial control systems that govern critical infrastructures (transport, nuclear, etc.) have traditionally been isolated from external networks (such as the Internet), but in recent years the trend is to incorporate modern technologies due to the drop in costs and the standardization of software and hardware [6]. As a result, these environments are facing a substantial increase in connectivity and complexity, which is making the traditional model of industry (rigid and hierarchical) evolve towards a distributed model where the various actors involved in the production process (e.g., suppliers, operators, customers) interact transparently, obtaining information without interruption and optimizing the production cycle in all sectors and at all levels.

In order to guide that evolution under a sustainable development and following common standards in the industry, different initiatives have emerged in the last years at international level. The first reference to the concept of ‘Industry 4.0’ was originally coined by the German government, after a series of projects aimed at promoting the digitalization of its production processes that led to the establishment of the *Plattform Industrie 4.0* program [12]. Following this proposal, various initiatives were implemented at the national level throughout Europe, in line with the objectives of the European Union H2020 strategy [13] and derived from the original platform. For example, the initiative *Smart Industry* (Netherlands, Sweden), *Catapult* (UK), *Industrie du futur* (France), *Fabbrica Inteligente* (Italy), and *Made Different* (Belgium). In Spain, AMETIC (Asociación Multisectorial de Empresas de Tecnologías de la Información, Comunicaciones y Electrónica) set up the *Industry 4.0* commission [14], whose objectives are in line with the objectives of the H2020 strategy, and which gave rise to the *Industria Conectada 4.0* initiative [15], which is responsible for raising awareness of the Industry 4.0 concept at national level between companies, promoting regional conferences, and providing information to the AENOR standards group.

The synergy between the multinationals and the strategies of the European Union in favour of the digital transformation of the industry has also led to the formation of important consortia, PPPs (Public-Private Partnerships) and working groups at international level. For instance, the *Factories of the Future (FoF)*, a PPP that emerged under the EU economic recovery plan and involves industrial companies and academic institutions to implement the vision of Industry 4.0 [16]. As a result, 240 projects were carried out with the involvement of more than 2000 entities throughout Europe until 2017. Another PPP aligned with the EU's industrial growth initiatives is *SPIRE (Sustainable Process Industry through Resource and Energy Efficiency)*, which comprises industrial companies with innovative processes in all sectors and more than 130 research-related entities throughout Europe. Its agenda until 2050 includes the implementation of good practices and innovative technologies in production systems, as well as an intense reduction in CO<sub>2</sub> emissions. Similar objectives are pursued by Energy Efficient Buildings (E2B), another PPP emerged under the H2020 program, extending the digital transformation of the industry to other domains such as intelligent cities, in this case seeking technological solutions for the construction of districts and connected buildings [17].

There are also several important consortia of companies with great impact at European level that join the efforts of entities and academic institutions from multiple disciplines, in order to promote the new model of Industry 4.0. Examples include the *European Technology Platform on Smart Systems Integration (EPoSS)*, focused on the use of R&D with smart systems), the *ARTEMIS* association (researching cyber-physical systems for industry) or *AENEAS* (with research into new electronic systems and components). The members of these three are grouped under the European Union PPP *ECSEL* (the Public-Private Partnership for Electronic Components and Systems), which has a current capital of 2.6 billion euros in research projects in this area [18]. These associations include companies such as NXP, Hitachi, Intel and Airbus.

Beyond European borders, there are various government efforts and confluences of companies that pursue the objective of promoting a model equivalent to that of the original German Industry 4.0, either independently or directly in collaboration with the German *Industrie 4.0 program*. For example, the United States formed a national network for industry innovation with funding of \$1 billion in public funds to advance research in digital design and manufacturing [19], and \$2.2 billion dollars to renew the American industry and compete with the large Chinese market [20], in which a growth of 12.9% is expected for the year 2023. This US national network includes a consortium of global scope oriented to the integration of IIoT and CPS technologies, the *Industrial Internet Consortium (IIC)* [21]. Its primary objective is the integral automation of the industry in different domains, critical infrastructures and applications. It contains different work groups (including a specific one on cybersecurity) composed of four subgroups specialized in mobile devices, applicability in real contexts, testbed and trust; and in which companies such as Dell, Huawei, Cisco, General Electric or IBM collaborate.

With regards to China, the government launched in 2015 the strategies *Made In China 2025*, *Industrial Internet* and *Internet Plus*, all of them keeping certain similarities to the German

*Industrie 4.0* program, not only to stay in the competing market but to position itself as the number one leader in that competing market along with the United States and Germany. According to the *Made In China 2025* strategy, the goal is to significantly reduce inventory costs (20-50%), increase automation and production (45-55%), and increase the precision and quality of production (+85%), without forgetting cybersecurity in all areas of application. To this end, in 2015, the Chinese Ministry approved 94 research projects [22], all within the theme of smart and safe manufacturing, in addition to groups such as *China Electronics Technology Group Corporation* (CETC) and *China Shipbuilding Industry Corporation* (CSIC) fostering smart manufacturing initiatives to promote the above strategies and update all elements of the industrial ecosystems in all areas (e.g., human capital, management, process optimization, quality control) and at all levels (from low-level manufacturing to high-tech industry-related processes) [23]. It is expected that by 2025 the exercise will be completed in all regions of China and all industrial sectors, with a return on investment in the order of 46% or \$32.3 billion in the future [24].

Finally, there are also national initiatives in other countries that were established following the trail of the original vision of Industry 4.0 but incorporating specific needs at the national level. For instance, in Japan, they collaborate bilaterally with Germany to address some relevant aspects in the different perspectives of Industry 4.0. Specifically, in April 2016, the *Japanese Ministry of Economy, Trade and Industry* (METI) and the *German Federal Ministry for Economic Affairs and Energy* (BMWi) signed a Japan-German cooperation agreement on IoT/Industry 4.0 [25]. In this way, Japanese initiatives such as the *Robot Roadmap Revolution Initiative*, which is related to the *Value Chain Initiative* (IVI), can benefit from incorporating Industry 4.0 elements within their production chains [26]. Similarly, the Australian and German Ministries initiated joint collaborations in 2015 by establishing the Australian-German Advisory Group [27]. One of their initiatives was to invest in a \$5 million pilot research program to explore the adaptation of Industry 4.0 in Australia through five universities, which were made available to support real-world installations as of September of 2018 [28].

From this panorama it can be deduced that, although the *Industrie 4.0* program was born as a German initiative with a mainly European scope, it has achieved a global reach due to its influence on other programs and collaborations between various consortia. In this sense, the IIC can be considered as the consortium with equivalent importance at an international level but with American origin, mainly due both to its global reach and to its maturity and the importance of its members. As aforementioned, this leads to one of the concepts that is frequently related to Industry 4.0: the IIoT, initially promoted by American companies (AT&T, Cisco, General Electric, IBM, and Intel) - although it currently has an international presence. Both pursue similar objectives (the digitization of industry), but with slight differences. While Industry 4.0 focuses its efforts mainly on manufacturing processes, Industrial IoT also seeks integration with various domains (e.g., critical infrastructure, smart cities). As an example, Industry 4.0 focuses on the processes related to the manufacturing of a car, while the IIoT also focuses on the physical interaction between the car and infrastructures such as Smart Grid, vehicle networks,

and others [29]. In addition, Industry 4.0 focuses more on hardware (production machinery and communications protocols) and coordination of production processes, while the IIoT focuses more on software (component integration) and the interaction between entities [30]. Even so, there are points in common between both initiatives, and they are currently working to align their two reference architectures, the Industrial Internet Reference Architecture (IIRA), developed by the Industrial Internet consortium [31], and the Reference Architectural Model Industrie 4.0 (RAMI4.0), developed by the Platform Industrie 4.0 consortium [32]. Both references provide two interoperable service-oriented architectures, which will combine IT and OT components accessible through common interfaces, and interconnected through communication infrastructures of various types [33], such as DDS (Data Distribution Service) and/or OPC UA (Open Platform Communications Unified Architecture), an evolution of the OPC specification that includes better semantic information modeling capabilities [34]. This would allow a transparent access to the various resources from all processes and entities of the organization, thus achieving the digitization of the network and the decentralized model pursued by the industry of the future.

Furthermore, it is worth mentioning that the scope of the aforementioned reference architectures, IIRA and RAMI4.0, goes beyond purely industrial and manufacturing environments and sectors, being applicable to several essential sectors in our economy such as electrical networks, logistics and transport systems, digital health, intelligent environments (e.g., smart cities), and many others. Precisely, there are several test benches where the capabilities of these architectures are being studied, offering services such as the provision of Platform-as-a-Service (PaaS) services in production lines, the creation of energy microgrids, the interaction between vehicles through vehicle-vehicle (V2V) and vehicle-infrastructure (V2I) infrastructures, the definition of an ecosystem for the remote monitoring of patients, intelligent baggage management in airlines, and intelligent water supply in urban environments [35].

As a result, the Industry 4.0 ecosystem is especially varied and subject to various initiatives that guide the research and development of the industry of the future. In the following section, we delve into the actual principles and advantages of the 4.0 industry by analyzing the structural changes they bring over traditional infrastructures, and presenting the building blocks that shape this paradigm from a technological point of view.

### 1.1.2 Review of Innovations and Enabling Technologies

In order to better understand the innovations that Industry 4.0 introduces in the existing infrastructures, we must pay attention to its architectural changes. The ISA-95 standard [36] defines five levels of operations in the industrial automation, in the form of a pyramid, as illustrated in Figure 1.1. This way, the productive process itself is located in the base (level 0), whereas those devices that interact with it (i.e., PLCs) are set in level 1. On top of these (level 2), we find the devices that control the production process such as SCADAs or Human-Machine Interfaces (HMIs), and those that control the workflow, like Manufacturing Execution Systems (MESs), are

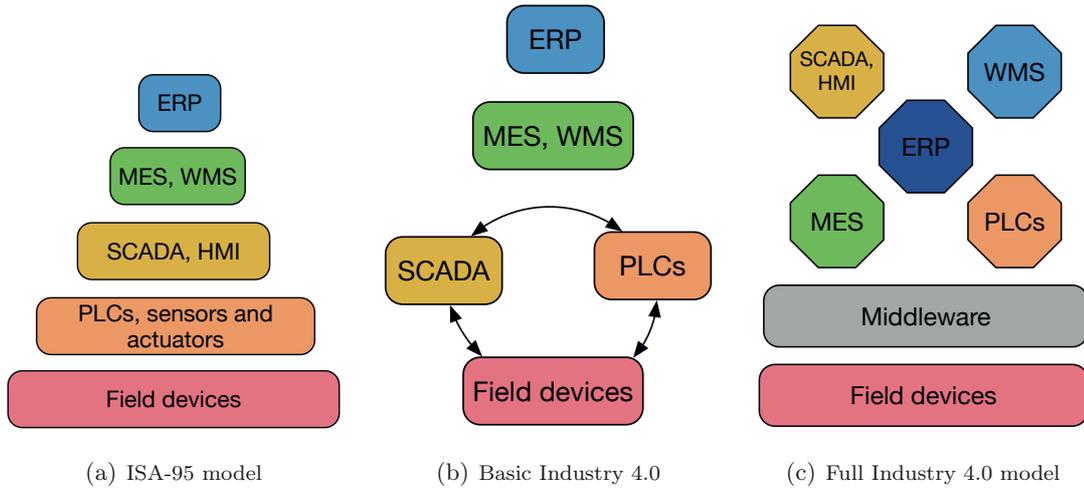


Figure 1.1: ISA-95 pyramid and evolution towards Industry 4.0

represented at level 3. Lastly, the highest level contains the infrastructure of logistics, inventory, and Enterprise Resource Planning (ERP).

In traditional industrial environments, the information processing infrastructure follows the pyramidal structure reflected by this standard. One of the objectives of researchers in the field of Industry 4.0 is to analyze how to change this pyramid to a model that provides a more dynamic and reconfigurable decentralized infrastructure [37], as depicted in Figure 1.1. By creating well defined services and interfaces, in which each element of the ecosystem has a specific functionality and purpose, it would be possible to redefine the structure of an industrial environment through various configurations, enhancing new services and optimizing existing ones [38]. The following is a summary of the most common conceptual features that this new model would enable:

- **Interoperability.** The application of the technologies that belong to the Industry 4.0 would ensure an interoperability between each of the elements of the productive processes.
- **Virtualization.** Within industry 4.0, it would be possible to create a virtual copy of each of its elements.
- **Decentralization.** Each of the elements of Industry 4.0 might be able to intelligently make decisions for itself, in conjunction with other elements, or globally.
- **Capabilities in real time.** The ecosystem would allow the acquisition and analysis of data in real time.
- **Service orientation.** The elements of Industry 4.0 would be able to abstract their functionality into a service-oriented architecture, and would also be able to consume services offered by other assets. In addition, these services would be indexed and easily accessible by authorized entities.

- **Modularity.** An Industry 4.0 environment would not function in a monolithic way, but would allow adaptation to new requirements by integrating new modules and extending or replacing existing modules.
- **Interactivity.** Industry 4.0 operators at all levels would be able to interact with various physical and logical elements in a simple and effective way.

In addition, these advantages of Industry 4.0 have been identified by most potential stakeholders:

- **Flexibility.** The adaptability of the production processes will allow greater flexibility during the operations of the production system, such as when producing customized products.
- **Efficiency.** The productive processes will improve in efficiency, both from the energy point of view and from the perspective of efficient use of the available assets, thanks to a better decision making as a consequence of a greater data collection and analysis.
- **Productivity.** Due to the optimizations of the productive process, it will be possible not only to increase the global efficiency of all the processes, but also to increase the speed of order processing.
- **Risk reduction.** Thanks to the knowledge of potential problems within the productive processes, it will be possible to reduce the risks in the use of the machinery, as well as to increase the quality of the manufactured product.
- **Decision making.** The access to a large amount of information by the operators of the Industry 4.0 will allow them to make better decisions. This information will go from the long term production plans of the factory to real time information about the availability of each product or relevant events in the production line.

This set of improvements aims to revolutionize the industry model in many fields. Although Industry 4.0 has been traditionally oriented to production systems, it is possible to apply its ideas and those of other related paradigms (e.g., Industrial Internet of Things) to other sectors and critical infrastructures of society. The following is a list of potential markets, sectors and critical infrastructures where Industry 4.0 and its related strategies can be applied:

- **Production systems:** the objective is that the productive processes can govern themselves, taking corrective actions that avoid unplanned stops and readjusting the system components in real time according to the needs.
- **Energy:** including electricity generation, transmission and distribution (i.e., the Smart Grid), gas, oil, and nuclear industry.

- **Water:** ranging from the provision of water to control of quality.
- **Health:** medical and hospital care, as well as medicine infrastructures, pharmaceuticals and bio-laboratories.
- **Food supply:** including its safety and security
- **Transport:** whether by road, sea or air traffic, as well as border surveillance and transportation of goods.
- **Financial Systems:** banking infrastructures, government financial assignments and payment services.
- **Chemical industry:** production, manipulation and storage of dangerous substances, pipelines of dangerous goods.

It is important to emphasize that the concept of sector in this dissertation is too broad to encompass all the underlying complexities of each of these infrastructures. That is why in this work we focus on services, as well as on the information flows between physical and cyberphysical entities in Industry 4.0, taking into account their function within society. From a more technical point of view, all the principles and advantages of Industry 4.0 can be accomplished by a set of enabling technologies that can be generally summarized into five main areas: Industrial Internet of Things, cloud and fog computing, Big Data, blockchain and virtualization.

Firstly, the goal of the **Internet of Things** (IoT) paradigm is to massively interconnect the objects that surround us – the ‘things’ – using standardized interfaces, allowing them to produce and consume services [39]. Applied to the industrial context, the so-called Industrial Internet of Things vertically integrates all the components within the architecture, ranging from control systems to machines or even the product itself. Moreover, due to their interconnection capabilities, all entities could interact with each other at a horizontal level, enabling decentralized interactions such as monitorization (between human operators and machinery) and decision making (between the machines themselves). There are other concepts that are related to the IoT and can also be applied to this context, such as Cyber-Physical systems (CPS). This term was coined by Helen Gill at the National Science Foundation in the United States, who defined it as ‘a new generation of systems with integrated computational and physical capabilities that can interact with humans through many new modalities’ [40]. Note that CPS focus on feedback between systems (i.e., looping) in a more local environment, while IIoT assumes a greater global connectivity.

**Cloud computing** can be considered as another of the pillars of Industry 4.0 for a variety of reasons. On the one hand, it carries on the analytic procedures with the data provided by the industrial process, retrieved by IIoT devices. On the other hand, it provides support for the delegation of production processes and control to the cloud – enabling new productive processes (e.g., product customization) and innovative services such as ‘cloud-based manufacturing’ [41].

However, there are various situations, such as management of swarms of robots, where the cloud might not be the most suitable solution due to its inherent features (high latency and jitter, lack of local contextual information). For this very purpose, it can be possible to apply emerging paradigms such as *fog computing* [42], which focus on the deployment of cloud-like services at the edge of the network.

Third, Industry 4.0 facilitates the evolution of industrial decision making processes, mainly due to the multiple sources of information that are available to both human operators and systems alike. In order to distill all this information and extract both business and operational intelligence, it is necessary to conduct advanced data analytics procedures. This area includes both the analysis of information at a more local level (e.g., the independent optimization of the operation of a machine based on its interactions with other elements of the production line) and the concept of **Big Data** - the processing of all information provided by entities of the industrial ecosystem, looking for added value services such as monitoring the operation of the ecosystem entities, process optimization, and the identification of anomalies.

Beyond the analysis of information, the integrity and security of data at rest in the long term is also critical for uptaking auditing procedures, which can be enabled by the promising Distributed Ledger Technologies (DLTs) such as a **blockchain**. These have been used in this area as a transparent, tamper-proof and secure system that enables a plethora of business applications, ranging from P2P energy trading in microgrids [43] to record keeping systems with privacy protection [44]. A blockchain consists of a shared and distributed database that offers the synchronization of immutable but linkable information sorted in chronological order. When combined with smart contracts (i.e., user-defined programs executed in the ledger), it enables an accurate traceability of events between the different devices and partners, ensuring the veracity of data while also removing the need of intermediaries.

Lastly, we can highlight a group of technologies whose target is to change the way of designing and interacting with the production chain, denoted here as **virtualization**. One of these consists in the creation of virtual representations (e.g., 3D abstractions [45]) of all machines and components involved in the production process. This is facilitated by the previously mentioned enabling technologies, and it will allow the creation of novel services based on the concept of ‘digital twins’, where it will be possible to conduct simulations to prevent failures and optimize the production line. Aside from this paradigm, the introduction of modern HMIs can also be included in this category, that make use of augmented and virtual reality devices that ultimately make the operations easier and more flexible for the workers. In addition, the use of advanced robots (autonomous, mobile, modular, multifunctional, etc.) also contribute to improve the performance of certain tasks within the production chain.

Due to the technological particularities that often converge between Industry 4.0 and the internet of industrial things (as explained above), and according to the intimate relationship with the work presented in this thesis, we will now go a little deeper into the software and hardware



that support the IIoT vision. This analysis is of special interest to understand the cybersecurity problems that occur to all current industrial elements.

### 1.1.3 Hardware and Software in the Industrial Internet of Things

As previously stated, there are multiple actors that are defining the technologies that comprise the Industry 4.0 and specifically the IIoT [46]. Such actors include various standardization groups and several consortia such as the IIC [31] and the Platform Industrie 4.0 consortium [32]. As a result, the IIoT technology ecosystem is very heterogeneous, ranging from standards that originated from specific industry verticals to protocols that were designed for general-purpose use. These technologies provide all the necessary components to build a functional IIoT infrastructure: from hardware and software platforms to communication technologies at the lower and upper layers of the networking stack.

From a **hardware perspective**, a ‘thing’ in the IIoT can be any sensing or actuating device that interacts with the physical world and can be accessed through the Internet protocol suite – either directly or indirectly. These entities range from existing industrial devices enhanced with additional networking capabilities and high-level services (e.g., PLCs equipped with the MQTT protocol [47]) to sensor/actuator devices equipped with wireless connectivity (e.g., WirelessHART sensors forming a capillary network [48]). The capabilities of these devices in terms of memory and computational power is also very heterogeneous, ranging from constrained nodes to more capable devices.

From a **software perspective**, there are various reference architectures whose goal is to provide additional services beyond the basic exchange of data, including operation, management, business logic, and security. As introduced earlier, the most important reference architectures are the IIRA and the RAMI4.0. Although as of 2020 there are no complete instantiations of these reference architectures, the functionality of some of their components is being verified through the use of testbeds. Moreover, certain major industry players, such as Siemens [49], already provide basic IIoT solutions.

As for the communication technologies and protocols, they can be classified into two categories: lower layer protocols and upper layer protocols. **Lower layer protocols** are deployed under the network layer (the IP layer of the TCP/IP stack), and in the context of the IIoT all protocols make use of a wireless transmission channel (cf. [46]). These protocols can be classified as:

- *Wireless Personal Area Networks (WPAN)*. WPAN protocols used in IIoT solutions include standards such as IEEE 802.15.4 [50] and Bluetooth. In most cases, due to the limited resources available to constrained nodes, WPAN networks will not make use of the standard IP layer (IPv4, IPv6) protocols, but different protocols – either standardized subsets of the IP protocol (e.g., 6LowPAN [51], 6TiSCH [52]) or other proprietary protocols (WirelessHART, ISA100.11a [53]). In all cases, it is mandatory to deploy a gateway between the WPAN network and the industrial network. Such gateway will be deployed at the industrial premises.

- *Wireless Local Area Networks (WLAN)*. The family of 802.11 standards is the most common WLAN technology used in industrial settings. As with WPAN protocols, the gateway that connects the WLAN network with the industrial OT and/or IT networks will be deployed at the industrial premises. It should be pointed out that in contrast with the majority of WPAN networks, in industrial WLAN networks no routing is necessary between the endpoint and the gateway.
- *Cellular Networks and Low-Power Wide-Area Networks (LPWAN)* This category includes both general-purpose cellular technologies (e.g., 4G, 5G) and solutions specifically designed for IoT devices (e.g., NB-IoT, Sigfox, LoRa [54]). Compared to WPAN and WLAN protocols, the information firstly traverses the telecommunications network before reaching the specific industrial network that consumes the information – which can be located on premises or in the cloud.

In the context of industrial networks, the main difference between these technologies is the location of the gateway with wireless connection with the industrial network. In WPAN and WLAN, gateways can be deployed and controlled at the industrial premises, while in cellular networks data must first traverse the telecommunications network before reaching the specific industrial network that consumes the information – which can be located on premises or in the cloud. Also, most WPAN networks make use of subsets of the IP standards (e.g., 6LowPAN) or proprietary protocols (e.g., WirelessHART).

**Upper layer protocols** are deployed over the transport layer (TCP or UDP), and allow the exchange of information in a shared data structure between participants. The most important upper layer IIoT protocols as defined in Liao et al. systematic literature review [46] (which largely correspond to the connectivity framework protocols presented in reference architectures such as [31]) can be categorized as follows:

- *Messaging and data-oriented protocols*. This category includes protocols specialized in providing asynchronous message queuing between various interested parties [55]. In the context of the IIoT, the most commonly used protocol is MQTT, which provides a lightweight publish-subscribe mechanism that is suitable for constrained nodes. Other actors have also considered the integration of other, more complex data-centric protocols such as DDS, AMQP [56], and XMPP [57].
- *Web Services*. Most IIoT web services rely on a RESTful style of architecture, where resources are mapped to URIs, and HTTP requests are sent to perform operations in such resources. Other, more complex web service frameworks like SOAP (Simple Object Access Protocol) are less used in this context. Note that there are protocols like CoAP that do not make use of HTTP, and are specifically designed to provide web services to constrained nodes [58].

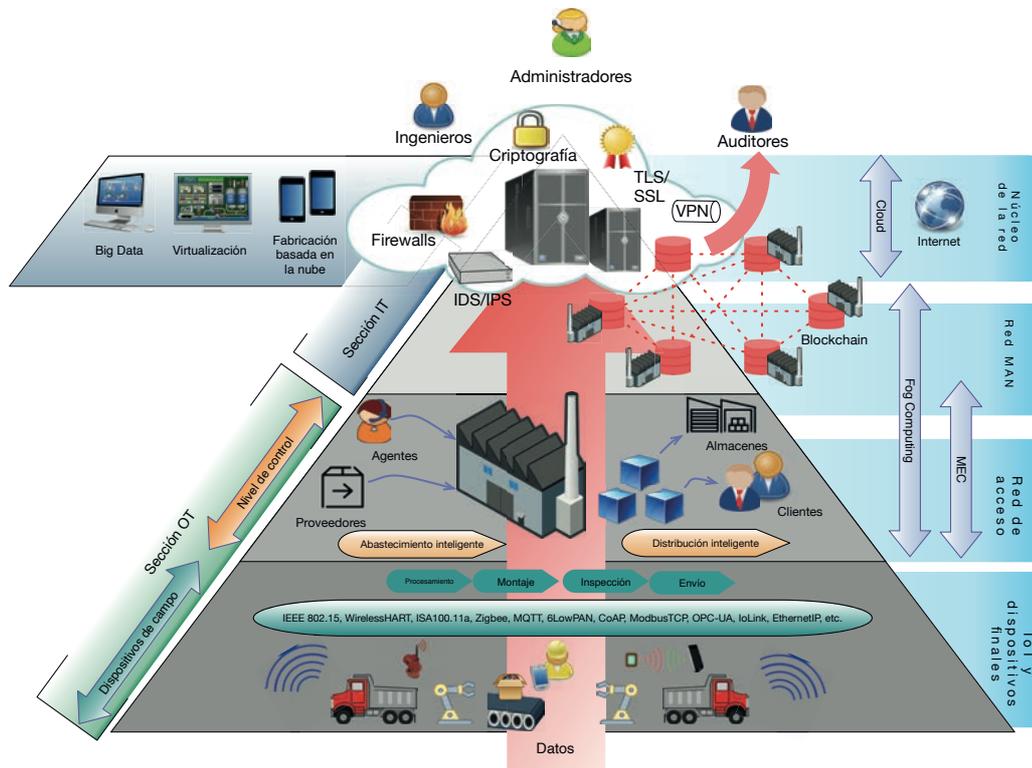


Figure 1.2: Overview of the Industry 4.0 infrastructure model and its enabling technologies

- *Specific frameworks.* There are various frameworks that were specifically designed to fulfil the needs of certain industry verticals (e.g., manufacturing, telecommunications), yet are flexible enough to be applied to most IIoT scenarios. Examples of protocols used by these frameworks include OPC UA and OneM2M (a service layer that provide efficient communication between application endpoints [59]).

Altogether, these areas of technologies will allow the industry to flexibly model the operations performed within the production life cycle, enhancing their efficiency based on the heterogeneous information exchanged between all the components involved, that ultimately reduces risks and achieves a better decision making. It is possible to perform a prospective analysis on the application of the aforementioned technologies in the short (directly applicable), medium (existing proofs of concept) and long term (nowadays limited to theoretical research):

- **Short term:** it includes communication protocols based on Ethernet or TCP/IP already being applied, and those aiming to achieve a higher interoperability between different systems, like IO-Link or OPC UA. It is also considerable some technologies that assist the personnel within the organization: visualization of throughput, assets location, smart inventory, etc.

- **Medium term:** they mostly consist in an evolution of already available processes in the short term, such as advanced assisting technologies (e.g., augmented reality, wearable devices). On the other hand, it includes the integration of cloud computing and IoT technologies to enable real-time communication and a deeper integration between all elements of the value chain.
- **Long term:** Industry 4.0 makes use of virtualization and artificial intelligence to run simulations and predictions of processes that ultimately deploy a fully decentralized, dynamic and reconfigurable model.

Once we have introduced the enabling pieces of the Industry 4.0 from a hardware and software perspective, we have a clear picture of the technological landscape that sets the background of this dissertation. In this sense, an overall view of the Industry 4.0 infrastructure with the integration of all these technologies is illustrated in Figure 1.2. In the following section, we will enumerate and analyse the set of future cybersecurity threats that might appear in the industry environment as a consequence of the introduction of the main technologies mentioned before.

## 1.2 Overview of Cybersecurity Challenges in the Industry 4.0

Several specialized consulting firms point out the need to allocate funds in cybersecurity in order to increase confidence and ensure the adoption of Industry 4.0 technologies in all sectors. Reports such as that of the International Data Corporation (IDC), which highlights that in 2018 24% of organizations conceive IIoT as a determining element for the transformation of their business, contrast with other less hopeful statistics: among them, a report indicating that 70% of manufacturing companies deal with sensitive information accessible over the Internet, despite the fact that only 55% of them claim to employ encryption in communications [60]. Another example is provided by Kaspersky with the proportion of industrial infrastructures attacked worldwide [1], as shown in Figure 1.3.

This trend translates into millions of dollars in losses. In its 2017 survey of 254 major industries in seven countries, Accenture estimated an average of \$13 million in costs in 2018 resulting from an increase of 12% in security breaches over 2017 [2], especially affecting critical infrastructure in the financial sector, as shown in the graph in Figure 1.4.

Most of these security threats arise due to the particular characteristics of Industry 4.0 environments and the infrastructure model applied. As pointed out before, a gradual transition is currently underway from the pyramidal model (which includes the integration of legacy systems and their associated vulnerabilities) to a more decentralized architecture [61]. This trend is still on the rise, as predicted by firms such as Gartner [62], which anticipate a Core modernization and decentralization of cloud resources in the infrastructure models of all industrial sectors by 2021. This tendency is favoring the emergence of collaborative spaces between industrial partners, such as so-called cloud manufacturing [63]. However, this transition also brings a more heterogeneous

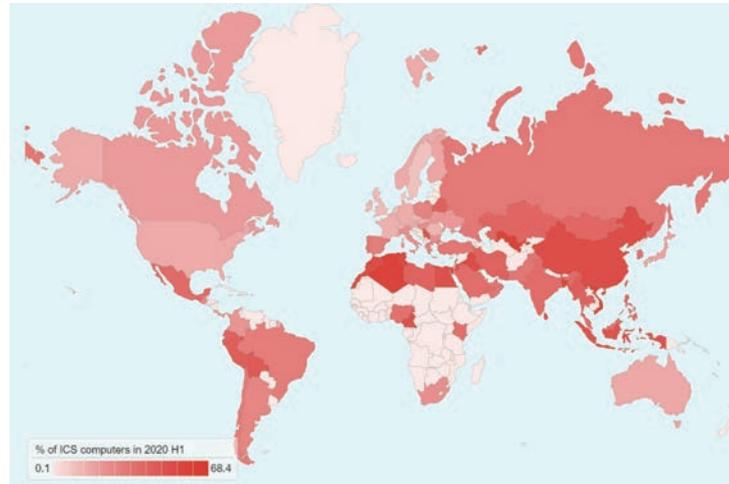


Figure 1.3: Geographical distribution of cyber attacks against industrial systems up to the first half of 2020 (percentage of resources affected in each country) [1]

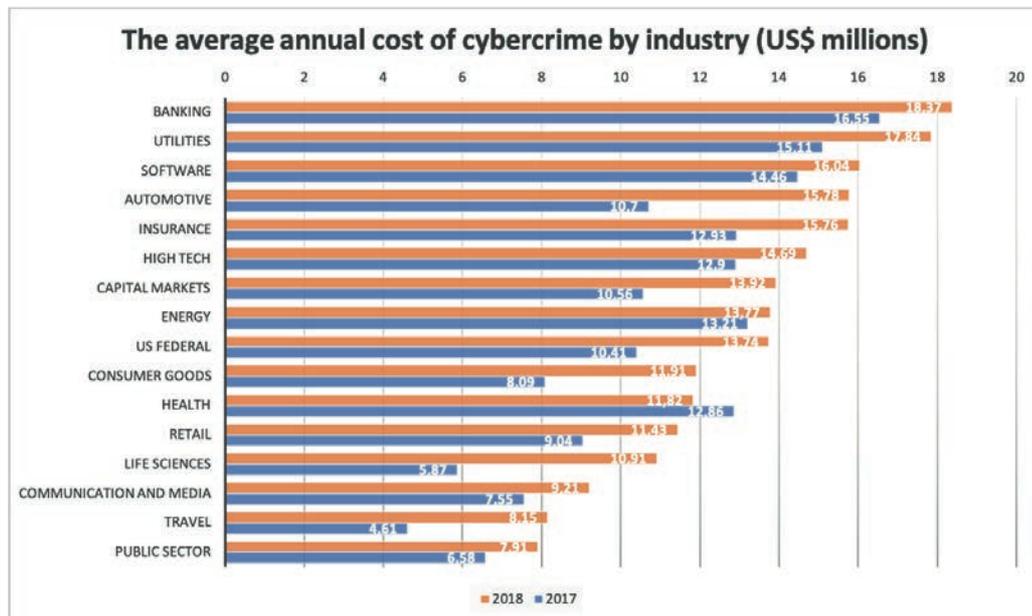


Figure 1.4: Average cost generated by the increase in security breaches in 2018 in major international industrial companies [2]

and complex environment, where any element could interact and cooperate with any other element in real time. This set of principles come enabled by a number of such disruptive technologies in traditionally isolated environments as the IoT, cloud computing, data mining (or Big Data), blockchain and virtualization, as introduced before.

In this sense, advanced interaction elements such as machine-human interfaces, ‘digital twins’ [64] or autonomous agents for the autonomous organization of the production chain [65] come into play. These elements can exert a direct influence on the behavior of the other agents. If the information collected by the agents is manipulated, or if the integrity of the agent itself is breached, it is possible to launch various attacks aimed at extracting the flow of information going to the agent and the information created by the agent itself, which can be spread by all the other components surrounding it. An example of the security issues generated by this increase in the complexity and heterogeneity of technologies is the statistic shown in Figure 1.5, which shows the wide range of attack vectors up to mid-2020 [66].

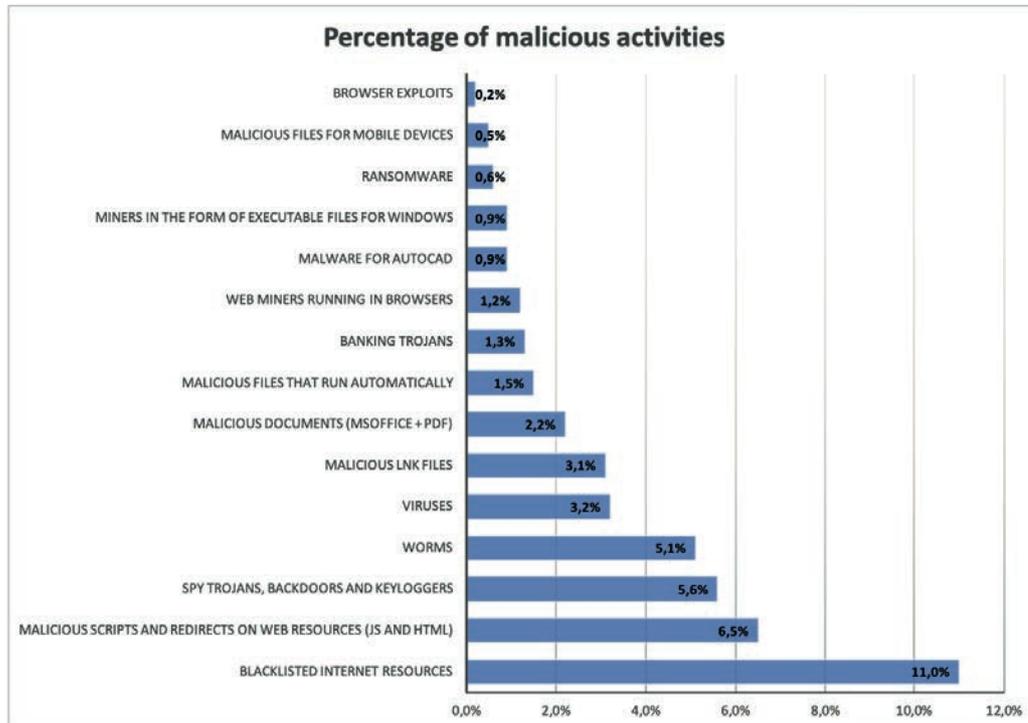


Figure 1.5: Attack vectors in industrial environments in the first half of 2020 [1]

Consequently, we can see that all the traditional security properties can be jeopardized with the new Industry paradigm due to the nature of the new threats [67] operating under different threat modes [68] that have not been addressed before. From the point of view of availability, it would be possible to launch a denial of service (DoS) attack from any element of the infrastructure itself. In terms of integrity, the manipulation of Industry 4.0 technologies can enables an adversary to manipulate not only local behavior, but also global behavior through distributed and cooperative

decision processes. At the confidentiality level, the amount of sensitive information managed by local entities will increase, with a consequent increase in the risk and impact of attacks. Authentication suffers as barriers between different subsystems become blurred and technologies such as Big Data and virtualization are integrated. Finally, privacy is also at risk, both at the human level and at the level of the industrial sector companies themselves.

As a result, an industrial system becomes complex and critical, besieged by multiple attack vectors that can be ultimately leveraged to perpetrate an APT [69, 70]. This represents a sophisticated attack perpetrated by an expert adversary, and is characterized for its ability to go undetected within the victim network for a certain period of time. Due to the complexity of these attacks – which involve several steps – and the high amount of successful APT campaigns perpetrated by malicious actors [71], it is crucial to understand what is the true scope and detection capabilities of the first line of defense; that is, existing intrusion detection systems. With respect to these, there are still some issues that need of further exploration as to develop effective tools capable of detecting, tracing and deterring APTs. It is thereby our mission in this thesis to explore the existing techniques and mechanisms that try to detect specific threat vectors within an industrial context, making emphasis on the special case of APTs but without losing sight of the future industrial paradigms.

This subject of research is particularly intense nowadays, and these safety issues have already been identified and considered by both the scientific and industrial communities. To begin with, there are numerous initiatives and working groups specifically focused on safeguarding the resources of production processes in the face of the advancements made by the digitization of industry. A notable example is the contractual PPP formed in 2016 between the European Commission and the European Cybersecurity Organization (ECSO), allocating €450 million with the aim of stimulating R&D work in this direction [72]. Among ECSO's efforts, Working Group 6 is in charge of managing cyber defense activities between the various cPPPs and the EU, particularly addressing issues related to the integration of Industry 4.0 in industrial environments, and their protection against advanced persistent threats. From a more technical perspective, the attack phases that may be involved in these threats are extensively studied by corporations such as IBM X-Force [73] or MITRE in their ATT&CK matrix (Adversarial Tactics, Techniques & Common Knowledge) [74]. Also, associations such as the European Factories of the Future Research Association (EFFRA), where funds are earmarked for the active integration of cybersecurity processes and practices in manufacturing environments, through projects and seminars, also in collaboration with ECSO, should also be taken into account.

On the other hand, ENISA is the European Union's Network and Information Security Agency, responsible for providing solutions and practical advice to the European public and private sectors, as well as publishing reports and studies on cybersecurity issues that contribute to the creation of new EU policies and legislation on network and information security. In terms of Industry 4.0, ENISA highlights in its latest report the need to develop solutions to solve the growing connectivity and complexity of production systems, ensuring the integration of IT/OT domains

and providing support for the incorporation of legacy systems in industry, among other needs that are satisfied with the development of this platform [75].

Regarding initiatives beyond European borders, in the United States there are several initiatives that provide support to the national network for industry innovation mentioned above. An example of these is the Department of Homeland Security, which through its strategic report ‘Strategic Principles for Securing the Internet of Things’ [76], and together with the ‘Cybersecurity Framework Manufacturing Profile’ defined by the National Institute of Standards and Technology (NIST) [77], pursues the establishment of security policies and controls applicable to any environment and industrial sector built under the umbrella of Industry 4.0. Aligned with this, the H2020 project AEGIS (accelerating EU-US dialogue in Cybersecurity and Privacy) has recently established in its ‘Policy Brief on Research and Innovation in Cybersecurity’ the five technological and application priorities in international cooperation in the coming years with the United States [78]. Among them, it is worth highlighting Cybersecurity in Industry 4.0, IoT and CPS, addressing issues related to trust, privacy and information security in all areas of application, whether in manufacturing, seaports or healthcare.

From an international perspective, there are also various standards that help organizations comply with security requirements and cope with future cyber threats scenarios. Specially, it is worth mentioning the IEC 62351 [79], a reference framework in the industry and power systems, that provides guidelines for introducing different security services concerning data and communications. Another example is the ISA/IEC 62443. These are a series of standards to provide a flexible framework that addresses and mitigates current and future threats and vulnerabilities. It has been developed by the ISA99 committee as American National Standards and adopted globally by the International Electrotechnical Commission [80].

On the whole, these standards and reference organizations are essential to pave the way for future cybersecurity services in the Industry 4.0. The specific features of these environments will bring new challenges that need to be understood and overcome when developing threat protection and detection mechanisms, which is one of the goals of this research.

### 1.3 Goals and Contributions of the Thesis

In the previous sections, we have supported the initial motivation of this thesis, where we have presented the challenges with respect to the detection and traceability of APTs in Industry 4.0 environments. This problem can be summed up in the absence of a single ‘silver bullet’ that can address all potential threats described in Section 1.2. Yet it might be possible to combine various solutions to provide an adequate level of protection against all kinds of attacks, including APTs.

When addressing these cybersecurity threats individually, the state of the art of cybersecurity for critical infrastructures shows that it is possible to detect threats against the availability of the system by detecting malicious network traffic and by mapping the behavior and location of existing devices. Likewise, there are other detection mechanisms that are specialized in the

detection of integrity threats: either directly, by detecting the presence of malicious entities, or indirectly, by uncovering the attacks and side effects caused by such entities. Finally, various techniques, such as in-depth traffic analysis, anomaly-based detection, and user monitoring can help in the detection of malicious insiders that bypassed the AAA (Authentication, Authorization and Accounting) infrastructure.

However, although the basic tools to detect and deter the attack vectors of an APT in a modern industrial ecosystem have already been developed, there are still some issues that need of further exploration. First, very few research works have made use of the existing research on APT behaviour [71, 74] to validate their detection mechanisms. Then, it is extremely important to facilitate the integration of holistic defense solutions in existing critical infrastructures, not only in terms of detection but also in terms of usability (e.g., availability of tools to facilitate the traceability of potential APT intrusions) and user training [81].

Based on this preliminary research, there are still certain aspects that require of more research and validation in the area of intrusion detection, attack traceability and intrusion prevention for the Industry 4.0. As discussed above, the integration of cutting-edge technologies such as IIoT and cloud computing must be carefully considered. Also, as the number of elements and business processes increases, the existence of misconfigured elements does so as well. Moreover, the opportunities for collaboration also increase the amount of information that is available to an adversary in case he/she controls a section of the system. Thus, it is essential to assure that all elements and evidence are properly monitored; making use, if possible, of lightweight accountability mechanisms based on granular information in which it is required to identify what, who and how these events were launched.

Therefore, we aim to approach the design of a framework for the detection and traceability of APTs in Industry 4.0 environments and applications. It is aimed to fill the gap between classic security mechanisms and APTs. The premise is to combine mechanisms capable of monitoring all the devices (whether physical or logical) that are interconnected within the organization, retrieve data about the production chain at all levels (e.g., alarms, network logs, raw traffic) and correlate events in a distributed way to trace the attack stages throughout its entire life cycle. These measures would provide the ability to holistically detect and anticipate attacks as well as failures in a timely and autonomous way, so as to deter the attack propagation and minimize its impact.

To cope with this cybersecurity goals, the aforementioned framework extracts the advancements of novel candidate solutions for IDSs in the Industry 4.0, such as the Opinion Dynamics [82]. These alternatives propose to apply advanced correlation algorithms that analyze an industrial network from a holistic point of view, leveraging data mining and machine learning mechanisms in a distributed fashion. Altogether, the framework serves as guidelines for the design and development of advanced detection systems that fulfill a set requirements for novel defense mechanisms in the Industry 4.0, namely:

- Coverage of all potential interactions and elements of an Industry 4.0 deployment, as well as the ability to be easily upgraded with new detection algorithms.
- Intelligence to take into consideration the existence of novel attacks and incorporate more advanced detection techniques such as behavioral analysis.
- Symbiosis with other protection mechanisms, such as prevention systems and authorization policies, but also with other relevant Industry 4.0 services, such as ‘digital twins’.

### 1.3.1 Thesis Outline

This chapter has introduced the main motivation and research scenario of this thesis, about the new cybersecurity challenges that appear on the Industry 4.0, and more specifically around the detection and traceability of APTs. In order to better understand such an scenario, we have reviewed the principal components introduced by this paradigm and introduced the main actors and stakeholders involved from an international perspective, and including the cybersecurity dimension. Based on the issues extracted, we determine that the goal of this thesis is to shed light to the problem of APT traceability. For this, we formalize a framework that enables the design and practical integration of such distributed mechanisms for the traceability of APTs, while also comparing the features of the aforementioned solutions according to the cybersecurity needs of the industry nowadays, both qualitatively and experimentally.

Before proceeding with the design of such framework, we review the main issues that menace current industrial architectures in Chapter 2. The goal is to characterize the context and create a taxonomy of attacks that may become part of an APT against current industrial assets and the future Industry 4.0 deployments.

To address these threats, we need to find what are the detection mechanisms that can be used as a first line of defense. For this purpose, in Chapter 3 we will provide an analysis of the evolution and applicability of IDSs that have been proposed in both the industry and academia. By this means, we can identify the areas that need of further research, regarding the applicability and integration of proactive detection mechanisms, and its integration with the advent of the Industry 4.0.

Based on the security and detection requirements extracted, Chapter 4 is devoted to defining the framework for developing solutions that enable the distributed correlation of APT events. This framework considers various network architectures, types of attack and data acquisition models, to later define the inputs and outputs that traceability solutions should include to support the aforementioned requirements. This lays the base for the development and comparison of novel solutions in this context. Indeed, as a means to validate the proposed framework, we define two novel protection mechanisms based on clustering and consensus, and carry out different experiments to compare their accuracy when tracing different APTs based on realistic attack models created from the analysis of threats conducted before.



In Chapter 5, we assess the accuracy of some response techniques that take advantage of the traceability features of the enabling correlation algorithms that meet our proposed framework. Similarly, we conduct a study on the feasibility of these detection systems in various Industry 4.0 scenarios, with the Smart Grid (to deploy mechanisms to ensure the security of the network and its authorization systems) and the Industrial Internet of Things being the most relevant.

In order to successfully validate all these findings, in Chapter 6 we perform the verification and validation of the framework defined, the correlation algorithms and the response techniques developed. On the one hand, this validation is conducted using theoretical demonstrations by elaborating the correctness proof of every approach presented. On the other hand, we also validate our detection approach from a practical point of view, by implementing a proof of concept of this approach in a real testbed, that integrates several kinds of industrial devices and protocols.

Finally, Chapter 7 summarizes the main contributions of this thesis and discusses some lines of future work and open research issues.

## 1.4 Publications and Funding

The main contributions of this thesis have been published in various journals and conferences, both national and international. Next, we provide a list of the main contributions organised by the type of publication:

### Journal articles ISI-JCR

1. Javier Lopez, Juan E. Rubio, and Cristina Alcaraz, *A Resilient Architecture for the Smart Grid*, IEEE Transactions on Industrial Informatics, vol. 14, issue 8, IEEE, pp. 3745-3753, 08/2019, 2018.
2. J. Lopez, and J. E. Rubio, *Access control for cyber-physical systems interconnected to the cloud*, Computer Networks, vol. 134, Elsevier, pp. 46 - 54, 2018.
3. Juan E. Rubio, Rodrigo Roman, Cristina Alcaraz, and Yan Zhang, *Tracking APTs in Industrial Ecosystems: A Proof of Concept*, Journal of Computer Security, vol. 27, issue 5, Elsevier, pp. 521-546, 09/2019.
4. Juan E. Rubio, Cristina Alcaraz, Rodrigo Roman, and Javier Lopez, *Current Cyber-Defense Trends in Industrial Control Systems*, Computers & Security Journal, vol. 87, Elsevier, 07/2019.
5. Juan E. Rubio, Rodrigo Roman, and Javier Lopez, *Integration of a Threat Traceability Solution in the Industrial Internet of Things*, IEEE Transactions on Industrial Informatics, vol. 16, issue 10, no. 6575-6583, IEEE, 10/2020.

6. Cristina Alcaraz, Juan E. Rubio, and Javier Lopez, *Blockchain-Assisted Access for Federated Smart Grid Domains: Coupling and Features*, Journal of Parallel and Distributed Computing, vol. 144, Elsevier, pp. 124-135, 06/2020.
7. Javier Lopez, Juan E. Rubio, and Cristina Alcaraz, *Digital Twins for Intelligent Authorization in the B5G-enabled Smart Grid*, IEEE Wireless Communications, vol. 28, issue 2, IEEE, pp. 48-55, 04/2021.

### International conference papers

1. Juan E. Rubio, Cristina Alcaraz, and Javier Lopez, *Preventing Advanced Persistent Threats in Complex Control Networks*, European Symposium on Research in Computer Security, vol. 10493, 22nd European Symposium on Research in Computer Security (ESORICS 2017), pp. 402-418, 09/2017.
2. Juan E. Rubio, Cristina Alcaraz, Rodrigo Roman, and Javier Lopez, *Analysis of Intrusion Detection Systems in Industrial Ecosystems*, 14th International Conference on Security and Cryptography (SECRYPT 2017), vol. 6, SciTePress, pp. 116-128, 2017.
3. Juan E. Rubio, Rodrigo Roman, Cristina Alcaraz, and Yan Zhang, *Tracking Advanced Persistent Threats in Critical Infrastructures through Opinion Dynamics*, European Symposium on Research in Computer Security (ESORICS 2018), vol. 11098, Springer, pp. 555-574, 08/2018.
4. Juan E. Rubio, Rodrigo Roman, and Javier Lopez, *Analysis of cybersecurity threats in Industry 4.0: The case of intrusion detection*, The 12th International Conference on Critical Information Infrastructures Security, vol. Lecture Notes in Computer Science, vol 10707, Springer, pp. 119-130, 08/2018.
5. A. Farao, et al., *SealedGRID: A Secure Interconnection of Technologies for Smart Grid Applications*, 14th International Conference on Critical Information Infrastructures Security (CRITIS 2019), vol. 11777, Springer, Cham, pp. 169-175, 12/2019.
6. Juan E. Rubio, Mark Manulis, Cristina Alcaraz, and Javier Lopez, *Enhancing Security and Dependability of Industrial Networks with Opinion Dynamics*, European Symposium on Research in Computer Security (ESORICS2019), vol. 11736, pp. 263-280, 09/2019.
7. Juan E. Rubio, Cristina Alcaraz, and Javier Lopez, *Game Theory-Based Approach for Defense against APTs*, 18th International Conference on Applied Cryptography and Network Security (ACNS'20), vol. 12147, Springer, pp. 297-320, 10/2020.
8. Juan E. Rubio, Cristina Alcaraz, Ruben Rios, Rodrigo Roman, and Javier Lopez, *Distributed Detection of APTs: Consensus vs. Clustering*, 25th European Symposium on Research in Computer Security (ESORICS 2020), vol. 12308, pp. 174-192, 09/2020.

### **National conference papers**

1. Cristina Alcaraz, Jesus Rodriguez, Rodrigo Roman, and Juan E. Rubio, *Estado y Evolución de la Detección de Intrusiones en los Sistemas Industriales*, III Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2017), 2017.

This thesis has been funded by the Spanish Ministry of Education under the FPU program (FPU15/03213). Some parts of this thesis have been partially supported by the following projects: SADCIP (RTC- 2016-4847-8), SealedGRID (H2020-MSCA-RISE-2017). Lastly, it is important to highlight that the visit to the Surrey Centre for Cybersecurity (University of Surrey) for three months has also been key for the continuity of this thesis.



## Chapter 2

# Cybersecurity in Industrial Ecosystems

To get into the problem of APT traceability, we firstly have to review the main cybersecurity issues that industrial architectures of the Industry 4.0 face nowadays. These can be classified into two types: intentional and unintentional threats. The former alludes to uncontrollable phenomena which, while not directly affecting control and automation systems, can jeopardize the production chain. For example, an incorrect use of USB keys may lead to a malware infection aiming to disrupt the correct functioning of the system [8]. The unintended ones, to the contrary, are the most influential in these types of critical contexts, and may come about from all those security gaps originating with the intentional threats. Outsiders, for example, could take certain remote actions, taking advantage of the architectural deficiencies of the information systems dedicated to the control to interfere with the operational infrastructures that support the manufacturing operations. The threat model, therefore, can be highly diverse where the attack vector and the adversarial influence are dependent on the nature and kind of attacker [83][84][85].

In this chapter, we firstly carry out a review of the threats affecting industrial systems today. Then, we analyze the cybersecurity issues presented by the technologies that enable Industry 4.0, both separately and integrated into the services offered by this paradigm. Finally, we study how these attack vectors can be part of the context of an advanced persistent attack, with the aim of exposing the challenges of current detection solutions.

### 2.1 Traditional Threats in IS and ICS

After several years of being subject to a multitude of threats [86], today's industry is still at risk. According to the annual reports of ICS-CERT [84], IBM® X-Force® Research [87], and Sikich [88], the number of threats has tended to rise annually in the manufacturing industry, either because of unforeseen occurrences or through planned actions. Irrespective of the causes, the consequences affect the normal performance of control and industrial process, thereby affecting the expected production rate and the final distribution to end-users. This situation is unfortunately aggravated when interconnecting traditional technologies and information systems to production

environments. For the purposes of our analysis, both types of attack vectors that affect the industrial environment (i.e., the ones inherited from the traditional industrial systems and those arisen with the interconnection of IT technologies) can be classified following the taxonomy given by the IETF standard RFC 7416 [89], in which the threats are grouped according to the attack goals against the minimum security services [90] such as availability, integrity, confidentiality and authentication.

**Availability threats:** apart from the typical subtraction of devices (e.g., PLC and RTU) or communication infrastructures, it is essential to highlight the threats related to distributed denial of services (DDoS) attacks, the techniques of which mainly focus on the routing (e.g., relay attacks, selective forwarding, grey hole, black hole or botnets).

**Integrity threats:** includes from the typical sabotage of the industrial equipment to the injection of malware [91] to slow down the operational performance, obtain sensitive information, modify the operation of the devices, etc. These threats are also related to the alteration of the industrial communication protocols and/or the real traffic values produced by field devices, controllers or corporate network equipment. Impersonation of nodes and spoofing are also applicable to an industrial context, due in part to the susceptibility to Man-in-the-Middle attacks and the existing weaknesses of the industrial communication protocols. We also have to consider that the vast majority of such protocols are still legacy protocols, in the sense that they were originally designed to transfer control information without considering various cybersecurity requirements such as authentication between peers, integrity of messages, or the confidentiality of the communication channels.

**Confidentiality threats:** within this category the illicit disclose techniques through passive traffic analysis (regarding topologies and routes) and theft of sensitive data (related to industrial process, customers, administration) or configurations should be highlighted. An example of information theft is that achieved by injecting code in the operational applications (often webs through cross-site scripting (XSS) or SQL Injection) so as to obtain or corrupt the control measurements/actions, the company and/or end-user privacy, or the security credentials.

**Authentication/authorization threats:** the authentication in this point includes those attackers that generally try to escalate privileges by taking advantage of a design flaw or vulnerability in the software in order to gain unauthorized access to protected resources. For example, according to the IBM® X-Force® research report [87], 45% of all attacks registered in 2015 focused on unauthorized accesses, followed by malicious code (29%) and sustained probe/scan (16%) attacks. In order to carry out these attacks, attackers need to apply specific social engineering techniques (e.g., phishing attacks, chain of spam letters) to collect strategic information from the system. Apart from this, the easy mobility of in-plant operators and their interactions through the use of hand-held interfaces (smart-phones, tablets, laptops) also lead to numerous security problems, probably caused by mis-configurations or unsuitable access control, both at the logical (use of simple passwords) and physical (access to equipment) level.

## 2.2 Landscape of Cybersecurity Threats of Industry 4.0 Enabling Technologies

Besides addressing the aforementioned security issues, it is necessary to envision a set of future security threats that might appear, especially pertinent when integrating new trending technologies such as IoT or cloud computing infrastructures. As explained earlier, these technologies are already being applied to ICS and herald the so-called fourth Industrial Revolution, or Industry 4.0 [92].

There are various researchers that have identified the most impactful threats that affect current industrial infrastructure ecosystems. Examples include social engineering, malware infection, compromising Internet-connected components, and insider threats [93]. Still, while these threats are also applicable to Industry 4.0 environments, it is necessary to understand the threats that might arise due to the integration of the enabling technologies introduced in Section 1.1. For this very purpose, this section will provide a taxonomy of such threats. The taxonomy described here has also been created according to the IETF standard RFC 7416 [89], that proposes an analysis of security issues whose classification is based on their effect on the main security services: availability, integrity, confidentiality and authentication. Nevertheless, it is important to note that many of the threats affect several of these services. An overall summary of the main threats of each technology, which have been extracted from the current literature, is presented in Table 2.1.

### 2.2.1 Industrial Internet of Things Threats

IoT interconnects sensors and all kinds of devices with Internet networks, to gather information about physical measures, location, images, etc. The IIoT specifically pursues a vertical integration among all the components that belong to the industrial architecture, ranging from machines to human operators or the product itself. With respect to security, the situation is further complicated when we take into consideration the scarce autonomy and computational resources that these devices have. Continuing with the IETF standard RFC 7416 [89], we can distinguish the following range of threats:

**Availability threats:** comprises the disruption of communication and processing resources: firstly, against the routing protocol [94], influencing its mode of operation (creating loops, modifying routes, generating errors, modifying message delays, etc.) through different attacks, which can be directly committed at the physical level through jamming or interferences. Secondly, against the equipment itself, including the exhaustion of resources (processing, memory or battery) exploitation of vulnerabilities in the software (as well as reverse engineering) that govern control devices such as PLCs, in addition to running malicious code or malware: viruses, Trojans, etc. [95]. Thirdly, we have to stress the data traffic disruption, undermining the functionality of the routers in the network, causing a lack of availability of certain services. It is caused by vectors such as selective forwarding, wormhole or sinkhole attacks.

**Integrity threats:** it means the manipulation of routing information to influence the traffic and fragment the network, like a Sybil attack [96]. This becomes the gateway to other attacks such as black hole or denial of service, causing the routes to pass through the more congested nodes. The form of attack includes falsification of information (the node advertises anomalous routes), routing information replay, physical compromise of the device or attacks on the DNS (Domain Name System) protocol [97]. Node identity misappropriation can also be taken into account, opening the door to other attacks that result in the modification of data of all types.

**Confidentiality threats:** includes the exposure of information of multiple kinds: firstly, the one related to the state of the nodes and their resources (available memory, battery, etc.). One way is the so-called side channel attacks [98], where the electromagnetic emanations of devices leak information about the execution of certain operations. Secondly, it also includes the exposure of routing information and the topology, which constitutes rich information for the attackers as it enables them to identify vulnerable equipment. Since this information resides locally in the devices, attacks against the confidentiality of this information will be directed at the device, either physically compromising it or via remote access. Lastly, it is also possible to have the exposure of private data, usually collected by wearable devices belonging to operators within the organization, which can reveal information about their performance at work or their location. One attack vector could be the use of social engineering or phishing.

**Authentication threats:** we can highlight the impersonation and introduction of dummy / fake nodes, capable of executing code or injecting illegitimate traffic to potentially control large areas of the network or perform eavesdropping. An attack vector consists of the forwarding of digital certificates used in authentication protocols or physical or network address spoofing. Escalation of privileges can also be faced as a consequence of a non-existent or poor access control, when the attacker can take advantage of design flaws or vulnerabilities in IoT devices to access protected resources without authorization.

## 2.2.2 Cloud Computing Threats

In recent years cloud computing has changed the way in which information technology is managed, through an environment that provides on-demand resources over the Internet with a low cost of investment and easy deployment. For our work, cloud computing acquires dual importance. On the one hand, many organizations use the cloud to provide IoT services, acquiring sensor data and sending commands to actuators. On the other hand, it is also necessary to take into account the delegation of certain analysis and production processes to the cloud, in what is known as cloud-based manufacturing [99]. The ultimate goal of this model is to enable customers to design, configure and manufacture a product through a shared network of suppliers throughout its life

cycle, enhancing the efficiency and reducing costs. In summary, these factors make it necessary to analyse the full range of threats that cloud computing faces [100][101]:

**Availability threats:** This category includes the so-called service theft attack, which takes advantage of the vulnerabilities and inaccuracies that exist in the scheduler component of some hypervisors, where the service is charged considering the time spent running virtual machines – instead of based on the CPU time in use. This can be exploited by attackers in order to use services at the expense of other clients, making sure that the processes of interest are not executed at each tick of the scheduler. We also contemplate denial of service attacks: the attacker causes the service to become inaccessible for its legitimate users. This is the most serious type of attack on cloud computing, because of the ease with which it can be carried out and the difficulties in preventing them.

**Integrity threats:** the most important one comes with a malware injection attack, where the attacker replicates the service instance that is provided to a client (a virtual machine, for example) and replaces it with a manipulated one that is hosted again in the cloud. This means that requests sent by the legitimate user are processed in the malicious service, and the attacker can access the exchanged data. To do this, the most common way is to appropriate access privileges or introduce malware into multiple format files, jeopardizing the confidentiality and privacy of the data.

**Confidentiality threats:** firstly, side-channel attacks with virtual machines must be stressed, in which the attacker, from his virtual machine, attacks others that are running on the same physical hardware. This allows them to access their resources by studying the electromagnetic emanations, the processor cache, etc. This information can be useful in choosing the most attractive targets to attack. This category also includes attacks on shared memory systems: they work as a gateway to other types of attacks such as malware or side-channel attacks, and consist in analyzing the shared memory (cache or main memory) used by virtual and physical machines to obtain technical information about the infrastructure, such as the processes that are running, the number of users, or even the memory dump of virtual machines.

**Authentication threats:** the attacker tries to obtain information from the clients of different applications or trusted companies by posing as themselves. This is done through malicious services with the same appearance as those are normally offered through a link sent by email. Thus, the attacker can obtain sensitive information from his/her victims by entering their data, such as passwords or bank cards. This way, the attacker can illicitly host services in the cloud and access accounts of certain services.

### 2.2.3 Big Data Threats

One problem that is closely related to cloud computing is that the data owner (i.e., the client) hardly has control of where the data is located. Nowadays, Big Data is used in the industry to process petabytes of information about their business, so it becomes critical to securely store and manage this bulk of data by means of preventive, detective and administrative mechanisms [102]. The full range of threats that data analytics faces is discussed in the following:

**Availability threats:** they revolve around the inability to use computational resources or access the information stored. Data is processed in a parallel way by a distributed network of nodes in charge of running MapReduce operations [103]. It is hence difficult to know where the computation takes place and equally tricky to ensure the security of all components (e.g., databases, computing power, etc.), so a small weakness can bring down the entire system. The data availability is also at risk if there is not any policy to create redundant copies of files.

**Integrity threats:** data processed is characterized by its volume (huge amount), velocity (speed of generation) and variety (multiple formats), so it is important to implement techniques to prevent against its modification, insertion, deletion or replay. Due to the distributed nature of Big Data, individual untrusted mappers (i.e., nodes in charge of acquiring and elementary processing pieces of data) can fail, resulting in a corruption of the aggregated data. Sometimes, such systems do not apply integrity measures in order to improve efficiency, which jeopardizes the veracity of the information. Data input validation is also essential to protect the information during its transmission from several sources (e.g., the corporate network, field devices, the web, etc.).

**Confidentiality threats:** they are a major concern in Big Data. From a technical perspective, the lack of real-time encryption over files and communications (usually to achieve performance) leave all the sensitive data exposed in case a vulnerability is exploited, whose impact is higher when all the data lies in distributed systems. On the other hand, Big Data also has privacy implications when data is analysed massively, which can draw accurate conclusions about the infrastructure or behaviour patterns of workers within the organization. As a consequence, there should be a balance between privacy and security, by strong policies and a new generation of encryption solutions.

**Authentication threats:** Big Data was designed for performance and scalability, without security in mind at the level of tables, rows and cells. The many data flows involved in the analytics make some companies deploy their private storage to hold the information. The problem arises with the unauthorized access to sensitive data (by both insiders or external attackers) spread over multiple nodes. Therefore, it is crucial to classify data based on its criticality and establish granular access controls for all systems and applications. It is

equally important to introduce a real time monitoring of devices, together with exhaustive logging procedures to keep track of any action taken upon data.

#### 2.2.4 Blockchain Threats

Distributed ledger solutions such as a blockchain can enable multiple business applications in the Industry 4.0. These range from events auditing to authorization applications or energy trading processes [104], by establishing a trusted network of peers. This is especially interesting for the decentralization pattern exhibited by modern industrial infrastructures. When it comes to data ownership and visibility, a Blockchain can be public (also known as permissionless) or private (permissioned). If we are dealing with a public model, then all parties are granted access to read past transactions. On the other hand, permissioned blockchain schemes oblige the partners to be identified and authorized prior to participating in network operations, which reduces dramatically the number of nodes compared to public blockchains [105]. The latter are based on a *Proof of Work* (PoW) consensus between the partners, such as the Nakamoto algorithm (where the consensus depends on demonstrating the resource consumption implied by solving a complex mathematical problem), since there is no previous trust assigned to the rest of peers within the network (that in turn offer a higher peer-to-peer scalability). In contrast to them, permissioned blockchains allow the deployment of more efficient consensus algorithms featuring a higher transaction capacity [106], which is essential not to impact the control performance. In the following, we classify the cybersecurity challenges that this technology faces:

**Availability threats:** the most critical threat against the availability of a PoW-based blockchain is posed by a so-called 51% attack. An adversary with high hashing power (i.e., having at least 51% of the hash rate) could potentially insert invalid transactions into a block and hence compromise the consensus protocol, gaining full control of the network and denying the service to specific peers. Likewise, although the P2P characteristics of this technology make it harder to disrupt than conventional architectures, DDoS attacks could be also feasible at a network level by flooding the nodes with junk data. This causes long delays when processing normal transactions, which would be critical in time-constrained environments. This attack was reported against the Bitcoin network in 2014, when attackers attempted to overflow it with requests [107].

**Integrity threats:** the integration of sequential hashing and cryptography, together with a decentralized structure, makes it harder to tamper with the information stored in a blockchain. The main integrity issue arises with smart contracts: a bug in such programs or a vulnerable development platform might end up with the theft of cryptocurrencies or injection of code, which could have a domino effect on other parts of the network or leave the ledger in an unpredictable state. For instance, in 2016 attackers managed to exploit a vulnerability in the Go-based Ethereum client's smart contract implementation that prevented peers from

mining further blocks [108], and another attacker exploited a bug in a smart contract that led to the theft of 60M Ether [109].

**Confidentiality threats:** blockchain built-in features already provide organizations with data immutability and traceability. However, this may precisely pose a problem in public blockchains when fitting with data privacy laws that oblige to implement the right to be forgotten and hence erase sensitive information. One solution is to encrypt the data written in the ledger or writing only the hash of transactions to it, while the transactions themselves are stored outside. In this case, this leaves space for a confidentiality issue if the security of the encryption key are compromised [110]. The injection of false data in the ledger can also take place in the presence of dishonest oracles (i.e., entities that connect a blockchain with off-chain data) [111]. In the case of private blockchains, it is fundamental to protect the network and data access by means of effective authentication and authorization controls.

**Authentication threats:** as mentioned before, private blockchains already offer out of the box full encryption and AAA capabilities to make data inaccessible by unauthorized parties. The main threat against the authentication property is represented by a wallet theft, this is, the stole of the private key through social engineering or the compromise of uncommunicative or intermittently active nodes within the blockchain. This may lead to Sybil attacks when the attacker gains control of multiple nodes and manipulates the consensus process [112].

### 2.2.5 Virtualization Threats

The growing amount of virtualization technologies to simulate the product creation and assist the workers in the process originates the need to create standards for the information exchange between the physical assets and their virtual representation, while achieving interoperability among all the interfaces. The secure integration of these services imposes several challenges:

**Availability threats:** the virtualization of actual components within the organization requires gathering, storing and processing data from all sensors installed in the production system and making it available to data consumers, mainly simulators and HMIs (e.g., augmented/virtual reality glasses, smartphones). This multiplicity of devices (each one with its own vulnerabilities) and platforms complicates the assurance of fault-tolerance and the realization of multi-platform user interfaces.

**Integrity threats:** the representation of the cyber-physical world for the purpose of parameterizing actual processes and monitoring their throughput implies the synchronization of coherent data among virtual and real endpoints. These parameters often concern control commands and 3D coordinates for simulation models, which are evolving with the production life cycle. It is therefore vital to safeguard the integrity of such data, since a

Table 2.1: Main Cybersecurity threats of Industry 4.0 enabling technologies

	<b>IIoT</b>	<b>Cloud/fog</b>	<b>Big Data</b>	<b>Virtualization</b>	<b>Blockchain</b>
Availability	Exhaustion of resources (traffic, requests)	Network flooding, service theft	Multiple points of failure	Multiple points of failure	Network flooding, consensus manipulation
Confidentiality	Exposure of sensitive information	Data access by the provider, side-channel attacks	Lack of cryptography, privacy issues when massively analyzing data	Simulations information leakage	Privacy issues with transactions traceable to users
Integrity	Data or routing information manipulation	Malicious VMs	Untrusted mappers, lack of integrity measures	Disparity between physical and virtual parameters	Vulnerable smart contracts, code injection
Authentication	Identity misappropriation	Phishing	Lack of fine-grain access controls to nodes and tables	Lack of AAA services to access data from heterogeneous devices	Identity or node theft, Sybil attacks

slight difference between models could lead in dysfunctions or incorrect predictions. It also demands an operational training for workers of the setup of complex machines and the associated simulation software.

**Confidentiality threats:** the increase of usability and accessibility of data for the operators involved in the industry contrasts with the need to keep the intellectual property safe from disclosure. Data used in simulations could also be leaked if the storage and memory of the systems in charge of executing such programs are not properly updated. In addition, privacy must be taken into account, since the mobility of workers should be tracked to provide them with current information on-site.

**Authentication threats:** the dissemination of information over multiple platforms and the virtualization of services (making use of cloud computing) blurs the barriers of data protection and eases its access by unauthorized entities, which is aggravated with the use of smartphones and similar devices that are easily breakable. It is thereby necessary to establish trust management procedures when sharing critical information, as well as strict control over data produced when a resource escalation or a new partner affiliation is performed.

## 2.3 Cybersecurity Threats in Industry 4.0 Innovative Services

In the previous section we have introduced the security threats that affect the main enabling technologies of Industry 4.0. Yet it is also vital to review what are the threats that could affect the most innovative services of this novel industrial ecosystem. The reason is simple: while these services inherit the threats of their enabling technologies, there are also various novel threats that arise due to their particular features. For this analysis, whose results have been obtained through an expert review of the available Industry 4.0 state of the art, we will continue following the IETF standard RFC 7416 [89]. We also provide an overall summary of the main threats of each service in Table 2.2.

**Novel infrastructures.** The gradual transition to more decentralized architectures shown in Section 1.1.2 is bringing a more heterogeneous and complex environment, where any element could (theoretically) interact and cooperate with any other element. Besides the potential dangers of unresponsive components, from the point of view of *availability* this transition means that not only a malicious insider could target any element, but also that a DoS attack could be launched from any element of the infrastructure. In terms of *integrity*, we need to consider that an adversary can alter the overall global behaviour (e.g., process workflows) by tampering with local decision makers. This is related to the *confidentiality* issues, where malicious attacks against local entities might expose high-level behaviour. Finally, regarding *authentication* threats, as the barriers between the different subsystems are blurred, it is necessary to deploy adequate security policies that can limit the damage caused by unauthorized accesses. However, the expected complexity of such policies will surely result on misconfigured systems, which can be exploited by adversaries.

**Retrofitting.** It is possible to bring the benefits of the Industry 4.0 to legacy systems by deploying and connecting new technologies to older subsystems [113]. Still, these deployments also bring additional security issues that need to be considered. The existence of a parallel subsystem (e.g., a monitoring system) might bring certain *availability* and *integrity* issues: not only the components that serve as the bridge between the old and the new can become a single point of failure, but also the new technologies could be used to launch attacks against the legacy elements. *Confidentiality* threats also exist, as the new technologies usually act as a “sensing layer” that can expose information about the status and behaviour of the monitored industrial processes. As for the impact of *authentication* threats, it mostly depends on the granularity of the integration of the novel subsystems: black-box interfaces limit the amount of information that can be retrieved from internal subcomponents.

**Industrial data space.** One of the goals the Industry 4.0 is to create common spaces for the secure exchange of information between industrial partners [114]. The creation of such cooperative spaces could bring additional threats from the point of view of *availability* and

*integrity*: the existence of DoS attacks that interrupt the information flow at critical times, or tainted components generating bogus data, will probably affect other elements – opening the door to potential cascade effects. *Confidentiality* is also especially important in this context: it is essential to assure that the information exchanged by partners does not facilitate the extraction of competitive intelligence. Still, misconfigurations and other internal attacks might open the door to more serious information leaks. *Authentication* threats are also aggravated in this cooperative space, as unauthorized accesses can have a wider impact in the extraction of valuable information.

**Cloud manufacturing.** One of the tenets of this paradigm is the creation of cloud-based industrial applications that take advantage of distributed manufacturing resources [115]. This distribution of resources creates certain threats that have been already described in the context of the novel digital architectures: from DoS attacks that can be launched from anywhere to anywhere (*Availability*), to the manipulation of the distributed components (*Integrity*). The main difference here is the nature of these threats, such as malicious VMs targeting the hypervisors, DoS against the cloud/fog servers or the network connection, etc. *Confidentiality* threats also become more critical, as the cloud infrastructure not only contains sensitive data, but also sensitive business processes as well. Finally, the complexity in the management of these kinds of cloud-based infrastructures also opens more opportunities for *authentication* attacks.

**Agents.** There are already various proof-of-concepts related to the integration of agents in manufacturing, such as workflow planners to self-organising assembly systems [116]. But there are dangers associated to the deployment of agents in an industrial environment, too. A malicious agent can behave like a piece of malware, affecting the *availability* of other industrial elements. Besides the *integrity* of the agents themselves, we also have to consider how other manipulated elements can exert a (in)direct influence over the behaviour of the agents. By tampering with the environment that surrounds the agent, or even the agent itself, it is possible to launch several *confidentiality* attacks that aim to extract the information flow that goes to the agent, and the information created by the agent itself. Finally, without a proper *authentication* infrastructure, malicious/manipulated agents will tamper with the overall workflow.

**Other enhanced interactions.** As aforementioned, Industry 4.0 enabling technologies such as virtualization allow the creation of novel services such as “digital twins” (virtual representations of subsystems) and “digital workers” (interaction with advanced HMI). Yet there are certain threats related to the actual usage of such technologies and services that need to be highlighted here. These enhanced systems can be manipulated by their human operators, effectively increasing the damage caused by an insider: a malicious digital worker could perform several attacks such as launching DoS attacks (*Availability*), interfering with the decision making processes (*Integrity*), extracting confidential information (*Confidentiality*), and executing privilege escalation attacks (*Authentication*). On the other hand, these enhanced systems can become attackers themselves,

Table 2.2: Main cybersecurity threats of Industry 4.0 innovative services

	<b>Dig. Arch.</b>	<b>Retrofitting</b>	<b>Data Space</b>	<b>Cloud Manuf.</b>	<b>Agents</b>	<b>Others</b>
Availability	Wide attack surface	Single point of failure	Cascade effects	Wide attack surface	Agents as malware	Denial of service
Confidentiality	Global data in local context	Exposure of sensing layer	Information leakage	Business processes leakage	Agent data in local context	Information leakage
Integrity	Behavior manipulation	Cross-cutting attacks	Cascade effects	Manipulation of components	Tampered data / agents	Disrupt decision making processes
Authentication	Complexity and Misconfiguration	Fake legacy / sensing layers	Bigger scope of attacks	Management issues	Attacks from/to agents	Privilege escalation

causing damage in subtle ways. For example, a malicious attacker could manipulate the HMI to force the worker to perform an incorrect action – and pin the blame on him.

Altogether, many of these attack vectors (from both traditional and future threats in industrial systems) are implemented in advanced persistent threats. This is a class of sophisticated attack perpetrated against a particular organization, where attackers have significant experience and resources. Such attackers infiltrate victim networks by taking advantage of a multitude of vulnerabilities (often unknown, i.e., zero-day), and go unnoticed for a prolonged period of time [69, 70]. Stuxnet was the first APT recognized by the industry in 2010 [8], but later many others have appeared, such as Duqu, DragonFly, BlackEnergy, and ExPetr [117, 71], as presented in the following.

## 2.4 Understanding Advanced Persistent Threats in Industry 4.0

The interconnection of industrial environments with modern ICT technologies has increased the number of internal and external threats in this context, including those from traditional IT systems (e.g., malware, spyware, and botnets). The APTs are a new class of sophisticated attacks that are executed by well-resourced adversaries over a long period of time. They usually go undetected because they leverage zero-day vulnerabilities and stealthy and evasive techniques [70]. While APTs originally attacked military organizations, they are now targeting a wide range of industries and governments with multiple purposes: economic (espionage, intellectual property), technical (access to source code), military (revealing information) or political (destabilization of a company) [118]. Their goal is to get through the organization network and take over the industrial control systems.

Stuxnet was the first attack of this kind, reported in 2010, which sabotaged the Iranian Nuclear Program by causing physical damage to the infrastructure and thereby slowed down the overall process for four years. Ever since, the number of reported vulnerabilities concerning Industrial Control Systems has been dramatically increasing, as the research community has

become more involved and new attacks have been revealed. In total, 3253 vulnerabilities have been reported by ICS-CERT between 2015 and 2019 (see Figure 2.1 showing this growth [119]).

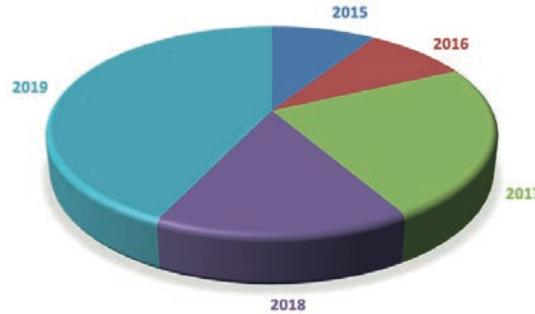


Figure 2.1: Reported vulnerabilities from ICS-CERT

As Stuxnet, every APT follows multiple steps, beginning with an initial intrusion, commonly using social engineering (e.g., by means of fraudulent e-mails containing Trojans). A successful intrusion results in the installation of a backdoor from which the attackers connect to the target network. Then, several exploits and malware are used to compromise as many computers in the victim network as possible (which is known as lateral movements), to ultimately modify the productive process or exfiltrate information back to the attacker domain. During the whole process, the threat actors make use of multiple tools to avoid detection and encrypt the external communication through publicly available services such as the Tor Anonymity Network [120].

On the whole, an APT is a meticulously planned attack adapted to the target infrastructure, one whose complexity makes the use of traditional countermeasures (e.g., antivirus, firewalls) insufficient to tackle them. Consequently, an additional effort is needed to mitigate the risks posed by these threats, which implies the effective detection of APTs through traditional countermeasures (e.g., intrusion detection systems, firewalls, antivirus) along with novel security services in continuous evolution within the company, involving all the organization with effective security awareness training and gaining knowledge from old use cases. Numerous surveys show the evolution of awareness about this field in the industry. Specifically, we can highlight the ISACA State of Cybersecurity 2020 report [121], that provides a view of the APT perception from security professionals belonging to many industries, mostly technology services, financial, military, telecommunications and manufacturing companies. Among all the statistics, it is worth commenting an increase in cybersecurity attacks in the 32% of the entities surveyed compared to 2019, where APTs are the second most common type of incidents after social engineering attacks. Additionally, 79% of respondents indicated some degree of likelihood that they would be attacked next year.

The industry as a whole is aware of the problems posed by APTs, and there are already various mechanisms that aim to facilitate their detection. Yet the solutions that are used in traditional

industrial control and automation systems are not directly applicable to Industry 4.0 contexts, as studied in the next chapter. The integration of Industry 4.0 principles, such as interoperability, decentralization, service-oriented management, and interactivity, are fundamentally changing all aspects of the industry: from the collaboration among supply chain partners, to the interactions between operators and machinery at the factory floor [122]. Yet it will also exacerbate the risks associated to APTs.

On the short term, industrial protocols like IO-Link and OPC UA will facilitate the interaction between existing and novel services. These and other technologies, like the Internet of Things, recognition services, and location services, will allow all individuals – from operators to administrators and executives – to access any relevant information anywhere at any time, helping them to make better decisions. Yet this interconnected ecosystem not only increases the attack surface, but also expands the influence that an APT can have in all actors once it has infiltrated into the system.

The deployment of open integrated factories and the integration of intelligent, dynamic processes are some of the medium and long-term goals of the Industry 4.0, respectively. Such goals will enable the creation of flexible workflows and production processes, the deployment of intelligent assistants using novel HMI interfaces (e.g., wearables, augmented reality), and the advent of novel services such as the “digital twins” (maintenance and management through simulation), amongst other benefits. Yet this flexibility and intelligence comes at a cost: APTs will be able to influence over the behavior of factory processes in subtler ways.

Moreover, we also have to consider how the Industry 4.0 and the Internet will be closely linked. Beyond the use of IoT devices, and the convergence of IT/OT infrastructures, there are novel approaches, such as cloud manufacturing, that will allow traditional manufacturing components to become virtualized and deployed in the cloud. These novel approaches will be surely become a target of APTs.

To put in place accurate defense techniques in this context, it is necessary to study how the precise attacks of an APT affect the detection of anomalies depending on their severity and the criticality of the victim nodes, which influences the application of traceability solutions. For this reason, in the following we review some of the most important APTs reported in recent years to define the attack and defense models.

### 2.4.1 Review of Reported Cases

For the specification of our APT traceability framework, we need to provide an accurate representation of APT attacks in the context of our network model. Therefore, here we firstly review the most important APT threats and groups that have specifically targeted industrial control systems. A more detailed review of these APTs – including exploited vulnerabilities, software modules, etc – is available at [71].

**Stuxnet (2009).** Stuxnet was one of the APTs that popularized this concept and brought it to the limelight. Developed by a state agent, the main goal of this worm was to hinder the enrichment of uranium in the Iranian nuclear facility of Natanz [123]. It is believed that its primary infection vector, which was used to infiltrate the facility, was USB flash drives. Once the malware was installed in the ‘patient zero’ computer, it also used other mechanisms (network shares, infected project files) to spread through the internal network, searching for the computers that directly controlled the uranium enriching centrifuges. Finally, the malware modified the code that controlled the centrifuges in order to silently destroy them.

**DragonFly group (2013-2014, 2015-).** Active since 2010, this particular APT actor has always focused on cyberespionage. On 2013, it started several campaigns against energy suppliers [124]. In its first wave of attacks, the main goal was to discover and map the existence of OPC SCADA servers located in the attacked network. For this purpose, after the initial infection, the malware queried the network in search of OPC servers using specific OPC DCOM (Distributed Component Object Model) calls. Its second wave of attacks followed a more conservative approach: it retrieved information mostly by extracting documents and screenshots from the infected computers.

**BlackEnergy (2015-2016).** The BlackEnergy malware, created by an APT actor known as Sandworm, was used to attack the energy infrastructure of Ukraine in December 2015 [125]. After the initial infection, the first goal of the malware was to replicate to as much computers as possible through Windows Admin Shares (e.g., through PsExec and remote file execution). The second goal of the malware was to set up various connections to external command&control networks. Using these networks, malicious operators were able to activate various components (KillDisk, circuit breaker manipulator) that caused havoc in electricity distribution companies.

**ExPetr (2017).** ExPetr was a wiper disguised as ransomware, which targeted local administrations and various industrial companies in Russia and Ukraine [126]. It used two primary infection vectors: a modified version of the EternalBlue exploit used by WannaCry, and a Trojanized version of the MEDoc tax accounting software. Once ‘patient zero’ was infected, this malware used both the EternalBlue exploit and the BlackEnergy propagation mechanisms to propagate over the local network. Immediately afterwards, the fake ransomware component of the malware would be activated.

**GreyEnergy (2018).** GreyEnergy is the name of the group behind the APT which is considered as the successor of BlackEnergy. It is believed to be active since 2015, targeting energy companies and other critical infrastructure organizations in Central and Eastern Europe [127]. GreyEnergy used more modern techniques than its predecessor, since the malware is built as a modular framework that can adjust to different target infrastructures, mostly for reconnaissance and information collection. Two infection vectors were used: compromising public-facing web servers

connected to the internal network, and sending spear phishing emails with malicious attachments. Then, the network mapping was performed and the malware was deployed.

**Zebrocy (2018).** This Trojan was developed by Sednit, a Russian-linked hacking group (also known as Sofacy) which is also allegedly associated to GreyEnergy. The infection took place using spear-phishing emails, and then a backdoor was installed on the victim computer to deploy further capabilities. Its targets were widely spread across the Middle East, Europe and Asia, and the first attacks were reported in Q3 2018. According to the first analysis, there are actually in the wild multiple versions of this Trojan that are implemented using multiple languages, in order to make them differ structurally and visually – and hence avoid their detection.

**NewsBeef (2019).** This is one of the aliases of the well-known APT33 Iranian group (also known as Charming Kitten or Elfin Team), who targeted multiple organizations of the petroleum and aviation industries in the Middle East, the US, and Asia. One of the attack waves was reported in February 2019, when they attempted to exploit a known vulnerability in compressed files, which were delivered via spear-phishing email. Once executed, the malware was able to download further commands and run additional malware. To prevent against its traceability, the group used several host layers to obfuscate its real command and control servers.

**Hexane (2019).** Associated with OilRig and CHRYSENE groups, this attack was reported in August 2019 and targeted the oil, gas and telecommunication sectors in Africa, the Middle East and Southwest Asia [128]. Its activity begun in September 2018, and their modus operandi was simple: firstly, password spraying and brute-force attacks were leveraged to compromise individual email accounts at the victim organization. Afterwards, those accounts were used to send phishing emails containing documents with macros as droppers (i.e., software that secretly installs malicious programs), following a trial-and-error process to search the best way to evade detection. Then, the attackers made use of RATs to run additional malware on the systems and finally conduct a DNS-based exfiltration to their command and control servers.

**WildPreassure (2020).** This is the name of a previously unknown APT campaign that distributed a Trojan called Milum written in C++. For their campaign infrastructure and to distribute the Trojan, the attackers rented OVH and Netzbetrieb servers and a domain registered with the Domains by Proxy anonymization service. Research studies concluded that this malware had been used since early 2019 to jeopardize industrial organizations in the Middle East. Among its features is the capability to control devices remotely as well as executing commands or collecting sensitive information to later send it to the attackers servers.

### 2.4.2 APTs Phases and Notations

Another element that is essential for the formalization of the behavior of APTs in our network model is the definition of the different attack stages (i.e., intrusion kill chains) that are performed by APTs. These attack stages have been extensively studied and described by various academic and industrial researchers [74, 70, 129], and can be summarized in the following steps:

- **Reconnaissance** (R). Adversaries gather information about the targeted industrial network to find exploitable vulnerabilities, and create an attacking plan to penetrate its defenses.
- **Delivery**. After choosing a set of vulnerable computers ('patient zero'), adversaries establish a communication (C) with the targeted industrial network and deliver the malware to those computers, either directly (e.g., through spear phishing emails or vulnerable services) or indirectly (e.g., contaminating websites of a third party such as a provider with malware) [130].
- **Compromise**. At this stage, the malware is executed (E) in the target machine and takes control of it, so that the first intrusion within the network is performed. This stage involves several steps, such as *privilege escalation*, maintaining *persistence*, and executing *defense evasion techniques*.
- **Command and Control**. Once the malware controls the 'patient zero', it opens a communication channel with the remote attacker by installing backdoors, which will be used to execute commands, extract information, etc. This phase may include the Tracking (T) of zero-day vulnerabilities based on the information collected by the adversary.
- **Lateral Movement**. The concept of lateral movement encompasses the different steps that the malware takes in order to achieve the propagation (P) of the attack to other areas of the network. Lateral movement includes *internal reconnaissance*, *compromise* of additional systems, and *collection of sensitive information*.
- **Final Execution** (F). The malware finally performs the attack against the targeted industrial network. These attacks include the *exfiltration* to send sensitive data back to the attacker domain (e.g., to later sell it in the black market) or the *destruction* of resources.

While this classification describes the most common attack path for industrial APTs, it is necessary to point out that not all APTs need to follow this particular template from beginning to end, or to implement all stages. For example, certain APTs only need to take control of a 'patient zero', and then they may proceed to extract sensitive information. Other APTs (like BlackEnergy) focus on creating a network of compromised nodes connected to the command&control centers, which allows malicious operators to cripple all the elements of the targeted network (both hardware and software) simultaneously.

A complete overview of the present and future threats faced by an Industrial System is summarized in Table 2.3, where all of them are linked with the APT stages introduced before. As discussed in [68], the exploitation of these threats may occur during multiple stages of an advanced persistent threat. More specifically, we can observe that most of these threats can be potentially leveraged for the first intrusion and the subsequent execution of exploits. However, the initial information gathering about points of entry and vulnerabilities is mainly performed by analyzing metadata emanated from servers to sensors, and also by social engineering. As for the final exfiltration of information, it normally requires that the attacker has taken over the device to send data such that it resembles normal network traffic, making any detection attempts challenging.

Even though most of these threats are in general inherited by IoT and cloud technologies, they also pose new hazards to be addressed. Firstly, because the technical constraints that the new devices and communication protocols feature create new vulnerabilities and attack vectors. Secondly, due to the impact they cause in the assets within the organization, which comprise control and corporate resources as well as end-users (e.g., clients or operators). Altogether, this makes it necessary to find new defense solutions and tailor the current detection mechanisms, as discussed in the following.

Table 2.3: Overview of threats that affect industrial systems

Threats	Traditional	IIoT	Cloud Comp.	Big Data	Block.	Virtual.	APT-states	Impact on				
								Control in-plant	Corp. Net.	End users		
<b>Availability</b>												
Subtraction of devices	✓	✓				✓	E	✓				
DDoS attacks	✓	✓	✓	✓	✓	✓	C, E, P	✓				✓
Attacks on-path	✓	✓			✓		C, E, T, F, P	✓				✓
Exhaustion of node resources	✓	✓			✓		C, E	✓				✓
Service theft			✓	✓	✓	✓	C, E	✓				✓
<b>Integrity</b>												
Incorrect configuration	✓	✓	✓	✓	✓	✓	C, E	✓				✓
Reverse engineering and/or malware injection	✓	✓	✓	✓	✓	✓	R, C, P, E, T, F	✓				✓
False data injection	✓	✓		✓	✓	✓	C, E, P	✓				✓
Spoofing	✓	✓		✓	✓	✓	C, E	✓				✓
Manipulation of routing information	✓	✓		✓	✓	✓	C, E, P	✓				✓
<b>Confidentiality</b>												
Sensitive information theft	✓	✓	✓	✓	✓	✓	C, E, F	✓				✓
Nodes status exposure (side-channel attacks)		✓	✓			✓	R, C, E, F	✓				✓
Passive traffic analysis	✓	✓		✓		✓	R, C, E, T, F, P	✓				✓
Infrastructure information exposure (shared memory systems attacks)			✓		✓	✓	C, E, T, F, P	✓				✓
<b>AAA</b>												
Privilege escalation	✓	✓	✓	✓	✓	✓	C, E, P	✓				✓
Social engineering	✓		✓	✓	✓	✓	R, C, E	✓				✓
Deficient control access	✓	✓	✓	✓	✓	✓	C, E	✓				✓
Impersonation of nodes (fake/dummy nodes)	✓	✓		✓	✓	✓	C, E	✓				✓



## Chapter 3

# Detection of APTs in Industry 4.0: State of the Art

After reviewing the surface of attacks that an APT can leverage in its sequence of stages, we are now in position to study what are the detection mechanisms that can be leveraged as a first line of defense. The goal of this chapter is to analyze the IDSs available commercially and in the literature of ICS protection, to identify the areas that need of further research in the specific area of Industry 4.0. Based on the knowledge extracted, we will be able to define an APT traceability framework, which represents the core of this dissertation.

Part of the analysis carried out in this chapter summarizes the contributions of SADCIP [131], an research project funded by the Spanish Ministry of Economy, Industry and Competitiveness. It revolves around the provisioning and development of advanced detection systems capable of dealing with sophisticated threats in the context of modern industrial ecosystems, and considering the specific characteristics of Industry 4.0.

### 3.1 Classification of Intrusion Detection Systems

Due to the variety of attack vectors that an APT exposes, multiple security solutions must be combined at different levels. In this sense, intrusion detection systems pose the first line of defense, as they detect unauthorized access to the network or one of its systems, monitoring its resources and the traffic generated in search of behaviors that violate the security policy established in the production process.

There are many methods for performing intrusion detection. One possibility is the *signature-based IDS*, which tries to find specific patterns in the frames transmitted by the network. However, it is precisely for that reason that it is impossible for them to detect new types of attacks whose pattern is unknown [132].

Another possibility is the *anomaly-based IDS*, which compares the current state of the system and its generated data with the normal behavior of the system, to identify deviations

present when an intrusion occurs. However, in the context of control systems, restrictions such as the heterogeneity of the data collected in an industrial environment, the noise present in the measurements, and the nature of the anomalies (attacks vs. faults) must be taken into consideration.

For this reason, numerous detection techniques have been based on areas such as statistics or artificial intelligence [133], each with a different level of adaptation depending on the scenario of the application to be protected [134]:

**Data mining-based detection:** based on the analysis of an enormous amount of information in search of characteristics that enable distinguishing if the data is anomalous. In this category we find:

- *Classification techniques:* creation of a mathematical model that classifies data instances into two classes: ‘normal’ or ‘anomalous’. This model is trained with already classified example data.
- *Clustering-based techniques:* like the previous category, they seek to classify instances of data but in different groups or clusters, according to their similarity. This is mathematically represented by the distance in the space between the points associated with that information.
- *Association rule learning-based techniques:* they process the data set to identify relationships between variables, in order to predict the occurrence of anomalies based on the presence of certain data.

**Statistical anomaly detection:** in this approach, inference tests are applied to verify whether a piece of data conforms or not to a given statistical model, in order to confirm the existence of intrusions:

- *Parametric and nonparametric-based methods:* while the former are those that assume the presence of a probability distribution that fits the input data to estimate the associated parameters (which does not have to conform to reality), the second tries to look for the underlying distribution. In general, both are accurate and noise-tolerant models of missing data, which allow us to find confidence intervals to probabilistically determine when an anomaly occurs.
- *Time series analysis:* they predict the behavior of the system by representing the information it generates in the form of a series of points measured at regular intervals of time. Although they are able to detect slight disturbances in the short term, they are less accurate in predicting drastic changes.
- *Markov chains:* they consist of mathematical representations to predict the future behavior of the system according to its current state. For this purpose, state machines

are used with a probability associated with transitions. Its accuracy increases when using complex multi-dimensional models.

- *Information based techniques*: they involve the observation of the information generated (for example, the capture of the traffic) and its intrinsic characteristics in search of irregularities associated with threats, packages for denial of service, messages to cause attacks by buffer overflow, etc. They are generally efficient systems tolerant to changes and redundancy in the information.
- *Spectral theory-based techniques*: these techniques use approximations of the data to other dimensional sub-spaces where the differences between the normal and the anomalous values are evidenced. They are usually complex and are used to detect stealth attacks, those which are specially designed to circumvent detection techniques.

**Knowledge-based detection:** in this case, the knowledge about specific attacks or vulnerabilities is acquired progressively, ensuring a low rate of false positives, thereby resulting in a system that is resistant to long-term threats. However, the security depends on how often the knowledge base is updated, and the granularity with which information about new threats is specified. Examples of these techniques include state *transition-based* techniques, *Petri nets* or *expert systems*.

**Machine learning-based detection:** this type of technique bases the detection on the creation of a mathematical model that learns and improves its accuracy over time, as it acquires information about the system to be protected. In this category we find techniques of artificial intelligence whose foundations are also closely linked to statistics and data mining:

- *Artificial neural networks*: they are inspired by the human brain and are able to detect anomalies when dealing with a large data set with interdependencies. It allows the data to be classified as normal or anomalous with great precision and speed, although they need a long time to create the model, which prevents them from being applied in real-time systems.
- *Bayesian networks*: events are represented in a probabilistic way through directed acyclic graphs where the nodes represent states and the edges define the conditional dependencies between them. The purpose is to calculate the probability of an intrusion from the data collected.
- *Support vector machines*: this is a technique that classifies the data according to a hyperplane that separates both classes (habitual and anomalous information). Since it works with a linear combination of points in space (given by the input data), its complexity is not high and its quality of precision is acceptable. However, it does not behave accurately in presence of similar data, for which there is no hyperplane that divides them correctly.

- *Fuzzy logic*: rule-based structures are used to define a reasoning with inaccurately expressed information, like humans do in everyday language (being able to differentiate when a person is ‘tall’ or ‘short’ or something is ‘slightly cold’). Therefore it models the behavior of complex systems without excessive accuracy (leading to speed and flexibility), but obviously it means the accuracy of the anomaly detection is not high either.
- *Genetic algorithms*: they simulate the phenomenon of natural selection to solve a complex problem for which there is no clear solution. In the first phase, a set of individuals of a population is randomly generated (representing the possible solutions to that problem). From there, numerous iterations are carried out where successive operations of selection, replacement, mutation and crossing are applied to ultimately find an optimal solution. Although it is moderately applicable to the detection of anomalies, it has been shown that it is unable to detect unknown attacks.

On the other hand, there are also *specification-based IDSs* [135]. The principle behind them is similar to systems based on anomalies, in the sense that the current state of the system is compared to an existing model. However, in this case the specifications are defined by experts, which reduces the number of false positives to the extent that they are defined in detail. State diagrams, finite automata, formal methods, etc. are often used. They are often combined with *signature-based* and *anomaly-based* IDSs.

One alternative to IDS solutions are precisely Intrusion Prevention Systems (IPS). These systems have the ability to (i) detect an anomaly within the system and (ii) mitigate the effect of the threat. Cubix’s TippingPoint [136] is a clear example of IPS capable of detecting traffic anomalies in VoIP infrastructures, routers and switches. Similarly, Extreme networks IPS also ensures business continuity by monitoring the behavior and state of the operating systems such as Windows [137]; and Corero Network Security offers in-line intrusion detection and automated response by combining behavior-based and signature-based analysis [138].

However, the inclusion of these systems within complex infrastructures of critical nature is not always feasible. The automation of response actions implies that we need to trust in the reliability and accuracy of such actions; yet, depending on the situation, it is very probable that the actions may not be so suitable for a critical context [139]. In addition, the false positive rates in the detection processes can also significantly impact on the final response – and indirectly affect the performance of the critical control systems [140]. These characteristics are widely reflected in the state of the art, where there are multiple approaches and researches in the field and for general contexts [141, 142], but not enough for critical contexts.

As specified in [143, 144], it is essential to provide customizable IPSs for critical environments, or at least for those remote areas where no human operator with reactive capacity is available – either remotely or on-site. This work evidently involves more research in the area, since it is essential to find the sequences of parameters and actions that best suit a situation, searching the

way to offer proactive measures that help respond to incidents or threats before major disruptions may arise [143]. This protection property was also referenced by the NIST in [145].

Even though IDS (and IPS) represent a valid solution to address the first stages of an APT, it becomes essential for security staff to introduce additional techniques and procedures to guarantee a minimum impact on the infrastructure [70]. Some of them can be summarized as follows:

- **Advanced detection of malware:** for instance, the execution of processes and files from suspicious provenance in sandbox mode, or the on-line malware analysis, in a non-intrusive way.
- **Data loss prevention:** as the last line of defense, this software protects against the breach of data by controlling access and use of sensitive information.
- **Whitelisting:** since the intruder intends to connect to an external server to set up a command and control service and ultimately filtrate some data, a countermeasure to prevent it is required. In this case, it would consist in the use of access control policies for the inbound and outbound connections (e.g., specifying the exclusive set of URLs that each device can access).
- **Trusted computing:** a secure environment is created by means of hardware modules that guarantee the integrity and reliability of the software that is installed and used within the industrial system. In this case, aspects of TPM (Trusted Platform Modules) [146] or TEE (Trusted Execution Environments) [147] should be contemplated.
- **Intelligence-driven Defense:** based on the knowledge provided by experts and victims of APTs, an intelligence feedback loop is created to identify patterns of intrusions and understand the adversaries' techniques, in order to accurately design and implement proper countermeasures.
- **Security awareness training:** training and consciousness about the best security practices becomes especially important to protect against APTs, since most intrusions are performed with the use of social engineering techniques.

So as to give a more detailed vision of actual technologies that make use of these and other mechanisms, a review of the state of the art of defense solutions in both the industry and academia is given in the remaining sections.

## 3.2 Academic Research

As it is crucial to protect industrial control infrastructures against all kinds of attacks, including advanced persistent threats, the scientific community has paid special attention to the development of intrusion detection systems for this particular context. In these systems, all the defense

Coverage	2013	2014	2015	2016	2017	2018	2019	2020
Field devices	2	-	3	15	9	8	10	6
Control networks – PLCs	4	8	9	5	9	9	10	5
Control networks	1	3	3	9	17	12	18	11
Complete system	-	1	-	5	2	6	9	7

Table 3.1: Evolution according to detection coverage

Protocol	2013	2014	2015	2016	2017	2018	2019	2020
Fieldbus protocols	2	1	2	3	2	2	4	2
Communication protocols	2	3	10	14	8	8	9	7
Control & management protocols	1	-	1	1	1	2	3	2

Table 3.2: Evolution according to protocol analyzed

Mechanism	2013	2014	2015	2016	2017	2018	2019	2020
Signature-based detection	-	3	-	4	5	6	4	5
Data mining mechanisms	2	2	4	5	6	7	10	6
Statistical anomaly detection	-	-	4	5	3	2	4	3
Knowledge based detection	1	1	2	1	-	4	5	4
Machine learning based detection	3	3	2	8	9	9	11	13
Specification-based detection	1	3	2	8	10	4	7	4
Other mechanisms	-	-	3	5	5	4	9	7

Table 3.3: Evolution according to detection mechanism

mechanisms described in Section 3.1 have been integrated to some extent, trying to cover all the elements of an industrial control network: field devices, the interactions between the control network and field controllers such as PLCs, the control network itself, and even the complete system in a holistic way.

Tables 3.1, 3.2 and 3.3 provide a classification by categories (according to detection coverage, protocol analyzed, and detection mechanism, respectively) of the number of articles published in the field between years 2013 and 2020. Within this classification, we have included the most relevant articles that appeared in international journals and/or conferences. This relevance has been measured by factors such as the relevance of the corresponding journal or conference, and the number of references per article.

### 3.2.1 Analysis: Detection Mechanisms

In recent years, all detection mechanisms described in Section 3.1 have been taken into account. We can observe in Table 3.3 that research in the field has been growing over time. We can also observe that the academia has been paying special attention to machine learning mechanisms. Still, the importance of signature and specification-based detection techniques remains high. One possible reason is that the elements of the control networks can behave in a more or less

predictable way [148]. As such, these elements can be modelled through various sets of rules and anticipate well-known attacks between the corporate network and the control network.

There are certain detection strategies, which will be highlighted here, that are still being studied only within the academia. For example, in the last years, several authors have started analyzing parameters such as industrial telemetry and response time. Mainly due to the behaviour of control networks, these parameters are providing novel and exciting insights over the behaviour of such control systems. For example, through indirect or direct analysis (e.g., via ICMP messages) of these parameters, it is possible to detect variations in the traffic patterns that are indicative of ongoing attacks [149], detect fake control devices [150], discover covert manipulations of the controller device code [151], and even deduce the CPU load of PLCs [152]. There are also researchers who have considered other less traditional parameters within the context of anomaly and intrusion detection, such as the radio-frequency emissions emitted by the control devices [153], or even their power consumption [154].

There are also other researchers that incorporate concepts such as the physical simulation of the monitored system [155]. This simulation allows not only to predict the malicious intent of a command, but also to predict an imminent system failure. In addition, within the context of specification-based research, there are a large number of publications that seek to generate the system behavior rules in an automatic or semi-automatic way. Various works, such as [156] [157], retrieve this information by analyzing the configuration and system description files. Other approaches, such as [158], extract the system states by analyzing the bursts of traffic that are exchanged between the control network and the PLCs.

Another recurring trend in recent years is to design hybrid mechanisms, combining more than one detection technique. This is particularly useful in application scenarios where data from the control network must be processed before conducting anomaly detection. For instance, [159] proposes a privacy-preserving method that filters sensitive data from power systems using statistical approaches that perturbate the information prior to applying Gaussian mixture models. Another example is provided in [160], where the authors propose a two-stage IDS that firstly executes Ethernet/IP traffic prediction with time series forecasting and then applies a one class support vector machine to detect malicious control instructions.

Besides, there are also other strategies whose goal is to identify and analyze the most critical elements of a control network. An example of this is the system developed by Cheminod et al. [161], which can identify the sequence of vulnerabilities that could affect an existing system by (i) analyzing the elements of that system and (ii) analyzing vulnerability databases such as CVE (Common Vulnerabilities and Exposures) [162]. Other research lines provide a support to the aforementioned IDS/IPS technologies from a theoretical perspective, adopting a reactive policy by means of recovery mechanisms when topological changes are detected. Their target is to ensure the structural controllability of the network and achieve resilience [163], this is, the continuity of the industrial process and the connectivity between nodes in presence of attacks [164]. For such goal, graph theory concepts are leveraged. Finally, it should be mentioned that the vast majority

of new signature-based detection systems use, in addition to the SNORT tool [165], the BRO [166] tool and the SURICATA [167] tool to perform their analyses. These new tools are used because they provide additional benefits. For example, the BRO tool provides a modular and extensible framework that allows the generation and analysis of events through a Turing-complete language.

### 3.2.2 Analysis: Detection Coverage

Regarding the evolution of the coverage of detection systems developed in the academia, it is worth commenting that in 2016 the mechanisms in charge of protecting the field devices increased exponentially, and is still a very active area of research as of 2020. The reason is simple: these mechanisms can detect attacks against the field devices at the very moment they occur, making them a very useful last line of defense against APTs that aim to manipulate the field devices. Direct monitoring is usually done by extracting the data directly from the sensors and actuators, either through the machine's own interfaces [168] [169], or through a 'capillary network' that monitors the operation of the machinery through several types of external sensors [170][171]. On the other hand, there are also mechanisms that integrate a hypervisor within the control devices themselves (e.g., PLCs [172]). This hypervisor is then responsible for reviewing the behavior of all control programs executed within the device, either through a set of rules [173][174] or by modeling the different states of the program and checking for potential deviations [175].

Moreover, starting from 2016, various researchers have designed novel theoretical architectures whose objective is to protect all the elements of an industrial production system in a holistic way. This is achieved by deploying various detection components, both hardware and software, which obtain information and process it at a local level. This information will then be sent to a central system, which can more efficiently detect threats that affect several elements of the system in a covert way [176]. For example, some architectures allow field devices to be fully monitored alongside all other elements of the control system [170], while other architectures improve the detection of anomalies whose impact is distributed to all elements of the system [177]. There are also architectures, such as [178][179], that divide the overall system into several logical partitions, in order to facilitate the work of anomaly detection systems. Finally, some architectures deploy host agents that are specifically designed to look for APT malware infections [180]. these correlation systems can serve as a great inspiration for the development of holistic detection techniques and the traceability of APTs, which is the goal of the framework proposed in this thesis.

### 3.2.3 Analysis: Protocols Analyzed

Currently there are various scientific articles that have developed specific detection mechanisms for communication protocols such as Modbus/TCP [181], Ethernet/IP [167] and S7comm [182]. These works focus mostly on two strategies: (i) defining and detecting attack signatures, and (ii) analyzing the behavior of these communication protocols with the detection mechanisms

described in Section 3.1. However, there are very few works that have studied the security of control & management protocols such as OPC UA, although its interest is expected to rise. These protocols are considered as one of the cornerstones of Industry 4.0 [183], and there are already various commercial products that currently use these protocols in production environments [184]. Yet the amount of research that has been done in this area has been limited, and only a few works exist [185][186]. It is extremely important to analyze and protect these specific protocols in the near future.

Another important aspect related to the communication protocols is that many detection mechanisms that analyze the integrity of fieldbus protocols are focused on the analysis of wireless industrial IoT protocols such as WirelessHART [187] or Zigbee [188]. This is mainly because an attacker can more easily manipulate a wireless network if he or she has the necessary information. Namely, he or she can not only inject information from anywhere within the range of the network, but he or she can also deploy a malicious element in a covert way. Finally, it is important to note that there have been multiple developments in the area of anomaly detection systems for certain industry-specific protocols, such as CAN bus (vehicular systems) [189] and IEC 61850 (electrical substations) [190].

### 3.3 Industrial IDS Products

Defense Strategies	Leading Companies
Zone-based	<i>Advenica, ARGUS, BAE Systems, Bayshore, Checkpoint, Deep Secure, Distrix, Fortinet, Fox-IT, Icon Labs, Intel, Moxa, Nexor, Paloalto Networks, Phoenix Contact, Positive Technologies, Seclab, Sophos, Tofino Security, Towersec, Waterfall Security</i>
Configuration-based	<i>Verve, PAS, Nextnine, DL2C, AlgoSec, Sigmaflow, Dragos Security, Amenaza Tech. LTD, Positive Technologies</i>
Signature-based	<i>Cisco, Cyberark, Cyberbit, Digital Bond, ECI, FireEye</i>
Context-based	<i>AlertEnterprise, WurldTech (GE)</i>
Honey-pot-based	<i>Attivo Networks</i>
Anomaly-based	<i>Control-See, CritiFence, CyberX, Darktrace, HALO Digital, ICS2, Indegy, Leidos Nation-E, Nozomi, Claroty, PFP Cybersecurity, RadiFlow, SCADAfence, SecureNok, Sentryo, SIGA, ThetaRay</i>

Table 3.4: Leading companies in the market

At present, there are various commercial solutions whose goal is to provide protection mechanisms that can deter the attacks caused by APT actors. Such protection mechanisms not only include the detection mechanisms described in Section 3.1, but also other solutions such as enhancing user awareness, separating the industrial network into various protected zones, and analyzing the configuration of the system. Most of these solutions are passive (i.e., do not affect the operation of the system), transparent (i.e., almost invisible to the existing control systems), and easy to deploy.

Table 3.4 provides an enumeration of the leading companies in the market that provide such protection mechanisms. In addition, a short summary of the main solutions available in the market as of 2020 is provided in the next sections.

### 3.3.1 Zone Separation

These products focus on facilitating the separation of the industrial network into different security zones, using traditional security solutions such as firewalls. The main challenge here is the structure of industrial networks. Due to their complexity, it is necessary to consider the deployment of various zones, such as the enterprise systems (e.g., ERP), the enterprise middleware (e.g., message oriented middleware, enterprise service bus), the industrial control systems and the field device networks, and the different demilitarized zones.

Beyond the integration of traditional firewall solutions that focus on IT networks and protocols, there are various companies that provide specific solutions designed for industrial networks. One example is the FortiGate platform developed by FortiNet [191], which has the capacity to analyze multiple industrial protocols (e.g., Bacnet, DLMS, DNP3, EtherCAT, ICCP, IEC-60870.5.104, Modbus/TCP, OPC, Profinet) and industrial devices (eg ABB, Rockwell, Schneider Electric, Siemens, or Yokogawa). It is also important to note that, due to the manufacturing of extremely complex interconnected systems such as smart cars, there are now specific firewalls that are designed to protect these products beyond the assembly line, like the Harman Shield solution by Harman [192].

On the other hand, there are several commercial products focused on controlling and filtering the information exchanged between zones. Various platforms, such as Advenica ZoneGuard [193], provide a bridge between IT and OT networks that implement various information exchange policies. Other solutions, including Data Loss Prevention [194] and Nexor Border Gateway [195], also allow the definition of policies for certain network interactions, such as outbound connections and inbound email messages, respectively.

Besides, certain products implement the ‘data diode’ communication approach, which physically enforces a one-way flow of data. Some solutions, like Fox DataDiode [196], focus on the integration of these diodes between IT and OT zones. Other solutions, like SecuriCDS Data Diode [193], also implement additional defense mechanisms (e.g., dual power supplies) that avoid the creation of covert data channels. Finally, there are some approaches, like Waterfall FLIP [197], that actually implement reversible diodes, which can be activated by personnel on-site in case of emergencies.

### 3.3.2 Secure Configuration

There are various products in the market whose goal is to provide a holistic view of the configuration of the overall system. For example, platforms like the ICS Shield platform developed by Nextnine [198] focus on providing a centralized operations center for the management of

various security aspects of the system. They include the automatic discovery and classification of the system assets, the retrieval of hardware/software state information and the management of changes in this state, the management of passwords, the secure transfer of data, the management of software updates and backups, the creation and application of security policies, and the preparation of security reports, amongst others.

Other platforms revolve around the analysis of the system configuration, so as to manage and verify existing security policies. For example, the AlgoSec Security Management Solution [199] not only proactively assess existing network security policies related to firewalls and cloud access, but also is able to intelligently design policy changes and implement them whenever necessary. Continuing with the subject of verification, certain tools, such as NERC Compliance by Sigmaflow [200], provide automated compliance monitoring of existing security and reliability industrial standards. These tools not only analyze the documentation of the company in search of discrepancies with existing standards, but also validate certain compliance data in real time, such as security controls, local accounts, and logical access rights.

Finally, there are platforms whose goal is to analyze the configuration and the elements of the system in search of vulnerabilities. Some vulnerability assessment systems, such as MaxPatrol, are specifically designed for industrial settings. Due to their design, these tools can efficiently analyze the system without interrupting its regular use, and are able to monitor even ERP systems such as SAP [201]. On the other hand, there are some tools, such as SecurITree, that focus on the theoretical analysis of attack models and attack trees [202]. These tools can create reports that predict the most likely behaviour of attackers, and can help to identify risks that are otherwise undetected.

#### 3.3.3 Signature-based Solutions

These products consist mainly of devices that passively connect to the control network, accessing the information flow. One of the pioneers in this field is Cisco Systems, which has a large database of attack signatures on industrial environments [203]. Such attack signatures include not only generic attacks on elements of the industrial network (e.g., denial of service in HMIs, buffer overflows in PLCs), but also specific vulnerabilities in industrial protocols (e.g., CIP Or Modbus). This database is easily upgradeable, and can be integrated into all Cisco intrusion detection systems.

There are also other products on the market that, beyond the detection of attack signatures, provide several value-added services. An example of this is the monitoring system of Cyberbit [204]. This system monitors the traffic of the network in order to map existing devices, giving the operator a real-time view of the elements of a system. In addition, it is possible to take advantage of information acquired from the device to identify elements that have known vulnerabilities.

### 3.3.4 Context-based Mechanisms

One drawback of most products based on the detection of attack signatures and patterns is the lack of correlation between the detected events, which could provide valuable information regarding the actual scope of the attack behind those events. Another drawback is the absence of an in-depth analysis based on the context of the system: the parameters of a command can be valid in a given context, but harmful in another. As a consequence, there are several products that perform correlation and/or in-depth analysis tasks which take into account the general context of the system.

One example of these correlation systems is the Sentry Cyber SCADA software from AlertEnterprise [205]. It combines and correlates events and alerts from various domains (physical, IT and OT networks) and sources, with the aim of providing a complete security monitoring tool for industrial systems. To achieve this objective, this solution allows integration with other security tools, such as vulnerability scanners, SIEM (Security Information and Event Management) systems, IDS/IPS systems or security configuration tools.

Finally, an example of in-depth analysis solutions is Wurdtech's OPShield [206] system. OPShield performs an in-depth analysis of the network traffic, including the syntactic and grammatical structure of the protocols. Through these analyses, OPShield can inspect the commands and parameters sent to the different components of the industrial system, and even block those commands if the administrator has authorized OPShield to do so. Note that the blocking or not of these commands is determined based on the context in which they have been sent. Thus, it is possible to protect the system against seemingly valid and/or legitimate commands that are potentially dangerous for the correct operation of the system if they are sent outside the context for which they were defined.

### 3.3.5 Honeypot-based Techniques

Existing solutions based on honeypot systems usually create a distributed system, through which they collect and analyze information related to the threat or attack. Thanks to the analysis and correlation of the collected information, this type of IDS / IPS systems can be able to identify the type of attack launched, the (malicious) activities carried out on the system, as well as the existence of infected devices.

Within the current marketplace, one of the major existing honeypot-based detection platforms is ThreatMatrix from Attica Networks, which is able to detect real-time intrusions in public and private networks, ICS/SCADA systems, and even IoT environments. Its flagship product is called BOTsink [207], and is able to detect advanced persistent threats effectively, without being detected by the attackers. The client also can customize the software images that simulate SCADA devices. Such customization allows the integration of both the software and the protocols that are used in the production environment. As a result, fake SCADA devices can be made almost indistinguishable from real SCADA devices.

### 3.3.6 Anomaly-based Solutions

As of 2020, there are a wide range of products that make use of deep packet inspection and/or machine learning technologies to detect unusual behaviors or hidden attacks, of which there is no already identified pattern. Such products are usually deployed as rack servers, although many companies also provide virtualized solutions. Regarding the deployment location of these commercial products, most of them operate on the operational network, accessing the information flow through the SPAN ports of existing network devices. Other deployment strategies exist, though. Some products, such as UCME-OPC from Control-See [208], retrieve system information directly from the industrial process management layers. Other products make use of agents that are distributed throughout all the elements – devices and networks – of the industrial system. Finally, there are products in charge of monitoring the interactions with field devices, such as those offered by SIGA [209]; or even systems embedded within the field devices themselves, such as those offered by MSi [210], which are responsible for examining and validating the behavior of field devices.

As for the specific techniques of anomaly modeling and detection, each commercial product makes use of one or several of them. Some products, such as UCME-OPC from Control-See [208], create a model of the system based on certain conditions/rules. Whenever those rules are not fulfilled by the system parameters and values, a warning will be launched. Other products, such as XSense from CyberX [211], base their operation on the classification of system states: if a monitored system transitions to a previously unknown state, such state is classified as normal or malicious depending on multiple signals and indicators. There are also products, such as HALO Vision from HALO Analytics [212], which make use of statistical analysis.

Other products consider industrial control systems from a holistic point of view, and include the behavior of various actors, including human operators, into their own detection systems. For example, Darktrace's Enterprise Immune System [213] makes use of a variety of mathematical engines, including Bayesian estimates, to generate behavioral models of people, devices, and even the business as a whole. There are also other products, such as Wisdom ITI from Leidos [214], which offer a pro-active and real-time platform for internal threat detection. This platform not only monitors system activity indicators, but also the behavior of human employees. Another example of this is the Privilege Account Security Solution by CyberArk [215], which monitors user activity to detect not only anomalous activity caused by abuse of existing privileges, but also potential symptoms of compromised credentials.

Finally, it is necessary to point out that the majority of these products start with no knowledge about the environment or industrial system that they aim to protect. As such, they need to be trained, acquiring the knowledge they need mostly by monitoring the network traffic. Even so, there are some products, like the suites marketed by ICS2 [216] or the products developed by ThetaRay [217], that can acquire such behavior offline. For example, by loading and processing training files, or by retrieving information provided by the manufacturer about the expected

behaviour of the different system components. The aim of this is to reduce the time required for the deployment and commissioning of these products.

### 3.4 Current Industry 4.0 Detection and Traceability Solutions

As with traditional IT systems, Industry 4.0 deployments can be attacked by malicious adversaries, which could generate serious operation disruptions in critical infrastructures. In this context, IDSs become a necessary defense layer to detect potential attacks against these infrastructures. Even if the field of IDSs for Industry 4.0 technologies is not as developed as the field of IDSs for traditional industrial ecosystems addressed before (cf. [218]), there is still a plethora of detection approaches [67]. Some of these detection mechanisms focus on the integration of *signature-based IDSs* and Deep Package Inspection (DPI) technologies [219], which try to find specific patterns in the network frames. Other *anomaly detection systems* implement various machine learning techniques, aiming to detect instances of data (exchanged from IIoT devices) that do not belong to a learned class (i.e., a model that has been trained and validated).

Besides, there are several IDSs specifically designed for Industry 4.0. and IIoT deployments that benefit from the unique characteristics of industrial networks (e.g., deterministic operation procedures) compared to general IT networks [220]. According to the state of the art (cf. [221]), these intrusion detection procedures mainly focus on the analysis of the communication patterns and the protocols states to identify a deviation from a previously created specification. This leads to two main detection strategies: *specification based anomaly detection* and *physical state dynamic estimation*.

In the first strategy, *specification based IDSs*, human experts build a model that describes the legitimate system behavior (e.g., protocols, programs, operations) to latter compare it with the current state to detect anomalies. Some examples of this approach include [222], where an advanced metering infrastructure is modelled to represent a legitimate activity profile at various levels, and [223], where the specification is at protocol-level to model the Modbus TCP communication patterns. The second strategy, *physical state dynamic estimation*, complements the first strategy by modeling the physical dynamics of the operations performed in the production chain. For example, in [224], the authors propose a resilience framework for cyber-physical systems which permits to describe physical domains mathematically. Other examples include [225] and [226], which models the physical constraints of a power grid infrastructure and a water distribution network, respectively.

Regardless of the detection strategy used in the industrial premises, IDSs only pose a first line of defense, and further post-incident analysis of the generated evidences (e.g., alarms, network events) and raw traffic must be conducted all across the network to anticipate the effects of sophisticated and persistent attacks such as the APTs [227]. This is carried out by traceability and advanced correlation mechanisms, which provide information of the overall network health status and facilitate the deployment of accurate response measures based on the threat evolution.



This has been mostly addressed in traditional corporate environments, by means of proactive techniques (evidences are analyzed as incidents occur) and reactive techniques (evidences are studied once the events occur). Among the proactive techniques, [228] proposes a framework for flow-based analysis of network traffic in near real time to detect APTs in cloud computing. Also, in [229], researchers present a security framework for the analysis of high volumes of traffic to identify data exfiltrations and suspicious activities in TCP/IP networks. Some other approaches conduct advanced analytics with the outputs of external IDSs. For example, in [230], researchers propose an approach entitled TerminAPTor, a theoretical supervision system capable of linking multiple information flows from classical IDSs. In [231], the authors propose MLAPT, a machine learning-based system to detect and predict APT attacks by correlating the outputs of different detection methods. As the rest of approaches, it is experimentally validated in a corporate infrastructure (using a dataset of attack scenarios against a campus network).

As for industrial ecosystems, traceability solutions are provided by means of context-awareness solutions [232]. This process involves the monitoring of the physical devices that are interconnected by a communication infrastructure, to retrieve data about the production chain at all levels (e.g., alarms, network events, raw traffic). However, the introduction of increasingly dynamic topologies and the growing range of extremely localized attacks in the IIoT and Industry 4.0 complicate the process of information acquisition [233]. Therefore, it is important for industrial ecosystems to set up more than one detection solution to ensure the maximum detection coverage [218]. Moreover, all solutions should coexist with advanced detection platforms that take the infrastructure from a holistic perspective, correlate all events and track all threats throughout their entire life cycle [234]. This holistic perspective is even more necessary in light of the existence of APTs. In this sense, there is also the need to investigate holistic models of local and global information that are capable of anticipating, detecting and responding to failures and attacks at all times and autonomously. This implies the deployment of specialized techniques to prevent the extension of security problems to other areas of the system and minimize their impact. These are also known as situational awareness solutions [235].

In summary, to the best of our knowledge, all existing traceability approaches are designed for generic IT networks, and have not explicitly discussed how they could be implemented and validated using real attacks. Therefore, as the progress in the Industry 4.0 has not been significant with respect to actual APT traceability solutions, it is the main motivation of this work to provide a first step in this area. In this sense, the Opinion Dynamics approach [82] paves the way for a new generation of solutions based on the deployment of distributed detection agents across the network. The anomalies reported by these agents are correlated to extract conclusions about the sequence of actions performed by the adversary, and also to identify the more affected areas of the infrastructure. Such assessment can be conducted in a centralized entity or using a distributed architecture of peers [236]. At the same time, it is open to integrate external IDSs to examine anomalies in the vicinity of nodes, as well as the abstraction of diverse parameters such as the criticality of resources or the persistence of attacks.

Despite the many capabilities of this solution (explained in further sections), it is necessary to define a more general detection model to lay the base for the precise application of more APT traceability solutions in the Industry 4.0 paradigm. The reason is that the Opinion Dynamics capabilities can be implemented modularly, they can be integrated into other correlation algorithms and each one has a different effect on many security, detection, deployment and efficiency constraints. These points will be addressed in the next section, where we define the security and detection requirements involved, to later present the traceability framework in Chapter 4.

### 3.5 Detection and Security Requirements for the Industry 4.0

The analyses performed in the previous section have shown that Industry 4.0 threats are inherently more complex than the threats that target traditional industrial environments. Since networks and interactions are no longer compartmentalized, the attack surface increases – not only in terms of vulnerable entities, but also in terms of potential attackers and attack strategies (e.g., behavioral attacks). Besides, as the number of elements and business processes increases, the existence of misconfigured elements does so as well. Moreover, the opportunities for collaboration also increase the amount of information that is available to an adversary in case he/she controls a section of the system. These threats have considerable influence on how IDSs must be designed, deployed and managed in these kinds of contexts. Given the threats described in the previous sections, an IDS should comply with several requirements that are described below. They can be classified into detection and security concerns.

- (D<sub>1</sub>) **Coverage.** APTs make use of an extensive set of attack vectors that jeopardize organizations at all levels. Therefore, the system must be able to assimilate traffic and data from heterogeneous devices and sections of the network, while also incorporating the input of external detection systems.
- (D<sub>2</sub>) **Holism.** In order to identify anomalous behaviors, the system must be able to process all the interactions between users, processes and outputs generated, as well as logs. This allows to generate anomaly and traceability reports at multiple levels (e.g., per application, device or portion of the network, as well as global health indicators).
- (D<sub>3</sub>) **Intelligence.** Beyond merely detecting anomalous events within the network in a timely manner, the system must infer knowledge (by correlating current events with past stages) and should anticipate future movements of the attacker. Similarly, it should provide mechanisms to integrate information from external sources – that is, cyber threat intelligence [237].
- (D<sub>4</sub>) **Symbiosis.** The system should have the capability to offer its detection feedback to other Industry 4.0 services, by means of well-defined interfaces. This includes access control mechanisms (to adapt the authorization policies depending on the security state of the

resources) or virtualization services (that permit to simulate response techniques under different scenarios without interfering the real setup), among others.

On the other hand, we can also establish the following security requirements with regards to the deployment of the detection solution over the network:

- (S<sub>1</sub>) **Distributed data recollection.** It is necessary to find distributed mechanisms – such as local agents collaborating in a peer-to-peer fashion – that allow the collection and analysis of information as close as possible to field devices. The ultimate aim is to make the detection system completely autonomous and resistant to targeted attacks.
- (S<sub>2</sub>) **Immutability.** The devised solution must be resistant to modifications of the detection data at all levels, including the reliability and veracity of data exchanged between agents (e.g., through trust levels that weigh the received security information), and the storage of such data (e.g., through unalterable storage mediums and data replication mechanisms such as immutable databases or distributed ledgers).
- (S<sub>3</sub>) **Data confidentiality.** Apart from the protection against data modification, it is mandatory that the system provides authorization and cryptographic mechanisms to control the access to the information generated by the detection platform and all the interactions monitored.
- (S<sub>4</sub>) **Survivability.** Not only the system must properly function even with the presence of accidental or deliberate faults in the industrial infrastructure, but also the system itself cannot be used as a point of attack. To achieve this, the detection mechanisms must be deployed in a separated network that can only retrieve information from the industrial infrastructure.
- (S<sub>5</sub>) **Real-time performance.** The system must not introduce operational delays on the industrial infrastructure, and its algorithms should not impose a high complexity to ensure the generation of real-time detection information. Network segmentation procedures and separate computation nodes (e.g., fog/edge computing nodes) can be used for this purpose.

Notice that these requirements are also desirable for traditional industrial ecosystems, yet such requirements are very difficult to enforce in those contexts – mainly due to the inherent industrial features and necessary trade-offs (e.g., avoid false alarms that can put the production line in jeopardy, minimize the impact of the IDS components in the operational network, etc. [238]). Still, the cooperative, dynamic and complex nature of Industry 4.0 ecosystems requires that IDSs subsystems must interact more closely with the industrial components, in order to detect attacks before their impact becomes too severe.

Understandably, and also due to the specific features of industrial ecosystems, the actual state of the art on IDSs for the current industrial ecosystems (cf. [238]) do not fully cover the previously mentioned requirements. Besides, there are few or no components that search for anomalies in

the behavior of Industry 4.0 essential protocols, such as OPC UA; and the concepts of symbiosis and exchange of security information in this context are still in its infancy.

As for the creation of IDS mechanisms for the industry of the future, there is no need to start from zero. There are various elements in the state of the art that can be adapted and/or enhanced to fulfill the previously presented requirements. For example, there are various platforms that provide event correlation and knowledge extraction from a holistic perspective, although most of such platforms are based on a more centralized architecture, as studied before. Precisely, there are also agent-based architectures that validate the behavior of the monitored systems [239].

Moreover, there are preliminary works that could serve as a foundation for the more advanced features required by Industry 4.0 IDSs, such as the dynamic deployment of honeypots adapted to the requirements of the system, the automatic identification of critical elements, and the interaction with physical simulation systems in order to detect anomalies [155]. Based on these premises, we aim to define a detection and traceability framework that eases the development of appropriate solutions for the Industry 4.0 context, as explained in the following chapter.

## Chapter 4

# Detection and Traceability Solutions based on Distributed Correlation

Based on the security and detection requirements extracted previously, this chapter is devoted to defining the framework for developing solutions that allow the distributed correlation of APT events. In the first place, we introduce some preliminary concepts about structural controllability and graph theory that are necessary to define the aforementioned framework. Based upon these, the APT traceability framework is presented, by specifying its infrastructure model along with its inputs and outputs. Then, to illustrate the feasibility and effectiveness of this framework, we identify correlation algorithms that satisfy its specification. Lastly, we carry out a qualitative and quantitative comparison of those approaches, prior to experimentally applying them to Industry 4.0 scenarios in the next chapter.

### 4.1 Modelling Industry 4.0 Networks Using Graph Theory

In this section, we lay the theoretical base that permits the formal representation of Industry 4.0 infrastructures and actual APT attacks over a defined network, as well as the mathematical background of the detection techniques presented in this chapter.

#### 4.1.1 Structural Controllability

Considering the cost of the implementation of large control networks from a research point of view, it becomes mandatory to model and simulate the problem through graph theory, taking into account the network topology and the nature of its distribution. With the purpose of helping the reader understand the underlying theoretical concepts of the attack and detection models, topics related to structural controllability and power dominance are described here. The concept of structural controllability was introduced by Lin in 1974 [163], which associates the control of a network to a subset of nodes with the maximum capacity of dominance.

Let  $G = (V, E)$  be a *directed cyclic* graph that represents the network topology, given by its adjacency matrix, that is, a square binary matrix  $M$  with dimension  $|V|$  where  $M(i, j) = 1$  whenever  $(v_i, v_j) \in E$  and zero otherwise. Through  $G(V, E)$ , it is possible to characterize dynamic control networks including loops and weighted edges that represent the interconnection of control devices with field devices (e.g., sensors or actuators) to issue control commands and retrieve data. These links contain the maximum capacity to conduct the main traffic between two points, which is defined as the *control load capacity* (CLC).

To represent this traffic, we use the *betweenness centrality* (BC) [240], that gives an idea of the connectivity that every node or edge experiences. It is an indicator that represents the sum of the fraction of the shortest paths that pass through a given edge, so that edges with the highest centrality participate in a large number of shortest paths. The result is a weighted matrix related to  $G(V, E)$  whose weights are computed as follows:

$$BC(v) = \sum_{s,t \in V} \frac{\delta(s, t | v)}{\delta(s, t)} \quad (4.1)$$

where  $\delta(s, t)$  denotes the number of shortest (s,t)-paths and  $\delta(s, t|e)$  the number of paths passing through the node  $v$ . On the other hand, let the in-neighborhood  $N_i^{in}$  of a node  $i$  be the set of nodes  $v_j$ , such that  $(v_j, v_i) \in E$ ; while the out-neighborhood  $N_i^{out}$  is the set of nodes  $v_j$  such that  $(v_i, v_j) \in E$ . Consequently, let the in-degree  $d_i^{in}$  of a node  $v_i$  be the number of its incoming edges, i.e.,  $d_i^{in} = |N_i^{in}|$ , while the out-degree  $d_i^{out}$  is the sum of its outgoing edges, i.e.,  $d_i = |N_i^{out}|$ .

---

**Algorithm 1** DS( $G(V, E)$ )

---

**output** ( $DS = \{v_i, \dots, v_k\}$  where  $0 \leq i \leq |V|$ )  
**local:**  $BC(V)$  representing betweenness centrality of  $V$

Choose  $v \in V$  with highest BC  
 $DS \leftarrow \{v\}$  and  $N(DS) \leftarrow \{v_i, \dots, v_k\} \forall i \leq j \leq k \setminus (v, v_j) \in E$   
**while**  $V - (DS \cup N(DS)) \neq \emptyset$  **do**  
    Choose vertex  $w \in V - (DS \cup N(DS))$  with highest BC  
     $DS \leftarrow DS \cup \{w\}$   
     $N(DS) \leftarrow N(DS) \cup \{v_i, \dots, v_k\}$  where  $\forall i \leq j \leq k \setminus (w, v_j) \in E$   
**end while**

---

Taking these concepts and BC into account, the Dominating Set (DS) of a graph  $G$  can be defined as the minimum subset of nodes  $D \subseteq V$  such that for each vertex  $v_i \notin D$  is adjacent to at least one member of  $D$ , that is  $\exists v_k \in D | (v_k, v_i) \in E$ . These nodes  $D$  with the highest control capacity will be those with the highest edge *betweenness centrality*  $BC(v)$  for all their outgoing edges. The creation of this set is explained in Algorithm 1. Related to this concept, the Power Dominating Set (PDS) consists in an extension of the DS by including new *driver nodes* (denoted by  $N_D$ ), those with the maximum capacity of dominance within the network. Even though we



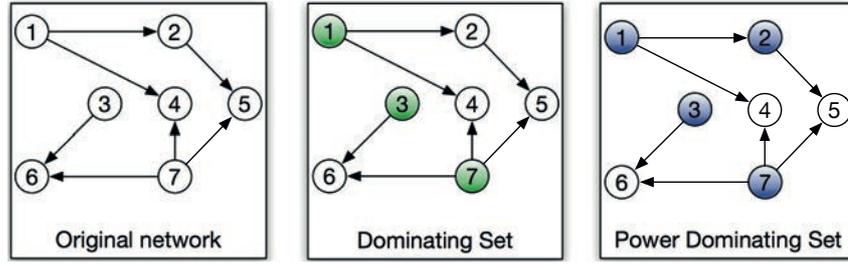


Figure 4.1: Observation rules for the election of the most dominating nodes

consider these sets for controllability purposes, it is important to note that they can be treated as observation rules. Their original formulation was given by Haynes *et al.* in [241], and was later simplified into two fundamental observation rules by Kneis *et al.* in [242]:

- OR1** A driver node,  $n_d$  in  $\mathbf{D}_N$ , observes itself and all its neighbors (i.e., the rest of nodes that share a communication link with  $n_d$ ), which conforms the DOMINATING SET of nodes. This implies that every node not in  $\mathbf{D}_N$  is adjacent to at least one member of  $\mathbf{D}_N$ .
- OR2** If an observed vertex  $v$  of degree  $d^+ \geq 2$  is adjacent to  $d-1$  observed vertices, the remaining unobserved vertex becomes observed as well. This also implies that **OR1**  $\subseteq$  **OR2** given that the subset of nodes that comply with **OR1** becomes part of the set of nodes that complies with **OR2**, which conforms the POWER DOMINATING SET.

An example of the election of these driver nodes is depicted in Figure 4.1. As explained later, for the purpose of threat detection, the dominating nodes can play the role of agents that detect topological changes in their surroundings that may be derived from an APT attack, and potentially apply response techniques or establish backup links that ensure the continuity of the network. All the three concepts introduced in this section and related to the network represented with  $G = (V, E)$  are summarized in Table 4.1 for future reference in the following sections.

We now aim to extend the graph  $G = (V, E)$  to characterize its topology according to current industrial standards. As discussed in Chapter 1, most industrial ecosystems are nowadays adopting cutting-edge technologies into their production chain and monitoring systems. The counterpart of the modernization of industrial technologies (which we have referred to as ‘operational technologies’ or OT) and its integration of IT (‘information technology’) in this context comes with the appearance of new cybersecurity threats, as studied in Chapter 2. Some of them are inherited from the IT paradigm and some other arise from the growing integration between IT and OT. We are talking about attack vectors such as denial of service, presence of malware in the control teams, exploitation of vulnerabilities in communication protocols, phishing and social engineering, etc. For this reason, since there are several reported APTs that attempt to compromise resources belonging to both the IT and OT parts of the industrial network, it makes sense that the whole industrial topology can be split into these different sections: IT and OT, which will be interconnected by firewalls.

Table 4.1: Summary of structural controllability concepts

Term	Concept	Definition
BC	Betweenness centrality	holds the connectivity degree of a node or edge within the network
DS	Dominating Set	Minimum subset of nodes within a graph that are adjacent to the rest, complying with the <b>OR1</b> observation rule
PDS	Power Dominating Set	Minimum subset of nodes within a graph that are adjacent to the rest of nodes and edges, complying with the <b>OR2</b> observation rule

Traditionally, the architecture of a typical control network has adopted the ISA-95 standard [36], as stated in the introduction of this thesis. Following a rigid pyramidal architecture, the manufacturing components (i.e., sensors and actuators) are located at the base (level 0), whereas devices interacting with them (i.e., PLCs, RTUs) are set at level 1. Level 2 comprises those devices that control the production process (i.e., SCADAs, HMIs), while those that manage the workflow (i.e., MES) belong to level 3. Finally, the highest level contains the ERP or resource management. However, due to the aforementioned integration of cyber-physical systems, this architecture is evolving towards a distributed and decentralized model. Therefore, the lines that separate every level are getting blurred, which is more noticeable in the highest level of the IT section, where several entities (e.g., ERP, SCADA systems) can be flexibly deployed in the cloud, as shown in Figure 4.2.

Due to this evolution of industrial topologies, the formalization of the proposed network architecture using the graph  $G(V, E)$  can be further extended. We can assume this network is composed by the IT and OT sections, which are respectively represented with subgraphs  $G(V_{IT}, E_{IT})$  and  $G(V_{OT}, E_{OT})$ . These sections are joined by a set of firewalls placed in between ( $V_{FW}$  henceforth), so that  $V = V_{IT} \cup V_{OT} \cup V_{FW}$ .

In order to understand how these network sections are merged, we have to recall the DS and PDS subsets explained before. In particular, the PDS will be used in the OT section of the industrial topology to represent the set of devices that are connected to the firewalls that also connect to the IT nodes, thereby merging both sections. The reason for such election is that multiple kinds of devices coexist in an operational environment. However, apart from sensors and actuators, PLCs and HMIs, only SCADA systems and high-level servers are actually connected to external networks (i.e., the IT section or Internet). Therefore, these last nodes are the ones that

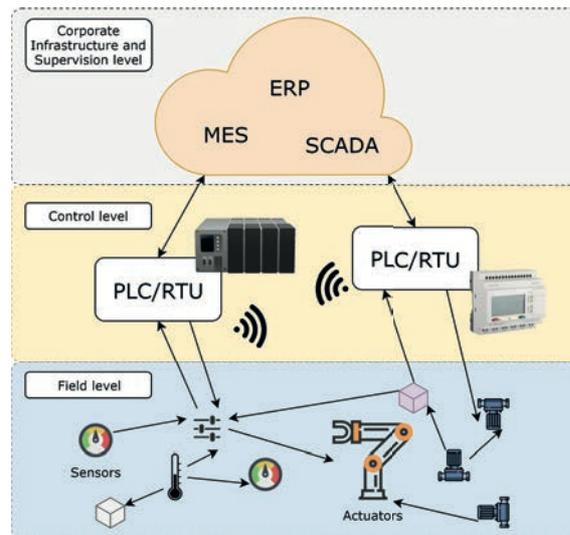


Figure 4.2: Architecture of modern industrial organizations

hierarchically have more connectivity (and therefore they will be linked to the firewall nodes), which is equivalent to the controllability concept introduced before. As for the IT section, since most of the devices range from ERP to customer-end systems (whose computational capabilities are not as restricted as OT devices), we assume that all nodes are connected to the firewalls and thereby can access the operational area.

#### 4.1.2 Topology Generators

Concerning the network topology of the IT and OT section, we must note that each of these subnetworks (represented with subgraphs  $G(V_{IT}, E_{IT})$  and  $G(V_{OT}, E_{OT})$ ) is built with a different network distribution. There are multiple topology generators available in the research community whose function is to generate random networks for specific studies (e.g., routing protocols, network recovery), imitate the hierarchical nature of real networks or reproduce their degree properties. In this work, certain models have been studied and selected for their ease in the implementation and their realism to produce replicas of real infrastructures. More specifically, we will make use of different models depending on the network subsection or the characterization of an industrial sector with particular connections or architectures (e.g., the IIoT).

For the moment and to lay the base of topology generators used in our theoretical simulations, in this section we focus on the topology generators leveraged in a generic Industry 4.0 scenario, for the two subsections introduced before. On the one hand,  $G(V_{OT}, E_{OT})$  follows a specific network construction centered on power-law distributions of type  $y \propto x^{-\alpha}$ , which is extensively used to model the topological hierarchy of an electric power grid and their monitoring systems [243]. These networks are also known as *scale-free* and commonly contain substations, which are nodes with high degree (i.e., the number of edges incident on the node) connected to nodes with lower

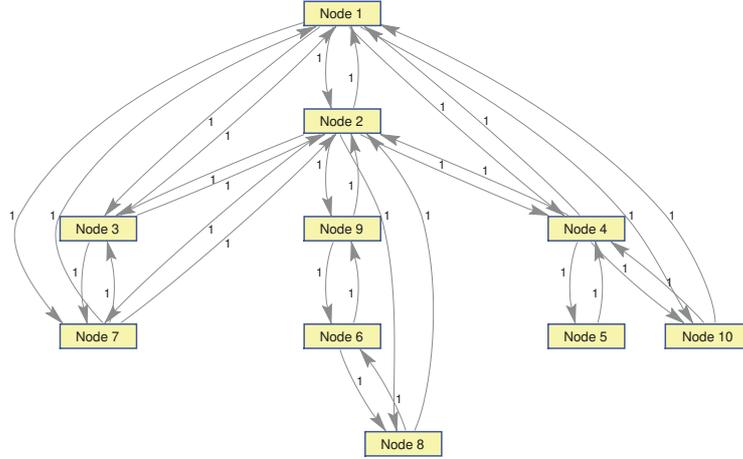


Figure 4.3: Example of PLOD-generated network with 10 nodes,  $\alpha = 0.1$ ,  $\beta = 1.5$

degree, such as sensors and actuators. We focus on this topology since the vast majority of critical control systems follow these structures, which produce small sub-networks similar to current control substations.

In greater detail, this generator considers the Power Law Out Degree (PLOD) algorithm to guide the construction of the graph [244], having the following form:

$$degree = \beta x^{-\alpha} \quad (4.2)$$

where  $x$  is a random number chosen in the interval  $[0, |V|]$ . This algorithm allocates a certain number of degree credits to each vertex in the graph and creates edges between them by deducting such credits, which are determined by  $\alpha$  and  $\beta$  parameters. Whereas  $\beta$  controls the y-intercept of the curve (so that increasing its value results in an increase in the average degree of vertices), the value of  $\alpha$  controls how steeply the curve drops off. Figure 4.3 shows an example of network with 10 nodes generated by the PLOD algorithm with  $\alpha = 0.1$  and  $\beta = 1.5$ . In our several simulations, we will use random values for  $\alpha$  uniformly chosen in the interval  $[0,1]$  and a  $\beta \simeq 1.5$ , which altogether generate hierarchical architectures similar to the figure, matching the desired control networks.

On the other hand, the IT section (given by  $G(V_{IT}, E_{IT})$ ) is modelled according to a small-world network distribution, that represents the conventional topology of TCP/IP networks [245]. In this category, the Watts-Strogatz is one of the most studied and implemented models. It was designed as a simple random graph generator that produces networks with short average path lengths and high clustering. A network features clustering if the probability that two nodes are connected is higher when both of them have a neighbour in common. This is present in real network topologies in Internet applications, where systems are assigned to private clusters or subnetworks in a more heterogeneous way.

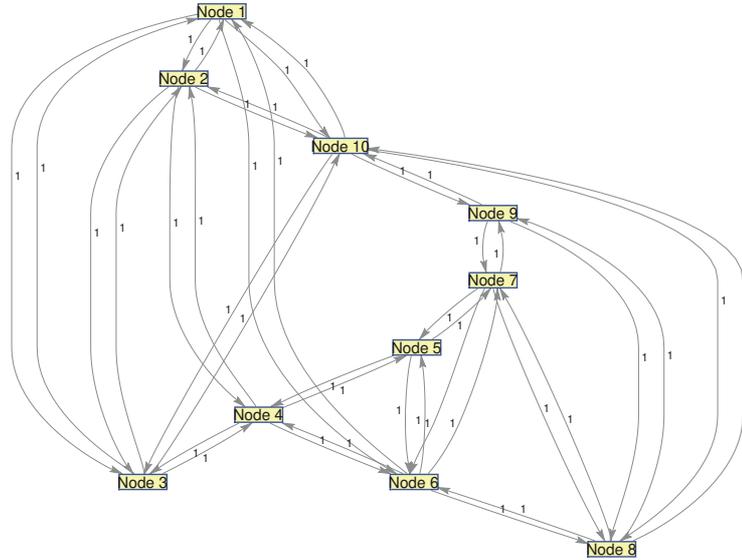


Figure 4.4: Example of network with 10 nodes generated following the Watts-Strogatz model, with  $p = 0.1$  and  $degree = 4$

To construct a graph with these properties, the Watts-Strogatz model receives as arguments the total number of vertexes  $|V|$  for the desired network, a *degree* value that represents the mean degree for all nodes and a probability  $p$ . Then, it generates a graph with  $\frac{|V|degree}{2}$  edges, according to the so-called rewiring process, which is outlined as follows:

1. Build a regular graph where each node has the same number of neighbours (the *degree* value)
2. Rewire each edge  $(v_i, v_j)$  in the network to a random node  $v'_j$  instead, where  $v'_j$  is selected uniformly at random from all nodes, as long as self-loops and link duplications are avoided.

In our case, we choose values close to 0 for the  $p$  parameter, since it generates highly clustered networks. An example of network graph generated with this configuration and  $degree = 4$  is represented in Figure 4.4.

Once the graph has generated for the IT and OT sections, both subnetworks are merged through firewalls following the process described in Section 4.1.1. Altogether, Figure 4.5 shows a simple example of a network with five IT nodes and five OT nodes, which are merged through two different firewalls.

Once we have established the architecture for the network, we are in position to not only simulate attacks over the topology, but also to develop distributed detection systems, which is the main contribution of our work.

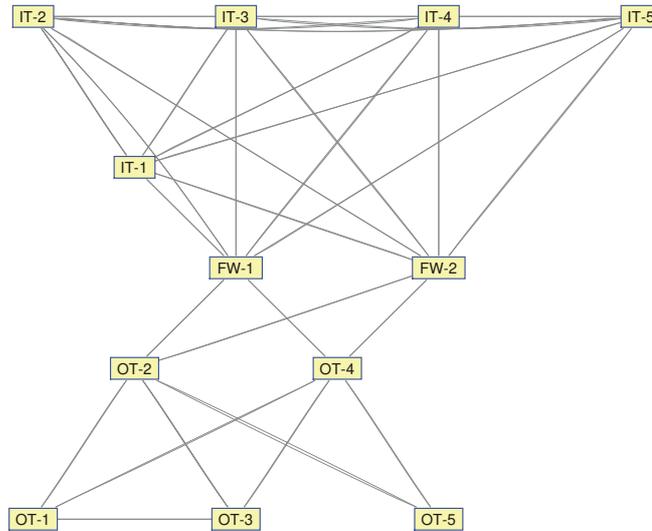


Figure 4.5: Example of network with five IT nodes and five OT nodes merged through two firewalls

### 4.1.3 Representation of APT Attacks and Detection Probabilities

After reviewing the behavior of industrial APTs and the state of the intrusion detection mechanisms, we can formalize a realistic attack and defense model for our network architecture, using graph theory. Our *attack model* is simple: we assume that, given a certain goal (exfiltration and/or destruction), adversaries are able to successfully perform an APT attack against the network architecture defined above, using any set of the attack stages defined in Section 2.4.2. As for the *defense model*, and given the state of the art in the area, we will assume that all the elements of the network are covered by distributed anomaly detection mechanisms, which are extensively investigated throughout this chapter. Compared to traditional detection mechanisms, these approaches feature the ability to correlate anomalies throughout the network and hence trace the location of attacks, also considering their severity and persistence. To achieve this, they securely retrieve information from any host-based and network-based detection mechanism deployed in the network defined by graph  $G(V, E)$ . At this point, to better understand the formal representation of APT attacks, we also assume that, as a result of the correlation of these detection mechanisms that monitor the behavior of a node and its neighbours, every node will be assigned a certain detection probability (i.e., probability of an attack taking place) for a given interval of time.

To formalize the attacker model, we can provide a representation of the intrusion kill chain of APT attacks. Let *attackStages* be a set of potential attack stages that an APT can perform against the industrial control network  $G(V, E)$  as defined in Section 2.4.2, such that  $attackStages = \{attack\ stage_1, attack\ stage_2, \dots, attack\ stage_n\}$ . This set comprises the following elements:

- **initialIntrusion**<sub>(IT,OT,FW)</sub>. The initial access that affects a node  $n_0$  (known as ‘patient zero’) of the IT network, OT network, and firewall, respectively.

- **compromise**. The adversary takes control of a certain node  $n_i$ , obtaining higher privileges, maintaining persistence, and executing defense evasion techniques. Moreover, this stage also includes the internal reconnaissance of the direct neighbourhood of  $n_i$ ,  $neighbours(n_i)$ .
- **targetedLateralMovement** $_{(IT,OT,FW)}$ . From a certain node  $n_i$ , the adversary chooses a FW, IT, or OT node  $n_j$  from the set  $neighbours(n_i)$ , and executes a lateral movement towards that node. Note that, in this model, the concept of lateral movement only encompasses the delivery of malware towards the target node.
- **controlLateralMovement**. From a certain node  $n_i$ , the adversary chooses the node  $n_j$  from the set  $neighbours(n_i)$  with the highest betweenness (i.e., the node with more connectivity), and executes a lateral movement towards that node.
- **randomLateralMovement**. From a certain node  $n_i$ , the adversary chooses a random node  $n_j$  from the set  $neighbours(n_i)$ , and executes a lateral movement towards that node.
- **exfiltration**. From a certain node  $n_i$ , the adversary establishes a connection to an external command&control network, and extracts information using that connection.
- **destruction**. The adversary either destroys node  $n_i$ , or manipulates the physical equipment (e.g., uranium enriching centrifuges) controlled by node  $n_i$ .
- **idle**. In this phase, no operation is performed.

Once the set  $attackStages$  is defined, it is possible to represent APT attacks that target our particular network model  $G(V, E)$ . In particular, for every APT, there can be an ordered set  $attackSet_{APT}$ , composed by one or more elements of the  $attackStages$  set, that represent the APT chain of attack actions. As an example, the attack set of Stuxnet [123] can be represented as follows:

$$attackSet_{Stuxnet} = \{initialIntrusion_{IT}, compromise, exfiltration, \\ targetedLatMove_{FW}, compromise, targetedLatMove_{OT}, \\ \dots, targetedLatMove_{OT}, idle, \dots, destruction\}$$

These particular instances are defined taking into consideration the overall goal of every APT. For example, in the case of the Stuxnet malware, its goal is to find a particular node  $n_{OT'} \in V_{OT}$  that manages an uranium enriching centrifuge. Therefore, after infecting patient zero  $n_{IT^0} \in V_{IT}$ , it seeks the location of a firewall node  $n_{FW} \in V_{FW}$  that connects the  $G(V_{IT}, E_{IT})$  and  $G(V_{OT}, E_{OT})$  regions. Afterwards, it moves inside the  $G(V_{OT}, E_{OT})$  region until it finds node  $n_{OT'}$ . Finally, after waiting for some time, the malware executes its payload, manipulating the centrifuge.

Regarding how the different attack stages influence the calculation of the detection probabilities, we need to consider that certain attack stages will generate more security alerts. This, in turn, will

increase the probability of detecting that particular attack stage. Therefore, we need to consider the existence of different classes of detection probabilities. Here, we define  $\Theta$  as an *ordered set of detection probabilities of size  $d$* , where  $\Theta = \{\theta_1, \dots, \theta_d\}$  and  $\theta_i = [0, 1]$ , such that  $\forall \theta_i, \theta_i > \theta_{i+1}$ .

<i>initialIntrusion</i> ( $n_0$ )	$\theta_3$
<i>compromise</i> ( $n_i \rightarrow neighbours(n_i)$ )	$\theta_2 \rightarrow \theta_5$
* <i>LateralMovement</i> <sub>IT,FW</sub> ( $n_i \rightarrow n_j$ )	$\theta_5 \rightarrow \theta_4$
* <i>LateralMovement</i> <sub>OT</sub> ( $n_i \rightarrow n_j$ )	$\theta_5 \rightarrow \theta_3$
<i>exfiltration</i> ( $n_i$ )	$\theta_4$
<i>destruction</i> ( $n_i$ )	$\theta_1$

Table 4.2: Map of *attackStages* to  $\Theta$

Once  $\Theta$  is defined, we can create a model that maps every element of the set *attackStages* to the elements of  $\Theta$ . Such model, where  $d = 5$  and  $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5\}$ , is described in Table 4.2. The rationale behind this mapping is as follows:

- We assign  $\theta_1$  only to the *destruction* stage, because any major disruption in the functionality of a device (e.g., unavailable resources, device turned off) will trigger multiple high priority alerts. Note that, as explained in our defense model, we assume that all field devices are also covered by detection mechanisms, thus any attack (e.g., the Stuxnet final payload) against these sensitive devices can be easily detected.
- $\theta_2$  is only assigned to the element at the left side of the *compromise* stage ( $n_i \rightarrow neighbours(n_i)$ ). The reason of this is simple: the act of compromising and taking control of  $n_i$  will not only trigger various host alerts, but also multiple network alerts due to the various discovery queries targeting all *neighbours*( $n_i$ ). The correlation of all these events will draw attention to the state of  $n_i$ .
- For  $\theta_4$ , we consider the security alerts caused by combination of a single anomalous connection to a node plus the delivery of malware to that node. As such, this  $\theta$  covers all the elements at the right side of the *lateralMovement* stages. Note, however, that in some particular cases (like the *initialIntrusion* stage and the \**LateralMovement*<sub>OT</sub> stages), additional anomalies will be detected: a potentially anomalous external connection, and a certain instability in the otherwise stable OT communication environment, respectively. Therefore, the  $\theta$  assigned to the elements of those stages will be  $\theta_3$ .
- Finally,  $\theta_5$  is assigned to those stages where the nodes produce or receive anomalous traffic (e.g., a connection that deviates from what is considered as normal traffic). Again, in situations where a connection with the outside world is made (e.g., *exfiltration* stage), as the possibility of anomalous traffic will increase, the  $\theta$  will be increased as well.

After introducing and formalizing the key elements involved in the attacker model and distributed detection, the background is set for the definition of a traceability framework that eases the development of solutions to tackle the issue of APTs raised in this work.

## 4.2 APT Traceability Framework for the Industry 4.0

After reviewing some of the most representative methods for the intrusion detection in IIoT environments and formalizing the infrastructure of Industry 4.0 and the behavior of APTs, in this section we present the distributed traceability framework, which is the core of this work. Compared to these aforementioned works, this approach does not limit to monitor the system in specific points of the infrastructure in the seek of anomalous behaviors with specific machine learning-based algorithms [218]. In turn, it proposes to aggregate the coverage of multiple detection systems that are strategically deployed over the infrastructure, under a common distributed framework that permanently correlates and learns from all the malware patterns detected and individual anomalies measured. We can summarize its contributions as follows:

- **To circumvent the heterogeneity of IDS solutions.** The review of the state of the art concludes that there is no ‘silver bullet’ that successfully addresses all the cybersecurity threats in IIoT. Instead, there are mechanisms that focus on specific attacks or leverage techniques that are tailored for specific sections of the control network. With this framework, we want to combine various solutions to provide protection at all levels.
- **To anticipate and accommodate new technologies and business scenarios.** Traditionally, the elements of the control networks have behaved in a predictable way. However, with the advent of the so-called Industry 4.0 technologies, new scenarios and services will appear, such as flexible production lines or predictive maintenance systems, for instance. These will allow the organizations, suppliers, users, etc. to collaborate under a fully interoperable model of industry. Therefore, it makes necessary to develop new detection systems capable of analyzing these autonomous systems and their interactions. At the same time, these systems are also expected to accommodate the integration of new technologies to the infrastructure, that will also bring with them new vulnerabilities and exploitable attack vectors. In this sense, our framework constitutes a fully adaptable solution.
- **To ease the traceability of attacks and the precise application of response procedures.** In order for the operators to gain knowledge from intrusions and effectively improve decision-making, the traceability framework facilitates the study of the evolution of these attacks throughout their entire life cycle. This is more critical in the case of APTs, where stealthy techniques are used to go unnoticed for a prolonged lapse of time, when the attacker propagates over the network. In other words, we can obtain meaningful information that correlates subtle events and evidences with actual attack stages and tactics, from a higher

strategic perspective (and beyond low-level alerts raised by traditional IDSs). This also includes, for example, noise filtering and the reduction of false positives, potentially provoked by misconfigured services or network overload. Altogether, this eases the deployment of response procedures that permit to anticipate the next attack action and hence reduce the impact of these threats.

After defining the detection and security requirements that a conceptual APT traceability solution must fulfill, we now describe the guidelines for the design and construction of its deployment architecture, the algorithms to be used, and the attacker model under consideration.

#### 4.2.1 Network Architecture and Information Acquisition

As introduced before, the industrial network topology is modelled with the cyclic graph  $G(V, E)$ , where  $V$  represents the devices and  $E$  is the set of communication links between them. This way,  $V$  can be assigned with parameters to represent, for instance, their criticality, vulnerability level or the degree of infection; whereas the elements in  $E$  can be associated with Quality of Service (QoS) parameters (e.g., bandwidth, delays), or compromise states that help to prioritize certain paths when running resilient routing algorithms.

For the interest of theoretical analysis, these networks are frequently generated using the random distributions introduced in Section 4.1.2, that model the architecture of real industrial systems. As also mentioned, the topology is usually subdivided into multiple network segments with different distributions [82], which is useful to study the effects of the attack and detection mechanisms over the corporate section (containing IT elements) and the operational section (OT, containing pure industrial assets), which can be connected by firewalls, so that  $V = V_{IT} \cup V_{OT} \cup V_{FW}$ .

Regardless of the topology configuration, the detection approach must acquire information from the whole set of nodes  $V$  to fulfill requirement D1 (Coverage, c.f. Section 3.5), by using agents that are in charge of monitoring such devices, complying with S1 (distributed data recollection). These are deployed as a middleware on top of the physical infrastructure, inspired by FIPA (Foundation for Intelligent Physical Agents) specifications to support the communication and coordination between intelligent agents [246]. This standard facilitates the development of multi-agent systems under a common definition of containers and interfaces for the agents, in order for them to run in one or more systems. In our case, each of these agents follows a basic life cycle whenever the traceability solution is executed, and consists in the retrieval of physical information from the environment followed by the correlation of anomalies with the rest of agents. More specifically, we can assume they are able to retrieve as much data as possible from their assigned devices, which encompasses the following items:

1. **Network parameters:** it mostly comprises the set of communicating devices and the state of every communication link in order to characterize the graph  $G(V, E)$ .

2. **Communication information:** low-level commands issued by the control and supervision protocols of the industrial applications (e.g., reading values, actions executed).
3. **Host-based information:** computational usage of the monitored device and locally stored information.

These data items are aimed to feed a correlation algorithm with inputs in the form of an anomaly value for every device audited, which is formalized by vector  $x$ . This way,  $x_i$  represents the anomaly value sensed by the corresponding agent on device  $i$ , for all  $i \in 1, 2, \dots, |V|$ , which is represented in a scale from 0 to 1 of continuous values. Such value is calculated by each agent, using two possibly simultaneous approaches: in an autonomous way (e.g., applying some machine learning to determine deviations in every data item analyzed with respect to its value in normal conditions) or leveraging an external IDS that is configured to retrieve the raw data as input (including events triggered by vulnerability scanners or antivirus software), thereby conforming to requirement D3 (Intelligence). Either way, the aim is to be open to include new anomaly indicators that serve as an input to agents, to realistically analyze the security state of each node and its neighborhood.

With this, we assume that the agent would have enough input data to compute a single anomaly value for the security state of its monitored device. At this point, the effectiveness from the use of specific ways to derive such value could be compared, which would strongly depend on the actual network setup (e.g., topology, technologies, communication protocols) and it is not in the scope of this work. Instead, we point out that the novelty and effectiveness of this approach resides in the ability to correlate anomalies throughout the network and thereby get insight into the location and severity of attacks. The way to uptake the individual anomaly detection is customizable and reliant on the security scenario that we want to achieve, thereby working as a framework.

From a deployment perspective, this leads to the question of where to locate the computation of anomalies and their subsequent correlation, as to implement this mechanism in an industrial infrastructure. In summary, these agents can be either logical or physical. Logical agents imply that we assume that the status of individual devices can be retrieved from a centralized entity, which consists of a computationally powerful node in charge of correlating the anomalies from all agents, that are executed virtually. Ideally, this node would then apply protection measures (e.g., data recovery, backup servers, honeypots) based on the security state of the network. In practice, this can be easily implemented by using switches in port-mirroring mode, so that all traffic from the nodes is relayed to a central correlator system, for instance. This setup model is already applied by several commercial platforms, such as [247], whose goal is to provide support for event correlation. These platforms can retrieve events and alerts from various domains (e.g., IT, OT networks) and from various sources (e.g., SIEM systems, vulnerability scanners) in a distributed way.

On the other hand, we could also consider that these agents can be physically deployed over the network, in form of monitoring devices or integrated with the software of the industrial assets. In other words, we assume that there is one agent attached to each node within the network (following a 1:1 relationship), which would be ideal for S1 (cf. Section 3.5). Such agent should measure the anomaly for itself and convey such value to its neighbors for the execution of a correlation algorithm, communicating via the original topology in a fully distributed manner. However, this option is not always feasible, since manufacturers and operators of critical infrastructures are reluctant to introduce modifications in their hardware and software, which could also be privative and hence not allow the execution of third party programs. Additionally, it is not always feasible to physically integrate monitoring devices into industrial assets due to computational limitations. Consequently, these processes may have to run in separate computational nodes.

However, we still want to achieve a close connection to field devices while avoiding a centralized implementation. Two potential solutions are proposed for these cases. The first one is the election of a subset of nodes within the control system to play the role of physical agents, depending on how easy is their integration via software/hardware. This way, those agents (which should be strategically dispersed over the network) would be the only ones in charge of detecting the anomaly in their devices and also in those other surrounding devices that lack an agent. In this regard, the concept of the Dominating Set introduced in Section 4.1.1 would be suitable for the agent election.

Another solution is to leverage the concept of *distributed data brokers* to carry out a partially distributed (or decentralized) implementation, assuming logical agents. These are independent physical components that collect the data from a set of individual devices via port-mirroring or network tapping, using data diodes to decouple agents from actual systems. This way, they ensure that data transmission is restricted to one direction, thereby shielding the industrial assets from outside access and complying with requirements S3 and S4.

Under this configuration, these data brokers can also convey the detection reports (i.e., the anomalies sensed by its logical agents over the area where it is deployed) to other brokers in order to execute the correlation in a collaborative way. Additionally, we could also contemplate a last implementation model by enabling a distributed interconnection between data brokers and existing physical agents within the network. Consequently, these broker entities should be strategically deployed in a separate network such that there is at least one path between every two brokers and the potential device agents.

These four interconnection models for the distributed acquisition and correlation of information (assuming the implementation of either physical or logical agents) appear depicted in Figure 4.6. It is necessary to emphasize that the election of any of these agent implementations is transparent for the anomaly correlation algorithm to be developed, so that it is conceived for any distributed environment independently of the origin of the data. Due to this distributed nature, the correlation algorithm can make use of two data models: *replicated database*, which assumes that every agent has complete information of the whole network (through distributed ledgers or

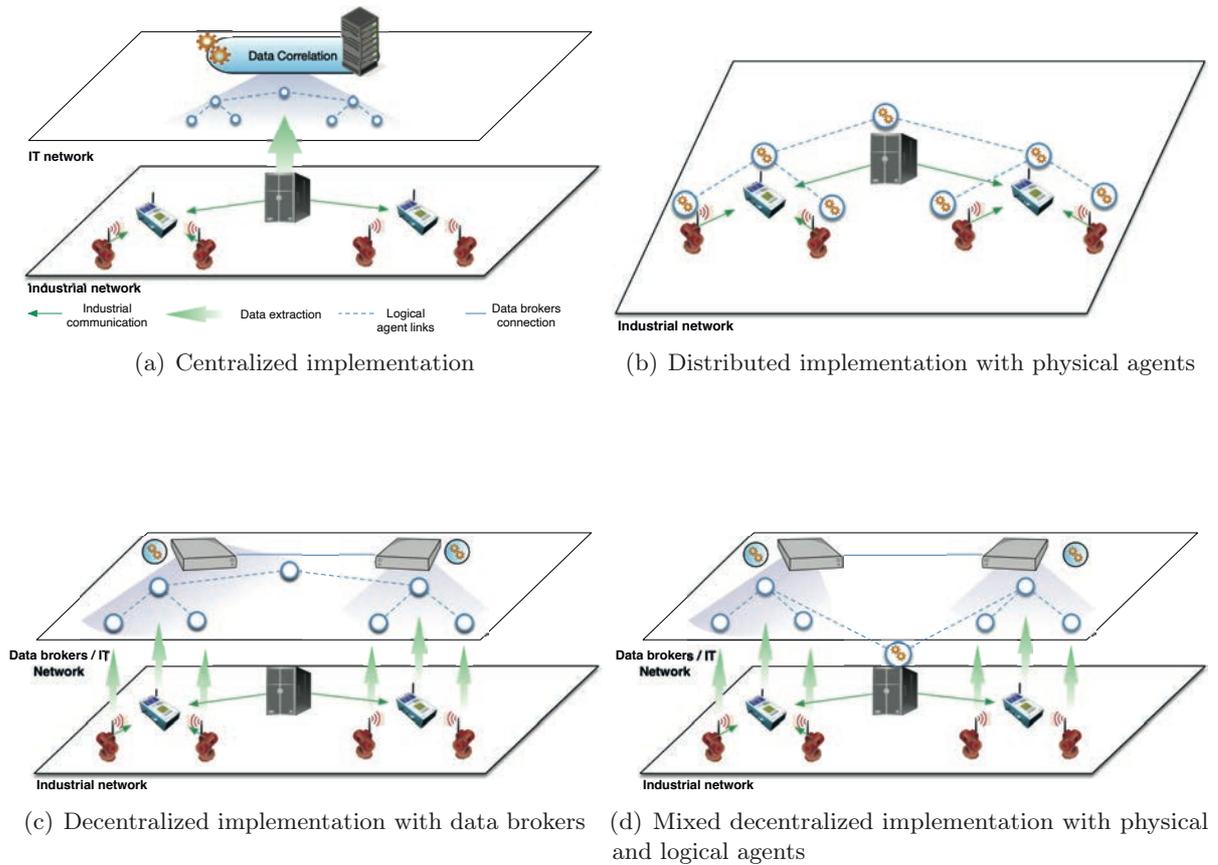


Figure 4.6: Agent implementations for information acquisition and correlation

using logical agents in a centralized entity), and *distributed data endpoints*, where the information is fully compartmentalized and the cross-correlation is conducted at a local level. Both approaches have their advantages and disadvantages. The replicated database provides all agents with a vision of the network, although it imposes some overhead with respect to the synchronization of information across agents. As for the distributed data endpoints, they reduce the number of messages exchanged, yet the algorithm must deal with partial information coming from neighbour peers.

After all, the ultimate election of the algorithm, data model and architectural design of the agents responds to performance and overhead restrictions. These parameters determine the detection mechanism at a physical layer, while at an abstract level it must also return a set of security features, described in the following.

## 4.2.2 Inputs and Outputs of the Traceability Solution

After introducing how the information from physical devices is collected by agents in practice and how the anomalies can be calculated in theoretical terms for our simulations, we summarize the set of inputs for traceability solutions as:

- ( $I_1$ ) **Quantitative input:** expressed with vector  $x$  to assign every industrial asset with an anomaly value prior to conducting the correlation. As previously mentioned, it can be calculated by each associated agent or using external detection mechanisms integrated with the data broker by taking an extensive set of data inputs to comply with D1 and D2. In our simulations, this value is given by the attack phases executed on the network in a probabilistic way, without the detection mechanism having any knowledge about the actual stages.
- ( $I_2$ ) **Qualitative input:** the previous values need to be enriched with information to correlate events in nearby devices and infer the presence of related attack stages, according to Section 2.4.2. At the same time, we also need to prioritize attacks that report a higher anomaly values. We assume that the resulting knowledge can be reflected in form of a weight  $w_{ij}$ , which is assigned by every agent  $i$  to each of its neighbours and represents the level of trust given to their anomaly indications when performing the correlation (fulfilling S2). This parameter can be subject to a threshold  $\varepsilon$ , which defines when two events should be correlated depending on the similarity of their anomalies (e.g., two neighbor agents that sense the same degree of anomaly due to communication delays would assign a higher weight to each other since that event could be probably related to their shared connection). Further criteria could be introduced to associate anomalies from different agents.

With respect to the outputs of the traceability solutions, they should include, but are not limited to the following items:

- ( $O_1$ ) **Local result** to determine whether the agent is generating an anomaly due to whether the actual infection of the associated node, as a result of a security threat in a neighbour device or a false positive.
- ( $O_2$ ) **Information at global level**, to determine the degree of affection in the network and the nodes that have been previously taken over, filtered by zones. This allows to distinguish what set of devices are experiencing the same degree of anomaly produced by a particular attack. This information is essential for applying effective response techniques and potentially isolate the attack, while the rest of the areas can keep functioning as in normal conditions, hence ensuring the continuity of the production.
- ( $O_3$ ) **Contextual information** that permits to correlate past events and visualize the evolution of the threat, but also anticipate the resources that are prone to be compromised (D3 & D4).

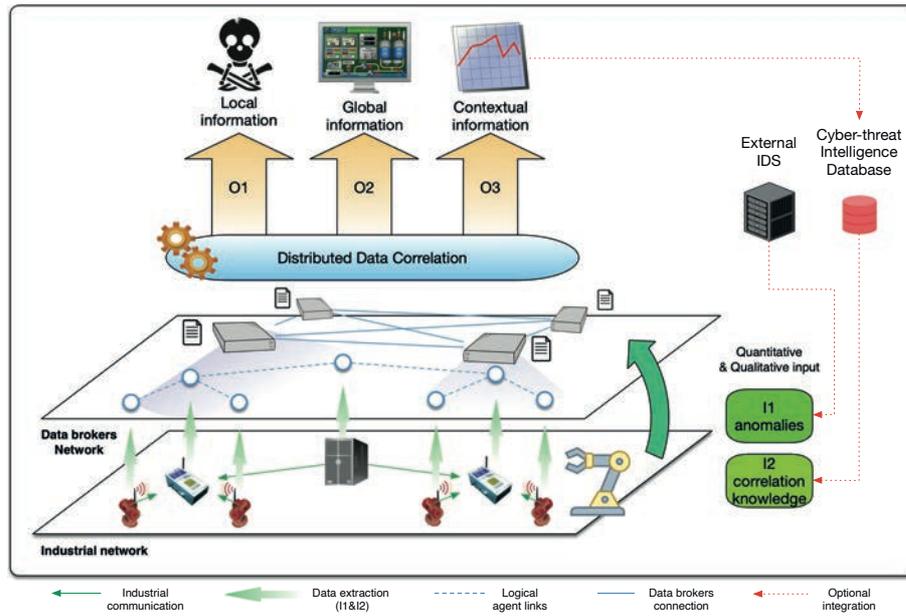


Figure 4.7: APT distributed detection and traceability framework

This includes the events occurred to the network since the very first moment the intrusion broke into it. In this sense, when it comes to APTs, we must also take the persistence of attacks into special consideration at all times, since an advanced threat can go unnoticed for months and suddenly perform a new attack. In terms of the traceability technique, this implies that it is also necessary to keep track of old subtle anomalies noticed in the network, to serve as feedback to the technique and correlate their relevance with current detected anomalies, which may be part of a more ambitious threat.

This comprehensive analysis of the requirements and techniques defines a framework for the development of distributed detection solutions for APTs in industrial scenarios, as depicted in Figure 4.7. This diagram illustrates the data flow since its acquisition from end devices until the correlation is computed, possibly going through the data brokers introduced in Section 4.2.1. The following section presents some of the candidate solutions that implement them, and hence achieve the APT traceability goals proposed so far.

### 4.3 Distributed Correlation Models

The proposed framework clearly defines an information acquisition model and interface configuration that must be suited by enabling correlation algorithms. According to their mathematical formulation and their application context, these may feature different ways to represent the environmental inputs to carry out the correlation, under a unified black box-style specification.

In this section, we present the candidate solutions for the traceability of APTs in a concise way, before analyzing them in detail in next section. We begin by introducing a group decision making technique based on consensus, to later instantiate distributed consensus using Opinion Dynamics. Then, an alternative model based on Clustering is included, to finally conclude with a brief discussion on the benefits and downsides of each proposal.

### 4.3.1 Consensus Model

Using graph theory as a foundation of group decision making models, the first approach considered here is *consensus*. We will use this model to study its fitness with the framework and illustrate how the underlying concept of consensus is appropriate to fulfill the correlation needs raised previously. This is a classical problem in distributed computing and multi-agent systems, that studies how a set of agents are able to obtain the same information in order to reach a common objective. It has been extensively addressed in real-world applications such as clock synchronization, data aggregation between nodes in a blockchain, or the coordination of autonomous robots, among others.

Compared to autonomous systems for particular isolated purposes, it has been demonstrated that a multi-agent deployment of systems operating in a coordinated fashion offers a greater efficiency and operational capability. Nevertheless, these cooperation capabilities are dependant on the coordination protocol used, which obliges each agent to share information about its state and environment with the rest. This involves challenges around what is transmitted, when and with whom, taking into consideration communication aspects like the bandwidth, the network connectivity and the computational resources available, hence forming the consensus problem.

For these reasons, reaching information consensus in a network that may be noisy or time-varying is critical for a successful coordination of tasks. To this end, a consensus protocol is intended to provide a concise formalism to deal with the dynamic conditions of the network and help agents communicate with each other so that all their information states converge to a common value.

Depending on how the protocol copes with dynamic conditions and how the negotiation process between agents works, we can distinguish between different types of consensus [248]. One of the most popular approaches in distributed environments without time delays is the average consensus, where the states of all agents converge to the average of their initial states. In formal terms, these averaging algorithms are linear (a combination of the initial information states) and iterative, so that they can be expressed as follows:

$$x(t+1) = W(t)x(t), t \in 0, 1, 2, \dots, \quad (4.3)$$

where

$$x(t) = \begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{pmatrix} \quad (4.4)$$

is the real vector that contains the values measured by each of the  $n$  agents in the network, matching the notation introduced before with regards to the quantitative input of the traceability solution that we aim to find. On the other hand,  $W(t)$  is the weighting matrix that satisfies that if two agents  $v_i$  and  $v_j$  are not connected in the network defined by  $G(V, E)$  (i.e., they are not neighbours), then  $[W(t)]_{j,k} = 0$ . If we assume that  $G$  is a connected graph (i.e., there is a directed path between any pair of distinct vertices to share their information state) and  $[W(t)]_{i,j} = 1/n$  for all agents  $i$  and  $j$ , then the consensus equilibrium is achieved in finite time  $t > 0$ , which is equal to the average of the initial information states and is called *average consensus*. This way, the information state  $x_i(t)$  of every agent  $i$  is driven towards the states of its neighbours as  $t$  increases, approaching the average, as shown in Figure 4.8.

This is the general average consensus model, which admits precise characterization to create more specialized models. Firstly, depending on the components of this matrix, the average consensus algorithms can be classified as *deterministic* (as the previous one, where  $W(t)$  is symmetric and time-invariant) or *randomized* (where agents are connected at random intervals and the weighting matrix hence changes). Another distinction is made whether the network allows continuous communication between agents (i.e., *linear consensus-time consensus*) or if the communication data arrives in discrete packets (i.e., *linear discrete-time consensus*). Also, in contrast with the general model, when the communication topology is not connected, but in turn has a directed spanning tree (i.e., a tree formed by the edges that connect all the vertices of the graph  $G(V, E)$ ), the consensus equilibrium is determined by the weighted average of the initial states of those agents that have a direct path to all the rest of agents.

In addition to the different communication models for the consensus protocol, it can be combined with special measurements to prevent against faulty agents and crash failures in the communication. In particular, this makes reference to the classical FLP theorem [249], which was named after the authors Michael J. Fischer, Nancy Lynch, and Mike Paterson and establishes that no agreement can be guaranteed in an asynchronous system in the presence of failures. Thus, it is well known that a consensus protocol tolerates halting failures when it satisfies the following properties:

1. **Termination:** eventually, every agents calculates its value.
2. **Integrity:** all the agents process the same data from the network.
3. **Agreement:** every agent agrees on the same consensus value.

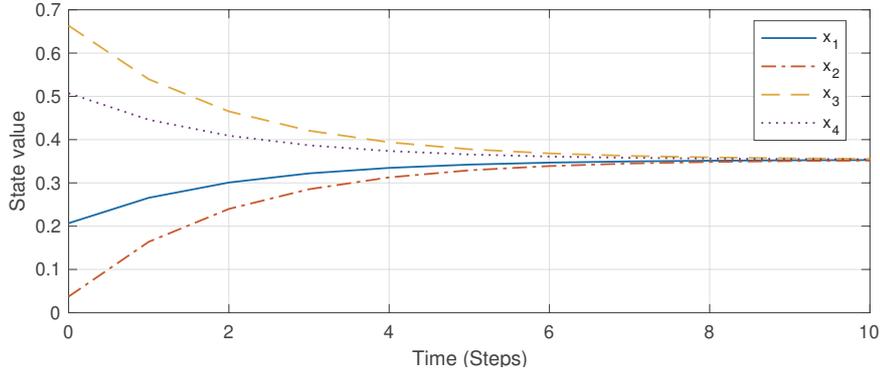


Figure 4.8: Average consensus for three agents

Besides crash failures, the distributed set of agents may also undergo byzantine failures, which are those that do not abruptly stop the protocol but still hinder the consensus process. They include the presence of malicious agents that send conflicting data to others, which is certainly harder to circumvent. In the literature, this is known as the Byzantine Generals problem [250]. It has been demonstrated that for a system with  $n$  agents, of which  $f$  of them are Byzantine, there is no consensus algorithm that solves the consensus problem for  $n \leq 3f$ . Protocols that comply with this property are called as byzantine fault tolerant.

On the whole, consensus poses a valid solution for distributed deployments where a common solution must be found in a collaborative manner. For the interest of our analysis regarding threat detection, we will represent the state of the agents in form of anomaly values. Having this, this mechanism can provide us with an indicator of how healthy the entire network is at any given time instant, based on the information provided by all agents after reaching a consensus. However, the correlation does not return any information at a local level, to comply with the  $O_2$  output desired for the traceability solution, as stated in Section 4.2.2. In other words, the consensus does not generate any insight about the degree of affection of particular nodes and portions of the network. As a consequence, potential countermeasures to be implemented would lack sufficient accuracy as to solve attacks located in precise devices.

Although this limitation hardens the suitability of the average consensus for the proposed traceability framework, we can tweak the original consensus concept to consider the presence of more than one consensus, thereby adapting the approach to our needs. This is illustrated in the following section with the Opinion Dynamics model.

### 4.3.2 Opinion Dynamics

In this section we describe a feasible method to allow the network to precisely locate subtle changes in certain parts. In the consensus approach, a collection of agents cooperate to reach a common objective by sharing information about their state and other environmental conditions [251].

Such negotiation depends on the network topology, so it can be leveraged to collectively build a global indicator of the entire network health at a given moment. Compared to this algorithm, Opinion Dynamics proposes a model that admits the fragmentation of patterns. This way, the aforementioned agents may differ in their opinions (i.e., their information states) during the negotiation process [252]. Therefore, there could be consensus among the agents or a polarization between them in multiple areas along the network, which makes it able to identify which areas of the network are more affected by the action of a potential APT and to what extent.

This information is extracted by means of a distributed cooperative algorithm called Opinion Dynamics [252], which originally models the influence among individuals in a group or the entire society, where there is a wide spectrum of opinions. Each agent crafts its own opinion taking into consideration the ones from the rest of agents to a certain extent. This process continues until reaching a steady state in which the agents no longer change their opinion. At that point, the opinions are distributed into several clusters, and it is possible to study their propagation. For our purpose, it implies fragmenting the network according to the multiple changes that could occur in separate areas, whose individual consensus value raises an indicator of the severity of the attacks over that particular portion of the topology.

Note that this model has attracted the interest of researchers in sociological studies over decades. Back in 1951, Asch and Guetzkow analyzed the effect of group pressure on social dynamics [253], while the Opinion Dynamics was firstly conceived as such in 1956 by French [254]. Then, a continuous-time model was proposed by Abelson in 1964 [255], and the most well-known model (and also one of the simplest) was established by DeGroot in 1974 [256]. Ever since, there has been extensive discussion in the research community with the proposal of specialized models for particular application scenarios, as it occurs with the consensus. These can be summarized in the following categories [257]:

**Continuous opinion space models:** models in this category assume that the opinion space is continuous, this is, each agents holds an opinion in the  $[0,1]$  interval, for instance.

- DeGrootian models: it is an iterative averaging model where the convergence is equivalent to the consensus model if the network is connected [256]. Its major extension is known as the Friedkin-Johnsen model [258], which introduces the idea of stubborn-agents, by including a susceptibility degree for every agent to be influenced. Its convergence and stability are studied in [259].
- Bounded confidence models: in these models, agents ignore the opinions that are too different from their own. This is achieved by introducing a confidence threat to the system. The opinions fluctuation can be calculated pairwise (like in the Deffuant-Weisbuch model [260]) or in a synchronous way for the entire set of agents in the network (such as the Hegselmann and Krause approach [252]).

**Discrete opinion space models:** contrary to continuous opinion space models, these approaches consider opinions that take discrete values (e.g., the binary value in 0,1).

- Galam model: it is the main work in this category that has inspired other researchers to create applications in real life, with democratic voting or decision making as classic use cases [261]. In particular, this model assumes a binary opinion space, and the update rule for each agent work as follows: (1) agents are randomly assigned to groups of a given size; (2) each group updates its opinion on the basis of the majority; (3) agents are shuffled and the process starts again at step (1).
- Snajd model: this model considers that agents are sitting on a 1-dimensional lattice with the opinion space  $O = -1, +1$  [262]. This way, if  $o_i^t$  represents the opinion of agent  $i$  at time  $t$ , two neighbours  $i$  and  $i + 1$  are selected randomly at time  $t$ . If  $o_i^t \times o_{i+1}^t = 1$ , then the preceding agent  $i - 1$  and the subsequent agent  $i + 1$  adopt the direction of agents  $i$  and  $i + 1$ . Otherwise, each agent adopts the opinion of the immediate neighbor. This way, the system reach equilibrium when all agents agree at either  $-1$  or  $+1$  or a stalemate.
- Voter model: this subcategory comprises multiple variations that commonly assume a binary opinion space. A random agent  $i$  is chosen at a given time  $t$  and then  $i$  chooses another neighbour randomly, adopting its state. Sood and Redner [263] investigate this model on a heterogeneous graph, whereas studies its convergence on a graph with two cliques [264]. Also, the influence of external sources is explored in [265].

In specific, in this work we will focus on bounded confidence models, where agents ignore the opinion that are too far from their own, which in our case helps to differentiate the anomalies of separate attack stages across the network. Among these models, introduced before, the most popular version is given by Hegselmann and Krause [252].

In the following, we formalize this multi-agent algorithm, which constitutes a light modification of the approach proposed in [252]. We start with the notion introduced before. Let us suppose a network defined by the *directed* graph  $G = (V, E)$  and represented by the adjacency matrix  $M$ , as formalized in Section 4.1. We suppose the presence of  $n$  agents deployed over that network (so that every node  $v$  in  $V$  has an associated agent). Our goal is to put into practice a distributed cooperative algorithm among these agents to detect precise attacks in their neighborhood by exchanging information on changes produced in their surroundings.

In this context,  $x_i(t)$  represents the opinion of a fixed agent  $i$  at time  $t$  (ranging from zero to one), where  $t$  refers to the iteration of the algorithm. The vector  $x(t) = (x_1(t), \dots, x_n(t))$  represents the opinion profile at time  $t$  for all the agents. On the other hand, given an agent  $i$ , the weight given to the opinion of any other agent  $j$  is denoted by  $w_{ij}$ , where  $\sum_{k=1}^n w_{ik} = 1$  (therefore, agent  $i$  also takes its own opinion into account). These weights can change over time or by opinion, so that an agent  $i$  adjusts its opinion in period  $t + 1$  by taking the opinion of each agent  $j$  into

consideration at time  $t$ . Finally, the formation of the opinion for agent  $i$  in the next iteration  $t + 1$  is described as follows:

$$\sum_{j=1}^n x_i(t + 1) = w_{ij}x_j(t)$$

In a matrix notation (as previously explained with the consensus model), it can be written as:

$$x(t + 1) = W(t, x(t))x(t)$$

where the matrix  $W(t) = [w_{ij}]$  is the square matrix that collects the weights, which summarizes the relationships between the agents' opinions. For simplicity, for a given agent, we assume in the original model that the weight value assigned to its neighbors is uniformly divided into those agents whose opinion is very close to its own value (we establish an epsilon value of 0.2 of deviation between both opinions). This models the fact that agents close to each other with the same degree of anomaly are likely to be detecting the same threat in their surroundings.

Consequently, every agent adjusts its opinion in period  $t + 1$  by taking a weighted average of the opinions of the rest of agents. When  $t$  tends to infinity, consensus of opinions are formed (and finally there are just a few opinions shared by clusters of agents), which can also be represented visually. Altogether, the correlation is performed by every agent as a weighted sum of the closest opinions, and such calculation can be performed by solely using the information from neighbouring agents, thereby adapting to the distributed architecture based on data brokers (either replicating data or not). Conversely, what we accomplish in this scenario is the representation of anomalies detected by some of the agents installed within the network, so that clusters of agents returning similar high values (provoked by the same threats) correspond to critically affected areas from a high-level perspective.

Figure 4.9 shows the Opinion Dynamics algorithm for a network of 30 nodes and 17 agents after suffering an APT comprising 10 attacks. The lines represent the evolution in the opinions for each agent, so finally there is multiple consensus between them. In particular, there are only two agents that indicate relatively large changes (more than 0.5 of anomaly). However, four agents agree on a change of about 0.25 points around their zone of influence, and many of them indicate a fault of approximately 0.1 in the zone governed by these nodes. As can be seen in the figure, a  $\mu$  value has been added to the plot, which holds the ratio of agents that find a consensus on the amount of degree experienced. This value, together with the opinion about the changes in the network, serves as the criticality indicator at a global and local level, complying with the two first outputs imposed by our traceability framework. Also, as explained in Section 4.6, it is possible to account for the evolution of these values over time to comply with the third output, related to the analysis of historic and contextual information.

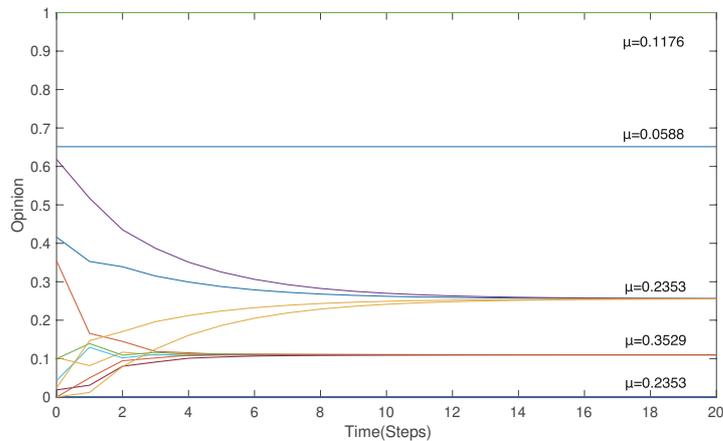


Figure 4.9: Calculus of the Opinion Dynamics for a set of agents

### 4.3.3 Clustering Algorithms

Opinion Dynamics belongs to a set of dynamic decision models in complex networks whose aim is to obtain a fragmentation of patterns within a group of interacting agents by means of multiple *consensus*. This fragmentation process is locally regulated by the opinions and weights of the nodes, that altogether abstract the effects of an APT dynamics on the underlying network. This ultimately enables to take snapshots of the current state of the network and highlight the most affected nodes, thereby tracing APT movements from anomaly events.

After all, the Opinion Dynamics simply divides a network into subgroups of devices that present a similar anomaly, and relates areas that may have experienced the same attack. This rationale can also be applied to different mechanisms with similar results, thereby fulfilling the established traceability framework of Section 4.2. Here we propose to adapt clustering algorithms as an alternative solution. These have been traditionally used as an unsupervised method for data analysis, where a set of instances are grouped according to some criteria of similarity. In our case, we have devices that are affected by correlated attacks (see Section 2.4.2) and show similar anomalies, which results in the devices being grouped together.

There exist several clustering approaches, each one suited to a particular data distribution [266], as illustrated in Figure 4.10:

- **Centroid-based clustering:** a simple division of the dataset into a predefined number  $k$  of disjointed clusters, so that each point in the original set belongs to one of these subsets. They are efficient and sensitive to outliers.
- **Density-based clustering:** in this type of approaches, a cluster is a dense region of objects surrounded by a low-density region. It is usually used when noise is present in the data, with the downside that it does not assign outliers to clusters.

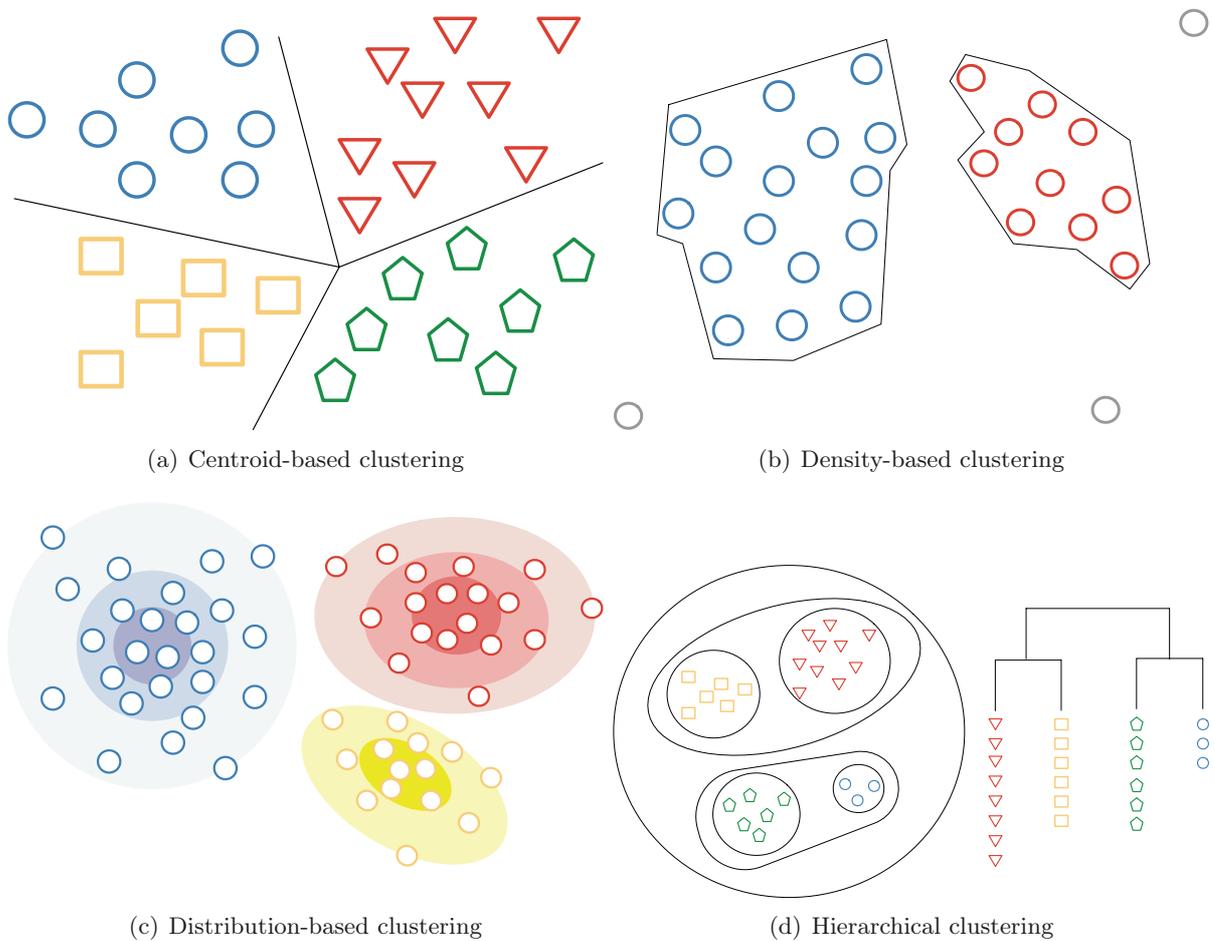


Figure 4.10: Different types of clustering approaches

- **Distribution-based clustering:** they work under the assumption that input data is composed of distributions, such as Gaussian distributions. This way, the probability that a point belongs to a cluster decreases as distance from its distribution center increases. As such, it is not recommendable when distribution of data is unknown.
- **Hierarchical clustering:** if in partitioning grouping each cluster is allowed to have sub-clusters, a hierarchical clustering is obtained. Therefore, clusters can be nested, so that they are organized in tree form.

For our purposes, we are dealing with a dataset of anomalies (denoted by the vector  $x$ ) whose distribution is not known in advance. However, we are interested in allowing outliers and the ability to group nodes into non-overlapping clusters, in accordance with separate attacks and levels of severity across the infrastructure. As a result, clustering approaches based on centroids are suitable for these needs. Classical methods such as K-means partition a dataset by initially

selecting  $k$  cluster centroids and assigning each element to its closest centroid. Centroids are repeatedly updated until the algorithm converges to a stable solution.

In the scenario of APT detection, the anomalies detected by the agents at some point play the role of the data instances to be grouped into clusters. The relationship between anomalies will determine which of those are grouped into the same cluster, which can be determined based on multiple criteria. As explained in Section 4.2.2, this represents the qualitative input to the event correlation solution. In the Opinion Dynamics, this is formally represented by the weight  $w_{ij}$  assigned between agents, that regulates when two opinions are correlated. As for the clustering approach, this is usually modelled in form of additional dimensions of the data points. For example, further values with respect to the traffic or computational usage reported in a given node (together with the  $x$  anomaly) would help to accurately isolate anomalies caused by the same incident.

In this sense, it is especially interesting to look into the representation of the network topology using a clustering approach. Whereas agents in Opinion Dynamics intrinsically take into consideration the connectivity among nodes to exchange their opinions in a distributed manner, clustering is conceived to be executed under a *replicated database* model, that assumes common global information about the whole network (e.g., using a centralized entity), as introduced in Section 4.2.1. These issues will be properly addressed in the following sections.

#### 4.3.4 Discussion

With the introduction of candidate correlation techniques, the satisfaction of the traceability framework has been partially illustrated, and some conclusions can be drawn in qualitative terms.

Firstly, it is necessary to highlight the similarities and differences between the three models presented: average consensus, Opinion Dynamics and clustering approaches. Following the interface of inputs and outputs established in Section 4.2.2, they all assume that agents deployed along the network (either physical or logical) gather information from their respective devices and end up calculating the vector  $x$ , which is the basic and quantitative input  $I_1$  to the correlation algorithm. From there, the three techniques leverage a different procedure to discern which anomalies are related, according to a different representation of the second input: the quantitative one ( $I_2$ ).

As for the average consensus and Opinion Dynamics, this piece of knowledge can be represented with a weight assignation between agents. In both techniques, this weight also describes the network topology, in such a way that agents associate zero influence to others if there is not a direct link that connects them within the network. Additionally, the Opinion Dynamics accepts further criteria to specify this influence value based on the closeness between opinions. On the contrary, the clustering approaches do not usually take into account the network connectivity as standalone techniques. A potential workaround is the modification of the own algorithms, or to provide such information in form of additional data items before applying cluster analysis.

In any case, once these data inputs have been preprocessed by agents, the three solutions considered put into practice an iterative process of knowledge discovery until all anomalies are correlated and the infrastructure resources are grouped into different alert levels. This way, we say that the system reaches equilibrium when the agents converge after a set of iterations, resulting in a consensus, polarization or fragmentation of the network. Formally speaking, a consensus happens when there is a single value that represents the overall sentiment for all agents, as in the average consensus model introduced previously. In contrast, the polarization is the state that only presents two clusters of agents, whereas the fragmentation occurs when there are more than two clusters. Altogether, this indicates the information represented with the first ( $O_1$ ) and second output ( $O_2$ ) of the traceability solution, as we can not only identify whether a device is actually compromised (and to what extent), but also check which devices are experiencing the same threat. It is worthy of note that this is not completely doable in the case of average consensus, as the algorithm only produces the arithmetic mean of the anomaly values measured by the agents. Although this matches with  $O_2$  (since it provides information at a global level), it offers no distinction between areas, which makes impossible to pinpoint the compromised nodes and leave out all those that are unaffected. Nevertheless, the underlying consensus concept from the original approach still poses much interest when instantiated in the Opinion Dynamics model, allowing the fragmentation of opinions that are formed through different criteria.

For this reason, just the Opinion Dynamics and the clustering approaches stand out as solutions that completely fit the specification of  $O_1$  and  $O_2$ . The question remains as to how these techniques are able to provide information to meet the third output argument, which allows the assessment of the evolution of events and the study of the persistence of attacks. As addressed in further sections, this functionality revolves around recording the fluctuation of the  $O_1$  and  $O_2$  values over time. Prior to that, we require a detailed analysis of the Opinion Dynamics and clustering approaches to deeply define how they can be instantiated for APT detection while extending their algorithms to overcome their issues and accurately accommodate both outputs. This is carried out in next section. At the end of this chapter, both techniques will be finally compared in quantitative terms, to analyze their accuracy in different attack scenarios with a theoretical model before applying them to real industrial scenarios.

## 4.4 Adapting the Opinion Dynamics Model

In terms of adapting this multi-agent algorithm to our particular scenario, the main question that appears is with regards to the representation of the weight given by each agent to its respective neighbors, in order to consider their influence on the opinion about the severity of the incidence detected. The original approach is based on a simple criterion to choose the weight assigned among agents, as explained in the previous section: the closer two opinions of two connected nodes are (their values), the higher the weight assigned between them will be. This is known as the homophily quality: agents are more open to be influenced by neighbours that hold similar opinions

to themselves as opposed to others. In our approach so far, this means that, for every agent, the weight given to its neighbors is uniformly divided into those agents whose opinion is very similar to its own, considering a  $\varepsilon$  threshold for the difference between both values. Intuitively, this simulates the fact that agents located nearby with the same degree of anomaly sensed are prone to detect the same threat in their surroundings. Again, although this may be a valid criterion to model the weight, it could be enhanced to realistically reflect other environmental conditions involved (e.g., Quality of Service), as discussed in Section 4.3.2.

Note that some extensions of the Opinion Dynamics models in the literature already address this problem, by regulating the opinion influence for specific use cases in sociology scenarios. In the case of the Hegselmann-Krause model applied in this thesis, Chen et al. [267] extend it by including the concept of biased agents, which results in their ‘Social-Similarity-Based HK model’. In this version, for two agents that interact, they should not only hold a similar opinion, but also meet a criteria of social similarity. This measure comprises other attributes that must be close as well.

Here we will be inspired by this model to especially look into the security of the opinion exchange, regardless of the method used for the anomaly detection. In this regard, the formal approach does not provide details about how the agents transfer their opinions between them (using the data brokers presented in Section 4.2.1) or to a central correlator. However, if the same communication channels are used to deliver the Opinion Dynamics values, we must prevent against an attacker being able to compromise these links and potentially forge malicious opinions. Likewise, it would be also critical if the deployment of the Opinion Dynamics approach is fully distributed across the network, and the agents are physically integrated in the own industrial assets. In that case, besides assessing the security of each node, the algorithm could also take the QoS of the communication links into consideration to safely send this information, as well as to route other messages (e.g., commands or data) between the devices. Given this situation, in the following we propose a modification of the weight calculation mechanism to consider the security of the communication links and the confidence assigned to neighbors for the opinion transmission.

To begin with, we need to consider the original model: each agent  $i$  determines the weight given to every neighbor  $j$  in its neighborhood  $N_i$  through this expression:

$$w_{ij} = 1/N'_i \quad (4.5)$$

where  $N'_i$  is the subset of neighbors of  $N_i$ , whose difference in opinion with agent  $i$  is below  $\varepsilon$ . Otherwise,  $w_{ij}$  becomes zero. Even though this is just a criterion to reflect the degree of similitude between agents, it lacks much accuracy since it leaves behind several other aspects involved; in this case, we want to introduce an additional factor to regulate this weight through considering the QoS of the channel in the neighborhood.

Let  $\mathcal{S}: E \rightarrow \mathcal{R}$  be a function that assigns QoS scores to communication links in the network defined by  $G(V, E)$ . The higher the score of  $\mathcal{S}$  for a given link is, the more QoS it provides. For a

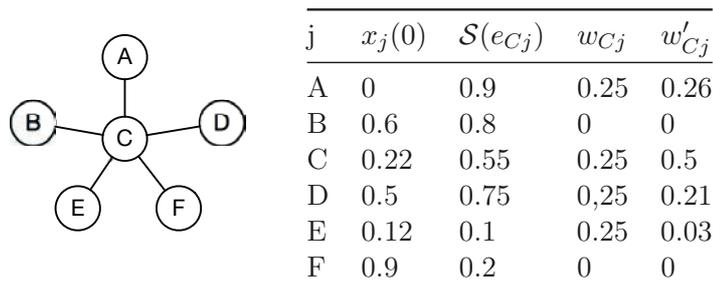


Figure 4.11: Example of weight calculation by agent C

given  $i$ , we aim to fairly distribute  $w_{ij}$  by giving a higher value to those agents  $j$  whose  $\mathcal{S}(e_{ij})$  is greater, where  $e_{ij} \in E$  represents the bidirectional communication link between  $i$  and  $j$ . This methodology complies with the following three conditions:

- **C1.** The sum of weights given by agent  $i$  to the neighbors in  $N'_i$  must be 1, also considering threshold  $\varepsilon$ .  $\sum_{j=1}^{N'_i} w_{ij} = 1$ .
- **C2.** The own agent  $i$  must have a sensitive fixed weight assigned to itself. For instance, we can assume  $w_{ii} = 0.5$ . The reason is that it is not fair that it associates a higher level of confidence to any other agent, whose link of communication can be minimally compromised.
- **C3.** The rest of weight (1/2 in this case) assigned by agent  $i$  is distributed among neighbors in  $N'_i$  proportionally to the quality of their communication links. If we define  $q = \sum_j^{N'_i} \mathcal{S}(e_{ij})$ , then the resulting weight value is defined by  $w_{ij} = (1 - w_{ii}) * \mathcal{S}(e_{ij})/q$ .

*Example.* The table in Figure 4.11 shows the calculation of  $w_{ij}$  for the node  $C$  in the example graph (where  $i = C$ ) following the proposed methodology, compared to the original one. The weight value that is computed using the new methodology is denoted by  $w'_{ij}$ . In both cases, a value of  $\varepsilon = 0.35$  has been considered. As we can see, the new distribution of weight results more equitable, where node  $C$  assigns a higher weight to nodes  $A$  and  $D$ , since their links show a better quality and security (which is represented by the  $\mathcal{S}(e_{Cj})$  column).

## 4.5 Adapting the Clustering Models

As briefly introduced in Section 4.3.3, the clustering models present some constraints to adapt the correlation to a scenario of distributed agents, to provide the same services as the Opinion Dynamics solution. In specific, the parametrization of this kind of algorithm imposes two main challenges to properly comply with the inputs and outputs of the APT traceability framework:

- **The election of  $k$ .** It is one classical drawback of the K-means, since that value has to be specified from the beginning and it is not usually known in advance, as in this case.

Numerous works in the literature have proposed methods for selecting the number of clusters [268], including the use of statistical measures with assumptions about the underlying data distribution [269] or its determination by visualization [270]. It is also common to study the results of a set of values instead of a single  $k$ , which should be significantly smaller than the number of instances. The aim is to apply different evaluation criteria to find the optimal  $k$ , such as the Calinski and Harabasz score (also known as the Variance Ratio Criterion) [271], that minimizes the within-cluster dispersion and maximizes the between-cluster dispersion.

- **Representation of topological and security constraints.** By applying K-means, we assume the dataset consists of a set of multi-dimensional points. However, here we have an one-dimensional vector of anomalies in the range  $[0,1]$ . Also, the clusterization of these values is subject to the topology and the security correlation criteria which might determine that, for example, two data points should not be grouped in the same cluster despite having a similar anomaly value. Therefore, it becomes necessary to provide this knowledge to the algorithm and reflect these environmental conditions as inputs ( $I_1$  and  $I_2$ ) to the correlation. In this sense, some works have proposed a constrained K-means clustering [272], and specific schemes have been developed to divide a graph into clusters using Spanning Trees or highly connected components [273].

As for the first challenge, we can assume that the value of  $k$  is defined by the different classes of nodes within the network depending on their affection degree, which corresponds to the number of consensus between agents that Opinion Dynamics automatically finds. Here we can adopt two methodologies: (1) a *static* approach where we consider a fixed set of labels (e.g., ‘low’, ‘medium’, ‘high’ and ‘critical’ condition) to classify each agent; or (2) a *dynamic* approach where  $k$  is automatically determined based on the number and typology of attacks. In this case, we can study the Variance Ration Criterion in a range of  $k$  values (e.g.,  $k=\{1-5\}$ ) to extract the optimal value with the presence of an APT.

This procedure needs further improvements to make the solution fully distributed, so that each agent is in charge of locally deciding its own level of security based on the surrounding state, instead of adopting a global approach for all nodes. This bring us to the second challenge. A first naive solution would be to introduce additional dimensions to the data instances representing the coordinates of every node, together with the anomalies in vector  $x$ . We call this approach *location-based clustering*. However, this approach still needs to figure out an optimal value of  $k$ , and does not take into account the presence of actual links interconnecting nodes in  $G(V, E)$ .

To circumvent this issue while also adopting an automatic determination of the number of clusters, we propose an *accumulative anomaly clustering* scheme, which is formalized in Algorithm 2. This algorithm begins by selecting the most affected node within the network and subsequently applies the influence of their surrounding nodes. This is represented by adding an entire value to the anomalies of such agents (initially from 0 to 1), which is proportional to the anomaly of the influencing node (see *max* in the algorithm). This addition is performed as long as the

---

**Algorithm 2** Accumulative anomaly clustering

---

**input:**  $x_i$  representing the initial anomaly value sensed by each agent  $i$  within the network, where  $x_i \in (0, 1)$   
**output:**  $z_i$  representing the agents  $O_1$  output of each agent  $i$  after clustering  
**local:** Graph  $G(V, E)$  representing the network, where  $V = V_{IT} \cup V_{OT} \cup V_{FW}$

```

max ← |V|, k ← 0
y ← x, x' ← x sorted in descending order
for all i ∈ x' do
    anyNeighbourFound ← False
    for all j ∈ neighbours(i, G) do
        if y_j ≤ 1 AND |y_i - y_j| ≤ ε then
            y_j ← y_j + max * 10
            anyNeighbourFound ← True
        end if
    end for
    y_i = y_i + max * 10
    if anyNeighbourFound then
        k ← k + 1
    end if
    max ← max - 1
end for
clusters, centroid ← kmeans(y, k)
for all v_i ∈ V do
    c ← clusters(v_i)
    Z_i ← IntegerPart(centroid(c))
end for

```

---

difference between both anomalies (i.e., the influencing and influenced node) does not surpass a defined threshold  $\epsilon$ , similar to the Opinion Dynamics approach in order to comply with  $I_2$ . Then, the algorithm continues by selecting the next one in the list of nodes inversely ordered by the anomaly value, until all nodes have been influenced or have influenced others. At that point,  $k$  is automatically assigned with the number of influencing nodes, and K-means is ready to be executed with the modified data instances. The resulting values of each agent correspond to the decimal part of their associated centroid. This is comparable to the ‘opinions’ in the Opinion Dynamics approach.

The intuition behind this model of influence between anomalies (which can be enriched to include extra security factors to specify  $I_2$ ) assumes that successive attacks raise a similar anomaly value in the closest agents, as Opinion Dynamics suggests. At the same time, it addresses the issue of selecting  $k$  and including topological information to the clusterization. It is validated from a theoretical point of view in Section 6.1.

In the following, we give answer to the question raised before the analysis of the opinion Dynamics and the clustering approaches, with respect to the fulfillment of the third output of the

framework, related to the traceability of events by both proposals. Afterwards, the accuracy of these two correlation approaches will be compared under different attack and network configurations.

## 4.6 Common Traceability Features

After formally representing the attack stages, plus their relation to the detection probabilities, we can now use the proposed detection probabilities as inputs to the correlation algorithm, and hence simulate its response in an industrial architecture when it faces a particular instance of APT.

Algorithm 3 describes the life cycle of an APT composed by a set of attack actions against a given network. Each of these attacks generates an anomaly that is detected by the corresponding agents (and possibly by their neighbors), increasing their opinion in a value defined by the previously introduced  $\Theta$ . After this, as commented in earlier sections, we also introduce an attenuation value on the quantitative input that represents the effect of old attacks in order to reduce their influence when computing the current opinion. This ‘decay’ value, applied in the `UPDATEOPINIONSWITHDECAY` function of Algorithm 3, depends on the attack stages suffered in the past by the agent and the criticality of its monitored device: the more devastating the alert generated is (during the detection phase), the longer its effect will take to disappear. Consequently, we define  $\Phi$  as an ordered set of decay values, where  $\Phi = \{\phi_1, \dots, \phi_d\}$  and  $\phi_i = [0, 1]$ , such that  $\forall \phi_i, \phi_i < \phi_{i+1}$ . Therefore, for all  $i \in d$ ,  $\phi_i$  is inversely proportional to the  $\theta_i$  value, and both are applied to the detected anomaly value after each stage. This procedure, explained in Algorithm 4, is a way to account for the persistence when computing the correlation algorithms. It is important to note that both the respective anomaly and decay addition or reduction implies a normalization of the opinion value, from 0 to 1.

Once the  $x$  vector of opinions is updated with the new attack action (with  $\theta$ ) and attenuated due to old stages (through  $\Phi$ ), the correlation algorithm (i.e., Opinion Dynamics or clustering-based) is executed to identify the affected areas of nodes and the level of severity of these attacks. However, although this gives insight of the location of threats (as it is visualized in the experimentation section), it would be also necessary to obtain an overall value of the network health. Therefore, we have created the so-called delta indicator, which represents a global anomaly value and is computed in the `COMPUTEDELTA` function. This value is calculated with the weighted average of opinions by the amount of agents that hold the same detected abnormality, as described in Algorithm 5. However, since this aggregated value is dependent on the number of agents to calculate the average, in practice we can compute it over different sections of the network (i.e., IT or OT), thereby increasing its granularity. Using these values, we can quickly know the overall anomaly degree of every portion of the network.

In the following, we present a test case for illustrating how we can apply the Opinion Dynamics-based technique while representing an APT against a given IT/OT industrial topology, as described



**Algorithm 3** APT life cycle - anomaly calculation

---

**output:**  $\delta$  representing the delta value  
**local:** Graph  $G(V, E)$  representing the network, where  $V = V_{IT} \cup V_{OT} \cup V_{FW}$   
**input:**  $attackSet \leftarrow attackStage_{APT_x}$ , representing the APT chain of attack actions

$x \leftarrow zeros(|V|)$  (initial opinion vector)  
{performedAttacks  $\leftarrow \emptyset$ }  
{attack  $\leftarrow firstattackfromattackSet$ }  
**while**  $attackSet \neq \emptyset$  **do**  
  **if**  $attack == initialIntrusion_{(IT, OT, FW)}$  **then**  
     $attackedNode \leftarrow random\ v \in V_{(IT, OT, FW)}$   
     $x(attackedNode) \leftarrow x(attackedNode) + \theta_3$   
  **else if**  $attack == compromise$  **then**  
     $x(attackedNode) \leftarrow x(attackedNode) + \theta_2$   
    **for** neighbour **in** neighbours(attackedNode) **do**  
       $x(attackedNode) \leftarrow x(attackedNode) + \theta_5$   
    **end for**  
  **else if**  $type(attack) == LateralMovement$  **then**  
     $previousAttackedNode \leftarrow attackedNode$   
     $attackedNode \leftarrow SELECTNEXTNODE(G, attackedNode)$   
     $x(previousAttackedNode) \leftarrow x(previousAttackedNode) + \theta_5$   
     $x(attackedNode) \leftarrow x(attackedNode) + \theta_{3,4}$   
  **else if**  $attack == exfiltration$  **then**  
     $x(attackedNode) \leftarrow x(attackedNode) + \theta_4$   
  **else if**  $attack == destruction$  **then**  
     $x(attackedNode) \leftarrow x(attackedNode) + \theta_1$   
  **else if**  $attack == idle$  **then**  
    No attack performed  
  **end if**

$x \leftarrow UPDATEOPINIONSWITHDECAY(x, performedAttacks)$   
 $performedAttacks \leftarrow performedAttacks \cup attack$   
 $mergedOpinions \leftarrow COMPUTECORRELATION(x)$   
 $\delta \leftarrow COMPUTEDELTAM(mergedOpinions)$   
 $attackSet \leftarrow attackSet \setminus attack$   
**end while**

---

**Algorithm 4** Decay of anomaly values over time depending on the attack action

---

```

function UPDATEOPINIONSWITHDECAY( $x, performedAttacks$ )
  for attack in performedAttacks do
    affectedNode  $\leftarrow$  GETAFFECTEDNODE(attack)
    if attack == initialIntrusionIT,OT,FW then
       $x(affectedNode) \leftarrow x(affectedNode) - \phi_3$ 
    else if attack == compromise then
       $x(affectedNode) \leftarrow x(affectedNode) - \phi_2$ 
      for neighbour in NEIGHBOURS(affectedNode) do
         $x(affectedNode) \leftarrow x(affectedNode) - \phi_5$ 
      end for
    else if type(attack) == LateralMovement then
      origin  $\leftarrow$  GETORIGINOFMOVEMENT(attack)
       $x(origin) \leftarrow x(origin) - \phi_5$ 
       $x(affectedNode) \leftarrow x(affectedNode) - \phi_{3,4}$ 
    else if attack == exfiltration then
       $x(affectedNode) \leftarrow x(affectedNode) - \phi_4$ 
    else if attack == destruction then
       $x(affectedNode) \leftarrow x(affectedNode) - \phi_1$ 
    end if
  end for
  return  $x$ 
end function

```

---

**Algorithm 5** Computation of delta value

---

```

function COMPUTEDELTA( $mergedOpinions$ )
  opinionClusters  $\leftarrow$  UNIQUEVALUES( $mergedOpinions$ )
  frequencyVector  $\leftarrow$  zeros(|opinionClusters|)
  for i:=1 to size(opinionClusters) step 1 do
    frequencyVector(i)  $\leftarrow$  COUNTOCCURRENCESOFOPINION(opinionClusters(i),
    mergedOpinions)
  end for
   $\delta \leftarrow 0$ 
  for j:=1 to size(opinionClusters) step 1 do
     $\delta \leftarrow \delta + frequencyVector(j) * uniqueValues(j)$ 
  end for
   $\delta \leftarrow \delta / size(mergedOpinions)$ 
  return  $\delta$ 
end function

```

---

$i$	1	2	3	4	5
$\theta_i$	0.9	0.7	0.5	0.3	0.1
$\phi_i$	0.01	0.025	0.05	0.075	0.1

Table 4.3: Detection probability and decay values used in the Stuxnet test case

before. For this test case, we have implemented the network topology and Algorithms 3, 4 and 5 in Matlab.

Let us assume that we have a topology composed by three OT nodes and three IT nodes connected by a firewall, as explained in Section 4.1.1. We will consider Stuxnet for the attacker model, since it is one of the most documented APTs in the literature. According to Section 4.1.3, it comprises a set of nine different attack actions that will be perpetrated against the proposed network, where each node counts on an individual agent to monitor its anomalies. If we execute the Opinion Dynamics algorithm after each stage, we can analyze the different clusters of anomalies detected by sets of agents. Following the model presented in Section 4.1.3, we have assigned values for each  $\theta$  and  $\phi$  according to the ordered set of probabilities in Table 4.3, considering a realistic scenario. We have also introduced a deviation of 0.1 to values in  $\theta$  to simulate a low level of noise or probability of detecting the corresponding anomaly after each attack stage. Figure 4.12 visually represents the resulting values in each agent after the four of the most representative stages, where (1) the attacker compromises the IT node and exfiltrates information, (2) compromises the firewall and then (3) moves to the last OT of the network and remains idle, right before the destruction of this node is performed (4). Four different idle operations are performed in this point, with a total of twelve attack actions. Numbers by the name of nodes represent the value of anomaly (opinions) that each agents holds.

As we can also see in Figure 4.12, the attacker traverses the whole network according to the Stuxnet behavior (where the current attacked node appears rounded), while the agents and its neighbors are able to detect the anomalies that consequently take place (the more red the node is, the greater the detected anomaly is). At the same time, we see how attenuation of anomalies also occurs, especially visible when the attacker leaves a node. In this example, the first IT node compromised is the number 1 while the final one is the OT number 3; the former is gradually attenuating its value as the attack evolves, according to the behavior explained in Section 4.6.

This ability to identify where the threat is active within the network is enabled by Opinion Dynamics. If we have a look at its value in form of a plot in some point, we obtain the graph in Figure 4.13. This corresponds to the execution of the algorithm (with 20 inner iterations) after the second stage depicted in Figure 4.12, where the FW is compromised after attacking the first IT nodes. As we can rapidly see in the resulting graph, there are two agents (the  $a_{FW}$  and the  $a_{IT}$  node) that successfully detect the same level of critical abnormality in their area; this is also detected by some of their neighbors mildly, which is represented with the central consensus. Apart from these, the rest of nodes only detect a negligible value of anomaly.

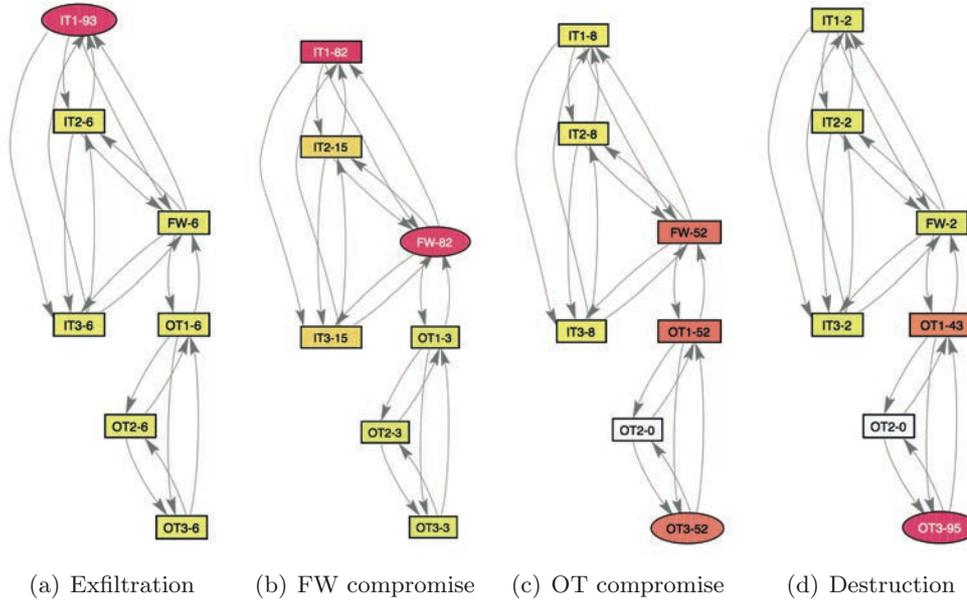


Figure 4.12: Execution of the Opinion Dynamics after multiple stages of Stuxnet

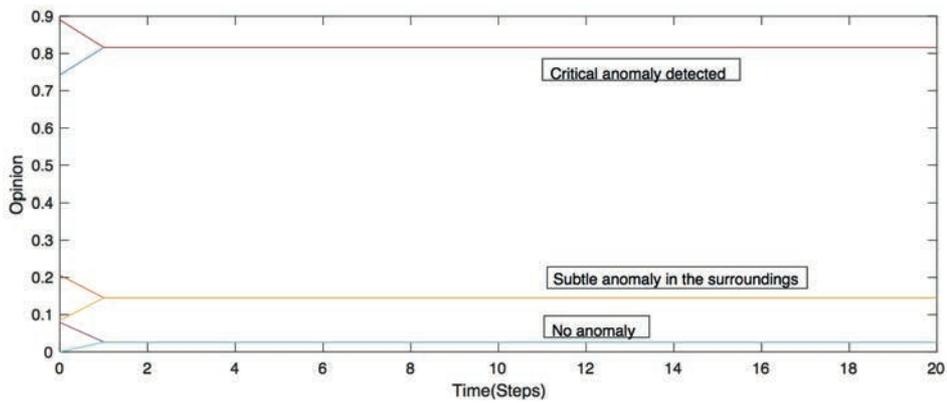


Figure 4.13: Opinion dynamics after the second stage

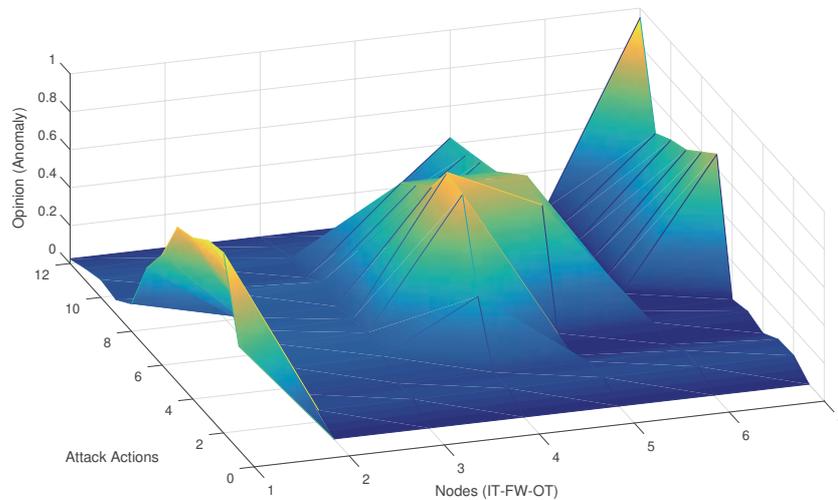


Figure 4.14: Evolution of the opinions over time to trace the APT stages

By this means, we can statically identify where the threat is located and which severity it experiences. However, as commented in Section 4.2, it would be also necessary to trace all the events of the APT and highlight the most affected nodes it has traversed. In this sense, if we represent the succession of opinions agreed by agents over time for the Stuxnet attack described previously, we easily have such information, which is represented with Figure 4.14.

As we can see there, the opinion profile for all agents evolves over the set of APT attack actions, showing a more pronounced value in the IT section in earlier stages and the OT in latter phases of the Stuxnet APT, as the attack aims to ultimately compromise a PLC by firstly intruding the network through a IT node. A similar effect is seen when we study the change in the delta value, which can be calculated either in the whole network or on any of its subnetworks (i.e., IT or OT). Figure 4.15 shows the progression of this indicator in each case, which also shows us how IT delta decreases over time and its value in OT increases according to the chain of attacks. In general, the value acquires the highest value when the last OT node is compromised, since the network has suffered most of the attacks in the previous stages. Beyond that point, delta decreases (due to the idle operations) and then it finally increases with the destruction of the node.

## 4.7 Comparison of Models

After presenting some alternative solutions to Opinion Dynamics that fulfill the distributed detection framework presented in Section 4.2, this section aims to put these approaches to the test. More specifically, we consider the attacker model explained in Section 4.1.3, which is applied against a network formalized by  $G(V, E)$ , following the structure introduced in Section 4.1.1.

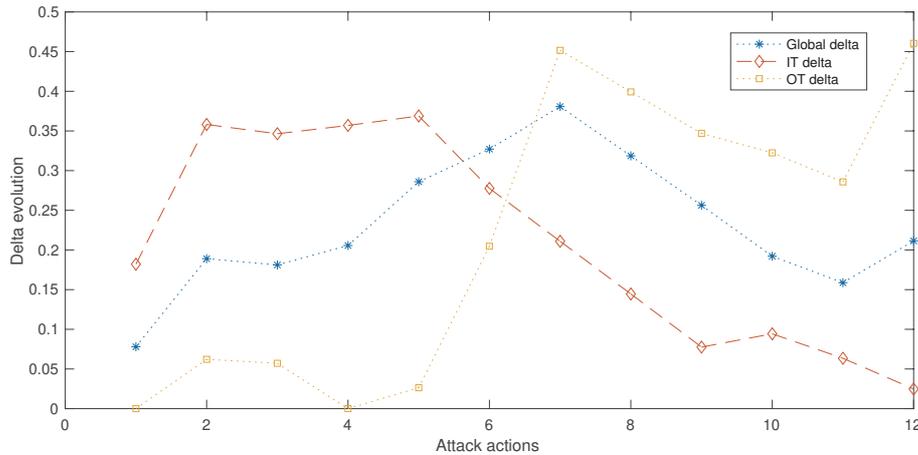


Figure 4.15: Evolution of delta opinions over the network for the Stuxnet attack

These theoretical APTs generate a set of anomalies that serve as input to compare the traceability capabilities of each correlation approach:

- **Location-based clustering:** as presented earlier, it consists of the K-means algorithm taking the anomalies and coordinates of each node as data instances. These are grouped in a number of clusters,  $k$ , which is selected in the range from 1 to 5 according to the Variance Ratio Criterion.
- **Accumulative clustering:** as previously presented, it allows to distributedly locate the infection while automatically determining the optimal  $k$ .
- **Opinion Dynamics:** is the approach that serves as inspiration for our framework and serves for comparison with the novel detection methods introduced above.

These traceability solutions are simulated under different network and attack configurations, as explained next. We start by running a brief attack test-case that illustrates the features of each approach in a simple network scenario. Based on Algorithm 3, Figure 4.16 shows the detection outputs ( $O_1$  and  $O_3$ ) of the three approaches when correlating the anomalies of an APT perpetrated against a simple infrastructure. This network is modelled according to the concepts introduced in Section 4.1.1, to include an IT and OT section of nodes connected by a firewall. Concretely, the figure shows an snapshot of the detection state after the adversary has performed a lateral movement from IT node 2 to compromise the firewall. The numeric value assigned to each node represents  $O_1$ , which will attenuate over time to highlight the most recent anomaly, according to  $O_3$ .

As noted in the figure, location-based clustering fails to accurately determine where the threat is located and selects a wide affection area instead, which is composed by IT1, IT2 and FW1

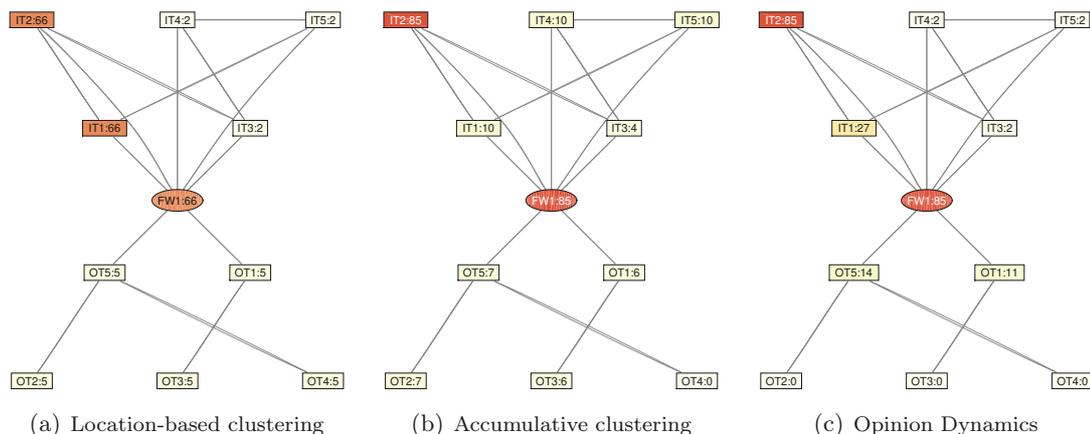


Figure 4.16: Network topology used in the test case

nodes (i.e., grouped in the same cluster due to the average anomaly in such zone). On the other hand, the accumulative clustering and Opinion Dynamics show a similar result, and successfully identify both IT2 and FW1 as the affected nodes in this scenario. As for the rest of nodes, they agree on a subtle affection value due to the noise present in the network and the anomalies sensed in the vicinity of the attacked nodes. As previously stated, this is modelled in a probabilistic way [82].

We now execute these solutions with a more complex network and APT model in order to study their accuracy. In the context of cluster analysis, the ‘purity’ is an evaluation criteria of the cluster quality that is applicable in this particular scenario [274]. It holds the percentage of the total number of data points that are classified correctly after executing the clustering algorithm, in the range [0,1]. It is calculated according to the following equation:

$$Purity = \frac{1}{N} \sum_{i=1}^k \max |c_i \cap t_j| \quad (4.6)$$

where  $N$  is the number of nodes,  $k$  is the number of clusters,  $c_i$  is a cluster in  $C$  and  $t_j$  is the classification that has the  $\max$  count for cluster  $c_i$ . In our case, by ‘correct classification’ we mean that a cluster  $c_i$  has identified a group of nodes that have actually been compromised, which is determined in the simulations (but not known by the traceability solutions). This value can be calculated after a single execution of these three approaches to study how the results of the initial test-case escalate to larger networks and more challenging APTs.

Specifically, we run 10 different APTs on randomly generated network topologies of 50, 100 and 150 nodes, respectively. For simplicity, we start by executing an individual instance of the Stuxnet APT [82] according to the attacker model established in Section 4.1.3. This attack can be formally defined by the following succession of stages:

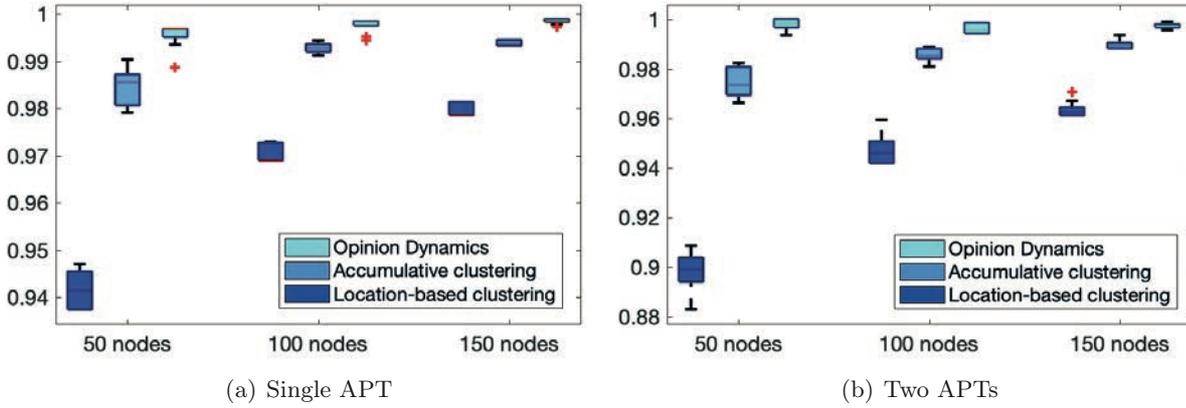


Figure 4.17: Purity average for the three test cases

$$attackSet_{Stuxnet} = \{initialIntrusion_{IT}, LateralMovement_{FW}, \\ LateralMovement_{OT}, destruction\}$$

At this point, it is worth mentioning that the lateral movement in the OT section is performed three times to model the real behavior of this APT and its successive anomalies. The purity value is then calculated after every attack stage of each of the ten APTs, to ultimately compute its average with respect to the number of nodes that have been successfully detected and grouped in the cluster with highest value of affection.

Figure 4.17(a) represents these average values in the form of boxplots, where each box represents the quartiles of each detection approach given the different network configurations. As it can be noted, the Opinion Dynamics stands out as the most accurate solution, closely followed by the accumulative clustering approach. The purity of the location-based clustering falls behind, and the three of them increase their value as the network grows in size due to the higher number of nodes that are successfully deemed as healthy, and hence not mixed with those that are indeed affected by the APT.

Similar results are obtained when we execute two APT attacks in parallel over the same network configurations, as shown in Figure 4.17(b). In this case, the former APT is coupled with another attack, which can be assumed to be part of Stuxnet or a completely different attack trace within the network, composed by the following stages:

$$attackSet_{AnotherAPT} = \{initialIntrusion_{OT}, LateralMovement_{FW}, \\ LateralMovement_{IT}, destruction\}$$

The second APT is located in a different area of the network so that it begins by sneaking into the OT section to subsequently propagate towards the IT portion of the infrastructure. This causes the spread of anomalies throughout the network hence putting location-based clustering to

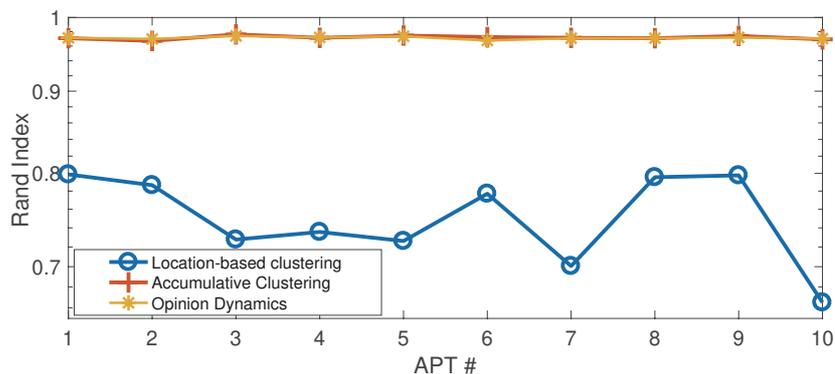


Figure 4.18: Evolution of the Rand Index for 10 APTs and 150 nodes

the test. Despite a subtle decline in the purity of the solutions (especially in the location approach due to the anomaly dispersion), they still output an appreciable accuracy.

On the other hand, the superiority of Opinion Dynamics and accumulative clustering over the first approach is also evident with the study of additional accuracy indicators, such as the *Rand Index* [275]. It penalizes both false positive (FP) and false negative (FN) labeling of affected nodes during clustering, with respect to true positive (TP) and true negative (TN) decisions, according to the following formula:

$$Rand\ Index = \frac{TP + TN}{TP + FP + FN + TN} \quad (4.7)$$

Figure 4.18 shows the Rand Index value after each of the ten APTs in the previous experiment (each one composed of two parallel attack traces), for the largest network size (150 nodes). The plot clearly shows a steady accuracy of the two latter approaches (close to 1), contrasting with a lower value in the location-based approach, which faces a lack of precision when it comes to correctly locating the affection areas, for the same reasons discussed before.

As a result of these tests, we can conclude that despite the fact that both techniques satisfy the specification of our framework, it is the Opinion Dynamics algorithm that shows higher accuracy in tracking complex attacks. As a consequence, we will use this technique in the next chapter to assess the effectiveness of our framework in different Industry 4.0 security scenarios. The main concepts, indicators and mathematical symbols related to the traceability framework and the Opinion Dynamics solution are summarized in Table 4.4 for future reference.

Table 4.4: Summary of concepts involved in the APT traceability framework

Term	Concept	Definition
$\alpha$ and $\beta$	Parameters of the Power Law Out Degree (PLOD) algorithm to guide the construction of the $G(V_{OT}, E_{OT})$ graph	Whereas $\beta$ controls the y-intercept of the curve, the value of $\alpha$ controls how steeply the curve drops off.
$\Theta = \{\theta_1, \dots, \theta_d\}$	Ordered set of detection probabilities of size $d$ , where $\theta_i = [0, 1]$	It defines how the $d$ attack stages in an APT influence the calculation of the detection probabilities
$\Phi = \{\phi_1, \dots, \phi_d\}$	Ordered set of decay values of size $d$ , where $\phi_i = [0, 1]$	It defines how the $d$ attack stages in an APT attenuate their influence over time, depending on their persistence
$x$	vector $x$	$x_i$ represents the anomaly value sensed by the corresponding agent on device $i$ , where $x_i = [0, 1]$ for all $i \in 1, 2, \dots,  V $
$I_1$	Quantitative input of the traceability framework	It assigns every industrial asset with an anomaly value prior to conducting the correlation (expressed with vector $x$ )
$I_2$	Qualitative input of the traceability framework	It is assigned by every agent $i$ to each of its neighbours $j$ to correlate events in nearby devices (expressed with weight $w_{ij}$ ).
$O_1$	Local result of the traceability framework	It determines the level of the infection of the node
$O_2$	Information provided by the traceability framework at global level	It determines the degree of affection in a zone of the network
$O_3$	Contextual information provided by the traceability framework	It permits to correlate past events and visualize the evolution of the threat
$\mu$	$\mu$ indicator after executing the traceability solution	It holds the ratio of agents that find a consensus on the amount of degree experienced
$\delta$	$\delta$ indicator after executing the traceability solution	It holds the overall value of the network health

## Chapter 5

# Protecting Industry 4.0 Scenarios against APTs and Use Cases

The APT traceability framework proposed in the previous chapter considers various network architectures, types of attack and data acquisition models to later define the inputs and outputs that solutions should include to support the detection and security requirements. This lays the base for the development and comparison of novel solutions in this context. As a means to validate the proposed framework, we have defined two novel protection mechanisms based on clustering and Opinion Dynamics. According to our theoretical experiments, the latter features higher accuracy for the traceability of events in a distributed setting.

In this chapter, we put into practice the detection mechanisms designed previously, so as to look further into their precise application when applied to multiple scenarios of the Industry 4.0, with particular interest in the support for response techniques that circumvent and diminish the consequences of advanced persistent threats.

### 5.1 Ensuring the Survivability of the Network

As a means to check the usefulness of the APT traceability framework (via the Opinion Dynamics algorithm), we firstly explore the implementation of secure routing protocols. Our goal is to take advantage of the information about the security state of the network (and more specifically the outputs  $O_1$  and  $O_2$  of the correlation technique) to guarantee the continuity of the infrastructure in the presence of attacks.

More specifically, this functionality is designed and implemented through two response techniques with different objectives. The first one (addressed in this section) assumes the presence of an attacker who takes control of some network nodes, with the ability to intercept traffic (jeopardizing the information confidentiality) or directly deny service on some communication links. Based on this attacker model, we propose the deployment of a redundant network architecture, which allows the effective sending of messages between any sender and recipient node.

We can summarize our contributions as:

- Modeling the evolution of an APT composed of subtle attacks against the network topology (i.e., their communication links).
- Implementation of a multi-agent system for the detection of an APT based on the topological changes suffered in selected parts of the network, observed by hierarchically chosen nodes in accordance with controllability criteria.
- Use of redundancy edges and random routing protocols to overcome the network deformation provoked by the APT and to avoid compromised systems, ensuring the reachability between nodes and the survivability of the network.

The remainder of this section is organized as follows: firstly, we describe the threat model used for the APT. Then, the detection of these attacks is addressed by means of the Opinion Dynamics correlation, which has been demonstrated to be the most effective solution in Chapter 4. Based on this mechanism, response techniques are implemented and subsequently analyzed from a theoretical and experimental perspective.

### 5.1.1 Threat Model based on Topological Changes

Assuming a successful intrusion inside a network represented by a matrix  $M$ , we model an APT with a succession of attacks perpetrated on its topology. Specifically, just as an actual APT works (and inspired by findings described in Section 2.4.2), the attacker firstly selects one node and then makes several lateral movements in order to find new nodes to compromise. Since we want to provide realism in this model and consider a scenario of high criticality, we assume the attacker always seeks those nodes with more controllability, that is, those belonging to the DS and hence the ones with the highest *betweenness centrality* (whose concept was previously introduced in Section 4.1.1).

In each of the steps in its life cycle, the APT can commit individual attacks on the topology, i.e., changing the edges from the compromised node at a given time instant. This consequently generates a new matrix  $M'$ . The types of attacks can be:

- \* **Removal of an incoming edge:** given the vertex  $v_i$  that represents the compromised node such that  $v_j$  exists and  $M(j, i) = 1$ , it implies setting  $M(j, i) = 0$ .
- \* **Removal of an outgoing edge:** given the vertex  $v_i$  that represents the compromised node such that  $v_j$  exists and  $M(i, j) = 1$ , it implies setting  $M'(i, j) = 0$ .
- \* **Addition of an incoming edge:** given the vertex  $v_i$  that represents the compromised node such that  $v_j$  exists and  $M(i, j) = 0$ , it implies setting  $M'(i, j) = 1$ .



- \* **Addition of an outgoing edge:** given the vertex  $v_i$  that represents the compromised node such that  $v_j$  exists and  $M(i, j) = 0$ , it implies setting  $M'(i, j) = 1$ .

In a simple version of the APT, we suppose that the kind of the attack and the first node compromised within the network are chosen randomly. From that moment on, the attack migrates to the adjacent node with the highest *betweenness centrality*, simulating the fact that the attacker can perform a reconnaissance of the network when looking for potential victims that deal with higher loads of control traffic. The resulting attacker behavior is described in Algorithm 6. An example of an APT with three attacks over a defined network topology is depicted in Figure 5.1, where driver nodes are marked in black to show how the APT always migrates to vertices with higher controllability. Firstly, node 4 is selected and an outgoing edge is added towards node 2. Then, the attacker moves to node 6 and removes the edge coming from node 3. Then, since node 6 still has dominance, the attack stays there and removes the edge going to 7.

---

**Algorithm 6** Advanced persistent threat based on topological changes

---

**output:**  $M'$  representing the resulting matrix

**local:**  $M$  representing  $G(V, E)$ ,  $numOfAttacks$

$attackedNode \leftarrow \text{random } v_i \in E$

$M' \leftarrow M$

**for**  $i:=1$  **to**  $numOfAttacks$  **step 1 do**

$attack \leftarrow \text{randomAttack over } attackedNode$  (edge addition or removal)

update  $M'$  based on attack

$attackedNode \leftarrow \text{SELECTNEWATTACKEDNODE}(M, attackedNode)$

**if**  $attackedNode == null$  **then**

$attackedNode \leftarrow \text{random } v_i \in E$

**end if**

**end for**

**function**  $\text{SELECTNEWATTACKEDNODE}(M, node)$

$childNodes \leftarrow \text{vertexes } v_j | M(node, v_j) = 1$

$parentNodes \leftarrow \text{vertexes } v_k | M(v_k, node) = 1$

$candidates \leftarrow childNodes \cup parentNodes$

$maxCentrality := 0$

$attackedNode \leftarrow null$

**for** vertex  $v$  **in**  $candidates$  **do**

$centrality \leftarrow \text{CALCULATEBETWEENNESSCENTRALITY}(v)$

**if**  $centrality > maxCentrality$  **then**

$attackedNode \leftarrow v$

**end if**

**end for**

**return**  $attackedNode$

**end function**

---

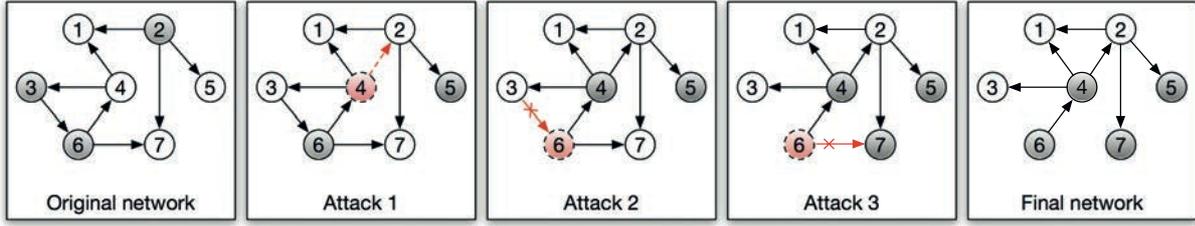


Figure 5.1: Example of APT with 3 attacks. 1st: Addition of edge from node 4 to node 2. 2nd: Removal of edge from node 3 to node 6. 3rd: Removal of edge from node 6 to node 7.

### 5.1.2 APT Response using Opinion Dynamics

After introducing the attack model, we execute the Opinion Dynamics algorithm assuming that  $x_i(0)$  will be theoretically calculated for each agent  $i$  as follows: let us suppose that  $BC(v_i)$  represents the original *betweenness centrality* for each agent  $i$  that, as explained, works as an indicator of the controllability of that particular node. If  $BC'(v_i)$  is the *betweenness centrality* of the same agent after being victim of a particular attack of those defined in Section 5.1.1 or another node in its neighborhood, we define the initial opinion  $x_i(0)$  as

$$x_i(0) = \frac{|BC'(v_i) - BC(v_i)|}{BC(v_i)} \quad (5.1)$$

Consequently,  $x_i(0)$  holds the ratio of change in the controllability of an agent  $i$  after an attack, compared to its initial state (due to an increase or decrease of adjacent edges). We assume that when the value was originally zero or the resulting ratio is greater than 1, the result is normalized to the value of 1.

Altogether, if we have the vector  $x(0)$  concerning the initial opinion of all agents in the DS, we can run the Opinion Dynamics algorithm to obtain a value of the change ratio of the network after suffering an individual attack, making it possible to distinguish between different clusters of agents with similar opinion. In this case, the closeness among opinions, which is represented by the matrix  $W$  with the weights assigned for each agent, has been modelled according to the difference in the degree of change (the individual opinion each agent holds): for two given agents  $i$  and  $j$ , if the difference is below a determined epsilon value (e.g., 0.3), they increase the weight given to each other; as explained in Section 4.3.2, this models the fact that agents that experiment a similar degree of change in their surrounding topology must agree on the presence of an anomaly in their respective area.

Once we have obtained a measure of the extent to which the network topology is at risk due to the effect of an APT attack, we are in a position to adopt multiple response techniques, which is also the aim of this section. We set the goal of preserving the connectivity for all those nodes in charge of delivering control signals to the rest of nodes of the network. According to the different

change ratios raised by the Opinion Dynamics algorithm, we can apply different techniques in separate nodes of the network.

Specifically, we suppose a scenario where we wish to ensure that one node  $i$  belonging to the DS wants to send control messages to another node  $j$  in the network. This is done in the presence of an APT that can remove certain edges that originally enabled both nodes to communicate over a defined path, traversing other points of the topology [276]. At the same time, we want to avoid hopping over compromised nodes that may be victims of the APT and hence intercept these sensitive packets, preserving confidentiality by this means. Moreover, it is desirable that the communication pattern (i.e., the paths described by the messages when being transmitted over the network) is as random as possible, so as to guarantee that the attacker cannot easily determine the topology of the network. As a result, we have a security service that ensures the continuity of the network until the APT has been successfully removed from the system. To sum up, we seek these three objectives when designing a response technique:

- (a) Ensure the presence of a path between node  $i$  and  $j$  when possible.
- (b) Define a routing protocol that prevents determining the path.
- (c) Introduce a mechanism to avoid the interception of messages.

To satisfy objective (a), we propose building an edge-redundant network with hidden edges that are added to the original network topology, so these auxiliary links can be leveraged in the event a path between two given nodes is lost after an APT attack. To accomplish this, we create a parallel network from  $G = (V, E)$ , which we name  $G' = (V, E')$ , where  $E'$  contains the same edges as  $E$  and includes new ones from the DS nodes to recover the controllability of the network. Specifically, we define and compare three different strategies:

- **STG1:** addition of redundant edges to all nodes in the network.
- **STG2:** addition of redundant edges only to DS nodes.
- **STG3:** addition of redundant edges only to nodes that are not included in the DS.

Our aim is to compare their level of response in terms of message loss and the overhead they experience. Algorithm 7 describes the procedure by which redundancy is added depending on the strategy selected. Namely, for each vertex, a set of candidates is created that includes the DS and excludes its parents and the node itself. In the case it is empty, the DS with maximum out-degree is selected as the new parent of the aforementioned vertex, creating a new edge by this means. It is important to note that during the process, it is ensured that the resulting network  $G' = (V, E')$  fulfills OR1 and OR2 conditions, as stated in Section 4.1.1.

On the other hand, to address objectives (b) and (c), we leverage a secret sharing scheme [277]: a secret (i.e., a control message) is divided into  $n$  shares or shadows that are distributed



---

**Algorithm 7** hiddenTopology( $G(V, E), DS, STG_x$ )

---

```

output ( $G' = (V, E')$ )
local:  $D_r \leftarrow \emptyset, E' \leftarrow E$ 

if  $STG_x = 1$  then
     $D_r \leftarrow V$ 
else if  $STG_x = 2$  then
     $D_r \leftarrow DS$ 
else if  $STG_x = 3$  then
     $D_r \leftarrow V - DS$ 
end if

for vertex  $v$  in  $D_r$  do
     $F \leftarrow Fathers^a(G(V, E), v)$ 
     $D_c \leftarrow DS - (F \cup v)$ 
     $Candidates \leftarrow \emptyset$ 
    for vertex  $c$  in  $D_c$  do
         $D \leftarrow Children^b(G(V, E), c) \cap DS$ 
         $O \leftarrow Children(G(V, E), c) - D$ 
        comment: checking of OR1 and OR2 fulfillment
        if  $v \in DS$  and ( $|O| \geq 2$  and  $|D| \geq 0$ ) or ( $|O| = 0$  and  $|D| \geq 1$ ) or ( $|D| = 0$  and
 $|O| = 0$ ) then
             $Candidates \leftarrow Candidates \cup c$ 
        else if  $v \notin DS$  and ( $|D| \geq 0$  and  $|O| \geq 1$ ) or ( $|D| = 0$  and  $|O| = 0$ ) then
             $Candidates \leftarrow Candidates \cup c$ 
        end if
    end for
    if  $Candidates = \emptyset$  then
         $Candidates \leftarrow MaxOutDegree^c(G(V, E), DS)$ 
    end if
    Arbitrarily select vertex  $c_1 \in C$ 
     $E' \leftarrow E' \cup (c_1, v)$ 
end for

```

---

<sup>a</sup>Selection of fathers of  $v$ , those belonging to its in-neighborhood

<sup>b</sup>Selection of sons of  $c$ , those belonging to its out-neighborhood

<sup>c</sup>Selection of the DS node with maximum out-degree

---

among the sender's neighborhood nodes and follow independent routes, so that the recipient cannot reconstruct the message until it collects, at least, a defined number  $k$  of them, where  $1 \leq k \leq n$ . In the case we have  $k = 1$ , it can be considered as the basic level of security, as the message in clear is sent over a determined path over the network. If we have  $k = n$ , then the recipient must collect all the shares to reconstruct the original message. At this point, since our aim is to provide a security mechanism that bases its robustness on the criticality of the attack detected, the election of  $n$  will depend on the number of DS agents whose opinion is similar, for which we make use of the  $\mu$  value defined in Section 4.3.2. Namely, the maximum number of shares to divide the original message into depends on the ratio of agents that have experienced the same severity in the attacks against their surrounding nodes. This means that the greater the number of DS that experience the same criticality, the greater the number of shares. However, the  $k$  value can be random (ranging from 1 to  $n$ ) in order to make the recovery method as stochastic as possible and thereby not leak any information about the topology when analyzing the stream of messages. The resulting methodology, to divide the messages into shares and send them over the network when it has been attacked, and Opinion Dynamics has been executed, is described in Algorithm 8. It is important to note that the respective shares are arbitrarily sent over the original and redundant links, in order to make the protocol as misleading for the attacker as possible. An example is shown in Figure 5.2, in which shares are divided and distributed over the network leveraging a pathfinding algorithm (e.g., *Dijkstra*, *Breadth-first search (BFS)*) [278][279]. In that example, the secret is divided into three shares with  $k = 2$ .

---

**Algorithm 8** SecretSharing( $G(V, E')$ )

---

**local:**  $M$  representing the set of messages to be sent.  
**for** message  $m$  **in**  $M$  **do**  
     $agent \leftarrow GetRecipient(m)$   
     $mu \leftarrow GetMu(agent)$   
     $n \leftarrow mu * |N_{agent}^{out}|$   
     $k \leftarrow generate\ random\ from\ 1\ to\ n$   
     $S \leftarrow divideSecret(m, n, k)$   
    send shares to  $n$  neighbours  
**end for**

---

This way, we have modelled the behavior of an APT against a control network represented with a graph, over which we have applied structural controllability concepts to define a dominance set of nodes (i.e., the DS). These take the role of agents that make a distributed decision algorithm determine the health of the network based on topological changes detected in their neighbourhood. From this information, they can leverage a parallel hidden topology with redundant links, over which they can continue to deliver their messages with enhanced privacy in the presence of the APT.

In the next subsection, we offer experimental results to show how effective it is when ensuring the continuity of the network.

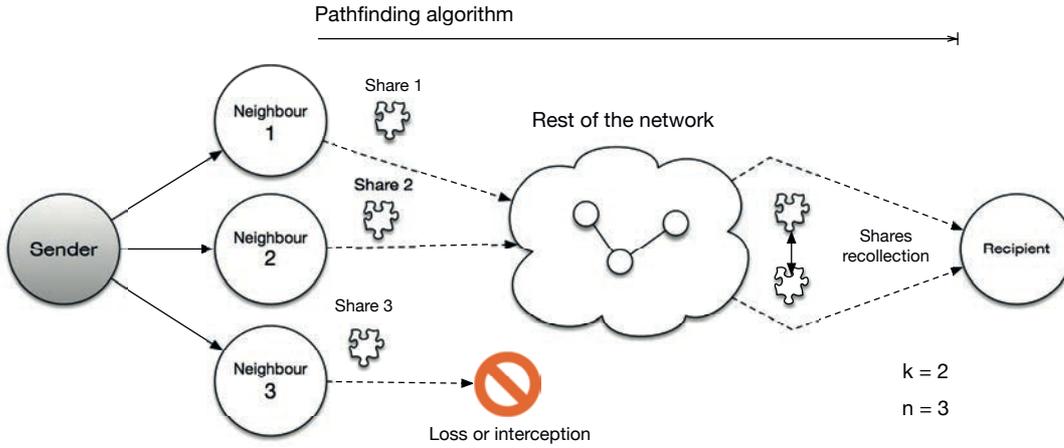


Figure 5.2: Secret sharing scheme and shares delivery

### 5.1.3 Experimental Analysis

After successfully detecting topological changes by using a distributed consensus algorithm and consequently deploying a response technique to ensure the continuity and preserve privacy in the network, our aim is to test these services in practice. We have conducted the implementation in MATLAB of an APT that follows the behavior described in Algorithm 6. After each attack of the sequence, Opinion Dynamics is executed on those agents belonging to the DS, which is calculated based on Algorithm 1. If we run different test cases, we can check how the opinion of agents evolve to reach a consensus with each other and form different clusters within the network. Figure 5.3 shows how the total number of DSs of three different networks (of 100, 200 and 300 nodes) is divided into substantial sets depending on the degree of change they experience after suffering a battery of 50 attacks. It is especially significant to note the presence of a big cluster in each of the three test cases, which indicates an important effect of the APT (of approximately 0.35, 0.25 and 0.45 ratio of change).

Opinion dynamics influences the  $\mu$  value that regulates the number of shares in which the secret messages are divided and distributed from each DS node to the rest of the network to their destination, as explained before.

To probe the effectiveness of our response technique that leverages a hidden topology comprising additional edges, we have generated a set of 100 messages whose sender belongs to the DS and the recipient is any other node within the network. Following the secret sharing scheme of Algorithm 8, each agent divides the message and gives each part to the corresponding neighbors, which are responsible for the delivery by leveraging the BFS algorithm. The path is calculated at each hop when traversing all nodes until the destination, since the topology can change over time, caused by the APT. In the event that the recipient is unreachable from a certain node at a given time, we consider that share to be lost. Consequently, taking into consideration the scheme, we deem a

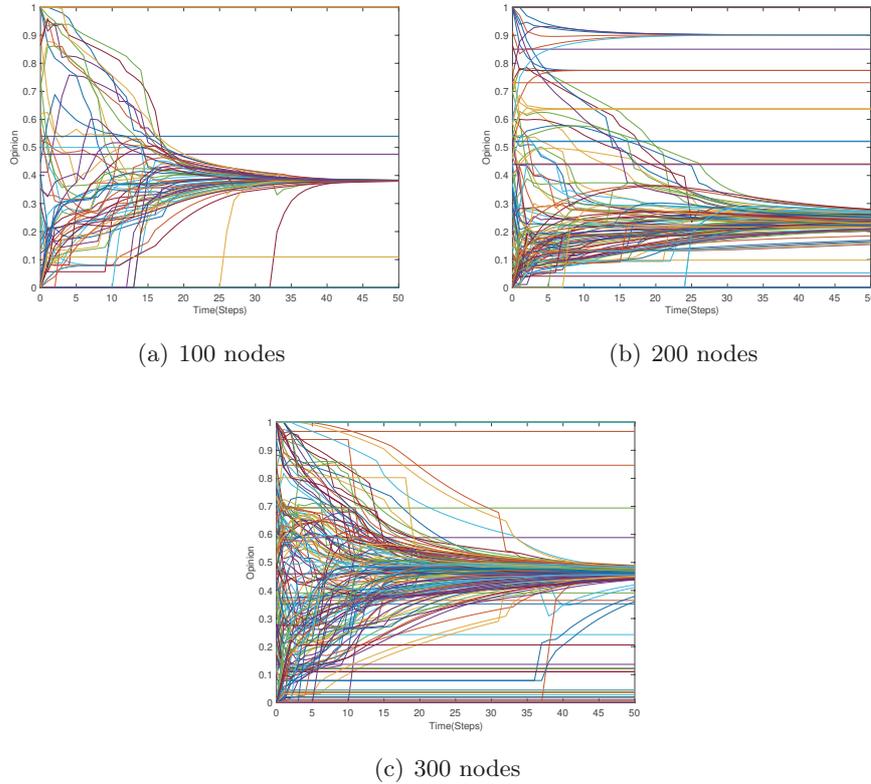


Figure 5.3: Opinion dynamics after 50 attacks

message to be lost if a number of its shares greater than  $k$  have been lost, since it is no longer possible for the recipient to construct the message.

Figure 5.4 shows the ratio of errors (i.e., message loss due to the unavailability of control paths) when using the normal and the hidden topology networks of STG1, STG2, and STG3. In more detail, we have run three test cases with a network of 100, 200 and 300 nodes, against which we perpetrate an APT of 50 attacks. Prior to executing it, we craft a set of 100 random messages for which we ensure the availability of paths from the sender's neighbors to the recipient. From that point on, the attacks take place and we try to send the original messages after each one. As a result, we can check how the loss ratio fluctuates as attacks occur. In this sense, based on the plots, the original network presents a higher quota of lost messages, whereas applying STG1 (i.e., a redundant edge for all nodes) experiments the lowest ratio, as expected in principle. However, we can see how redundancy in DS nodes (STG2) also achieves an acceptable degree of message reachability, comparable with STG1 and even better at certain points when running the same experiment in different topologies, as Table 5.1 indicates. This can be explained by the fact that attacks always move to nodes that deal with more traffic and hence have higher controllability (i.e., the DS nodes, as described in Section 4.1.1). Therefore, the addition of extra links to recover

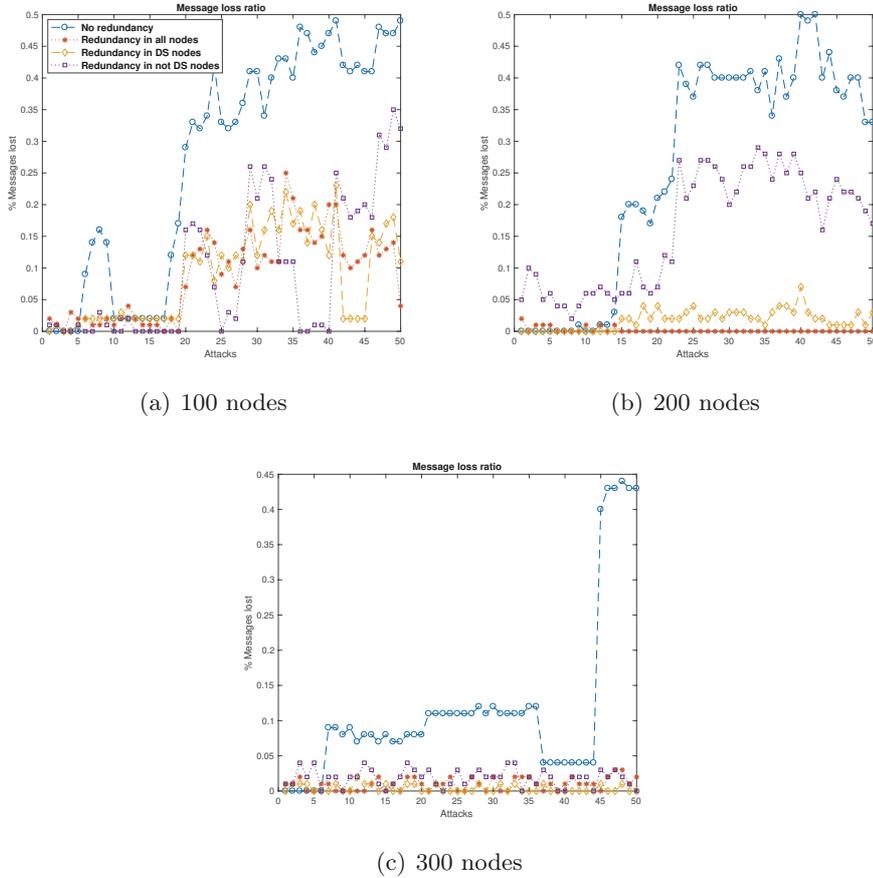


Figure 5.4: Message loss ratio with the different strategies, 100 messages and 50 attacks over a network of 100, 200 and 300 nodes

the connectivity between DSs results in a robust response that, on the other hand, does not introduce too much overhead because of the lower number of additional edges added.

The supremacy and higher connectivity of STG2 and especially STG1 are visible when analyzing the network *global efficiency* [280]. This measure indicates the efficiency of the information exchange in the network and how resistant it is to failures. If the distance  $d(i, j)$  between any two vertices  $i$  and  $j$  in the graph is defined as the number of edges in the shortest path between  $i$  and  $j$  such that  $i \neq j$ , the efficiency is expressed as  $1/d(i, j)$ . From this definition, the global efficiency of a graph is the average efficiency over all  $i \neq j$ . Figure 5.5 shows the evolution in this indicator when performing an APT attack over the original and redundant topologies, for the three test cases of 100, 200 and 300 nodes.

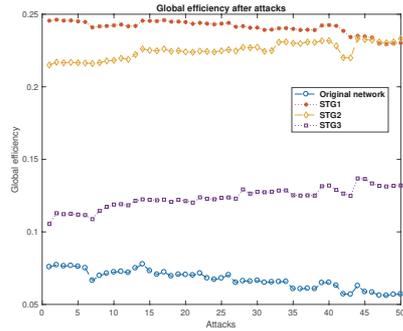
As a conclusion, we conclude that the Opinion Dynamics technique in combination with a network redundancy policy yields significant results in protecting the delivery of information throughout the infrastructure.



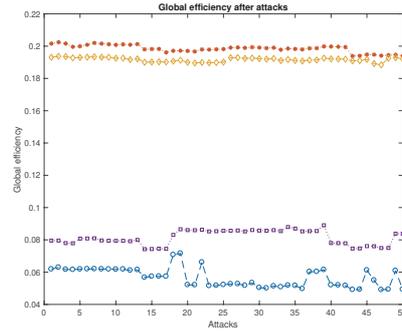
5.1. Ensuring the Survivability of the Network

Table 5.1: Message loss ratio after 50 attacks, 100 messages and multiple topologies

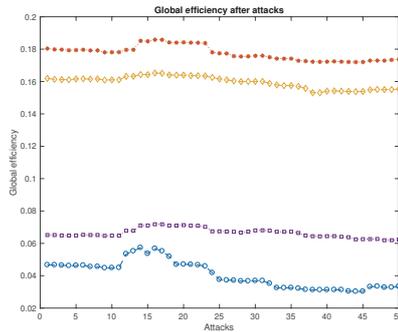
Nodes \ Strategy	Original network	STG1	STG2	STG3
10	0.81	0.64	0.64	0.58
50	0.076	0.25	0.3	0.36
100	0.62	0	0.03	0.01
200	0.24	0.04	0.02	0.14
300	0.71	0.02	0	0.21
400	0.07	0.04	0.02	0.03
500	0.39	0.1	0.07	0.19
600	0.4	0.05	0.02	0.03
700	0.32	0.03	0.07	0.1



(a) 100 nodes



(b) 200 nodes



(c) 300 nodes

Figure 5.5: Global efficiency with different strategies after 50 attacks



## 5.2 QoS-Aware Routing protocol based on Opinion Dynamics

In the previous section, we have illustrated a practical example of how to represent the quantitative input of the detection technique in form of anomalies detected in the topology of the network, according to the framework description of Section 4.2.2. That information has been used to design a simple message routing algorithm to ensure the reachability of nodes in presence of an attack, using the information provided by the Opinion Dynamics algorithm and a secret sharing scheme. In this section, we extend the aforementioned proposal to take into consideration the anomalies caused by the compromise of both devices and communication links. The result is a new routing technique for the secure transmission of information in networks with low reliability channels.

As argued in Section 4.4, the original approach based on Opinion Dynamics for the APT detection required further improvement, especially to attain realism in the weight assignment procedure and hence circumvent the issues of the approach presented in Section 5.1. First, the aforementioned approach only focuses on the detection of topological changes over a graph-defined network, where a subset of nodes of  $V$  (i.e., the Dominating Set) are in charge of exchanging their opinions, which are represented with the ratio of change in their *betweenness centrality* indicator. Accordingly, the attacker model just contemplates the compromise of nodes to perform a removal of links. Even though this is valid to show the applicability of the algorithm using graph theory, we must go beyond and come up with different ways to model such opinion value in a real industrial ecosystem. For example, by considering QoS of communication links, as briefly introduced in Section 4.4. The reason is that APTs comprise a wide range of attack vectors besides the mere denial of service of nodes and communication channels, which pose a source of different anomalies (mostly subtle), that are potentially measured and correlated by the agent associated with the affected node. This lays the goal of this section.

### 5.2.1 Quality of Service Indicators for Routing Protocols

Critical infrastructures governed by industrial networks require to work at all times, even in the presence of intruders. To do this, we propose the use of a routing protocol as a response technique that uses the security information provided by a distributed detection system. However, in order to guarantee the delivery performance, this protocol must also make resource reservation and excise network control, in order to respond in a timely manner. Therefore, we plan to take the QoS of the communication links into account for the weight calculation in the Opinion Dynamics algorithm, so that the security of the communication links is considered for the opinion transmission, in the event that a fully distributed deployment of the traceability technique is used (making use of physical agents, as explained in Section 4.2.1).

In traditional data networks, routing protocols simply use shortest-path algorithms for the path computation, based on a single metric like hop-count or delay. In turn, QoS-aware routing protocols take into account further metrics to address the quality of service, in particular [281][282]:

- **Delay time.** It measures the time taken to transfer data across the network from one node to another. This value is often used to establish allowance limits for the communication links, so as to select the fastest route. In real-time operations, jitter or packet delay variations are used, measured with a sliding window of fractions of seconds. This is due to the dependence on the application (e.g., isolated environment of sensors, Internet-enabled connection to the SCADA system) or the network congestion, which could potentially slow down the communications.
- **Bandwidth.** It holds the maximum rate of data that can be transferred from a source to a destination per time unit. In order for the industrial devices to measure it, it is reasonable to determine the maximum bandwidth available at a given time. However, the computation of this value (along with delay) for routing purposes is a challenging problem since it can frequently change, as well as delay [283]. Also, in presence of an APT, there could not be any centralized control for allocating bandwidth among the nodes. For this reason, most existing QoS-aware routing protocols in the literature assume that the available bandwidth is known [284]. There are some others that estimate this value with carrier-sense capability of the underlying protocols (e.g., IEEE 802.11) to measure the idle and busy time ratio, and then adding this information to the route control packets.
- **Packet loss.** Packet loss can be used to measure availability, which represents the probability that some recipient is reachable with the claimed quality at a given moment of time. The packet loss is usually calculated as the ratio of lost packets or dropped connections in connection-oriented systems (e.g., upon retrieval of information from sensors).

Based on the set of adequate metrics, QoS-aware routing protocols perform resource estimation at each node and proceed with the route selection [285][286][287]. Routes are usually chosen to maximize the available bandwidth while minimizing the delay and the loss probability. However, finding a path that simultaneously satisfies more than one constraint is an NP-Problem. For this reason, heuristic approaches resulting in more efficient algorithms are often used in the literature. For example, [288] adopts three different criteria for the Optimized Link State Routing Protocol [289]. Another efficient scalable heuristic applied in [290] is based on Lagrangian relaxation. Another approach is based on the shortest-widest path algorithm [291], where a path with maximum bandwidth is found using a variant of the Dijkstra shortest-path algorithm and if there exists more than one such path then the one with the lowest delay is chosen.

Apart from these approaches, it is also possible to generate a single QoS metric from multiple parameters of the communication links. For the sake of simplicity and with the aim of aggregating different metrics (i.e., delay, bandwidth, packet loss ratio), our approach applies the following QoS function [292]:

$$\mathcal{S}(c) = \frac{B(c)}{D(c) \times L(c)} \quad (5.2)$$

where for a given communication link  $c$ , the metrics applied are the link's bandwidth  $B$ , delay  $D$  and packet loss  $L$ . Due to the reasons discussed before, the estimation of these metrics at each node is out of the scope of this article.

The output of  $\mathcal{S}(c)$ , when evaluated for a given communication link, is directly proportional to the quality of service that it experiences. This information can already be used for establishing a priority when selecting the routes along the network. However, besides the QoS measures applied to communication links, we will also introduce a security-based criterion for the selection of nodes that are traversed by our routing protocol. This additional information is provided by the Opinion Dynamics based detection system, explained in the following.

### 5.2.2 QoS-Aware Routing

In response to an APT, the combined opinions determined by the monitoring agents on the industrial network with regard to the security of its nodes and the QoS aspects of their communication links can subsequently be used to improve network routing. Here we present a novel approach aiming to enhance routing algorithms used in industrial networks such that the probability of packets being intercepted by potentially compromised network nodes is minimized while the QoS of paths through which these packets are routed is maximized. This way, we can ensure the confidentiality and reliability of the network until the threat is completely eradicated from the infrastructure.

Note that our approach extends the initial response mechanism proposed in Section 5.1. That proposal served as a first approach to enhance delivery of messages in presence of APT by relying on a redundant non-compromised part of the network topology and using secret sharing to split packets into chunks that are randomly dispatched over multiple paths. Still, it has a number of shortcomings, as discussed in the following. First, their attack model is based on a complete removal of communication links by compromised nodes, and does not consider a more realistic scenario where such links may experience varying QoS levels as a result of an attack. As observed in the recent years, many APT usually rely on zero-day vulnerabilities and make use of stealthy techniques to go unnoticed for a prolonged period of time, until they finally exfiltrate information or destroy the physical equipment. Therefore, it is necessary to consider a more subtle behavior of the attacker who may not wish to fully disrupt the communication and be detected. Second, the assumption on the existence of a redundant non-compromised topology in industrial control networks may not always be realistic. The architecture of such networks very frequently responds to a fixed configuration where all resources are rigidly connected with each other and so installation of a separate network topology might require significant investment and modifications of existing hardware devices. Third, the approach relies on the shortest-path estimation for which sending network nodes are assumed to know the entire network topology and has therefore limitations when used in combination with existing routing protocols that may not require nodes to have this knowledge.

The approach introduced now is more general and realistic. It aims at enhancing available routing algorithms to take into account the anomalies determined by the monitoring agents for the QoS levels of communication links and the security of network nodes when making routing decisions rather than selecting an optimal route based on the shortest path only. To set the background for our approach, we consider a typical architecture of an industrial network following the ISA-95 standard [36], as already mentioned in this work. In practice, due to the modernization of industrial technologies in recent years, these networks have evolved towards a more distributed model. Control devices (i.e., PLCs or RTUs) govern the production cycle by retrieving data from field devices (i.e., sensors and actuators), according to the information exchanged with SCADA systems. These are evolving towards cloud-based solutions, that interconnect other services within the organization. This way, we see how the network is divided into two main sections: the industrial assets (which we have referred to as ‘operational technologies’, OT) and the IT (Information Technologies).

Let  $G(E, V)$  be the graph that describes the overall network topology, following the preliminary concepts stated in Section 4.1.1. This graph is composed of the two subgraphs  $G(V_{IT}, E_{IT})$  and  $G(V_{OT}, E_{OT})$ , which are interconnected by a set of intermediate firewalls  $V_{FW}$  so that  $V = V_{IT} \cup V_{OT} \cup V_{FW}$ . More specifically, both are joined by the firewalls  $V_{FW}$ , that have connections with the nodes of  $V_{IT}$  and  $V_{OT}$  that belong to the PDS of those subnetworks. Again, with respect to the network topology, each of these subnetworks has a different configuration. On the one hand,  $G(V_{OT}, E_{OT})$  follows a power-law distribution of the type  $y \propto x^{-\alpha}$  [243], which models the hierarchical topology of an electric power grid and its high-level substations, which are subsequently connected to nodes with less connectivity (e.g., sensors and actuators). On the other hand,  $G(V_{IT}, E_{IT})$  presents a small-world distribution, that models the traditional topology of TCP/IP networks on the Internet [245]. Note that these concepts were already introduced in Section 4.1.2.

Over this distribution of nodes, there are two types of communication flows: information about the production chain delivered from the lower layers to the managerial IT network and, in reverse way, control commands issued from that section (e.g., the SCADA system) to the industrial process. For both types of the communication flows we base our approach on the Bellman-Ford algorithm [293] that is at the core of the Distance Vector Routing (DVR) [294] protocol, which determines the path to remote nodes using hop count as a metric. Each node holds a table that contains the distance to each node and the next hop in the route. This information is exchanged periodically with the neighbors, to ultimately compute the path using the Bellman-Ford algorithm. This contrasts to the Dijkstra’s path-finding algorithm [278] used in the previous defense technique, that finds the shortest path by requiring all nodes to have overall knowledge of the network topology and is at the core of the Link-State Routing (LSR) protocol [295]. In this protocol, routers periodically flood the entire network to ensure that each node holds a synchronized copy of the routing table. By choosing DVR over LSR we can compute paths

locally without involvement centralized routers as communicating with such nodes in presence of APT would impose additional risks.

The Bellman-Ford algorithm uses a weighted directed graph  $G(V, E)$ . The shortest distance from a node to the rest is determined by overestimating the true distance, following the principle of relaxation. In our case, since we want to prioritize QoS and security for the chosen path over the distance, we represent the weight assigned to each link  $e_{ij} \in E$  as

$$W(e_{ij}) = \frac{X_t(j)}{S(e_{ij})} \quad (5.3)$$

where  $X_t(j)$  is the final anomaly value of node  $j$  after executing the Opinion Dynamics as specified in Section 4.4. We select  $j$  instead of  $i$  since we want to prevent the messages against propagating to a node that is potentially compromised. On the other hand,  $S(e_{ij})$  refers to the QoS score of the communication channel  $e_{ij}$ , as specified in Section 5.2.1. The higher the anomaly sensed by the agent in node  $j$  is, the greater the weight assigned to that link will be. Correspondingly, the  $S$  score is inversely proportional to that value. By this means, we take into consideration the security of devices and the quality of service of their links when deploying our response technique in form of routing protocol.

Such DVR-based routing approach can be executed at any time of the production chain, paired with a previous execution of the Opinion Dynamics algorithm for adapting the network to the current security level, thereby achieving resilience. Therefore, we assume the process to update the routes can be executed as frequently as the security scenario imposes, which would not imply additional computing costs for the devices if we consider that the detection algorithm is executed in a central correlator system separated from the industrial network. In the following, we prove the effectiveness of our technique by simulating successive attacks against a network. In Chapter 6, the approach is validated with game theory to consider dynamic attack behaviors and additional defense solutions.

### 5.2.3 Simulation and Evaluation

In this section, our primary aim is to prove that the proposed QoS-aware routing approach based on Opinion Dynamics can effectively minimize the interception of messages, avoiding paths that contain compromised nodes while ensuring an acceptable level of quality. First, we define the attack model used in our simulation that determines how the anomalies are generated over the network and measured by the agents. Then, we execute the technique (i.e., the delivery of messages and the QoS analysis) with different parametrization of the topology and attacks performed. Finally, we evaluate the simulation findings.

**Attack Model: Simulation of Attacks and Anomalies**

In order to define a more realistic attack model for our response technique, we assume an attacker can break into the infrastructure by leveraging zero-day vulnerabilities and then use stealthy techniques to propagate over the network, until information is filtered or disruption to the infrastructure is caused.

Therefore, contrary to the approach based on the alteration of links, we consider an attack model based on a succession of lateral movements over the network nodes (inspired by the findings described in Section 2.4.2), aiming to infect as many devices as possible so that the security when delivering messages is jeopardized. Let *attackSet* be this sorted set of attack stages that an APT can perform against the industrial network, which is defined by  $G(V, E)$  and is composed by the IT and OT sections, as explained in before. This set comprises a finite number of elements of the following kind:

- **attackITnode**: the adversary initializes the APT or propagates the attack to a device in the IT subnetwork.
- **attackFWnode**: the attacker compromises a firewall (when the previously compromised node has connection with it), to propagate to the other section of the control network.
- **attackOTnode**: the intruder compromises a node in the industrial section of the network.

Every time the attacker takes over a new device, two main variables change:

1. From the **security** perspective, the agent associated with the compromised node notices an increase in the anomaly level, that ranges from zero to one, as described before. If we define  $x$  as the initial opinion vector for all agents, then  $x_i^t$  is updated in the simulation after attack number  $t$ . For simplicity, we assign a value that is randomly generated according to a uniform distribution over  $(0, 1)$ , simulating the existence of both subtle and evident anomalies.
2. From the perspective of **quality of service**, the agent also senses a potential alteration in the QoS experienced in the incoming or outgoing connections, as a consequence of the attack. The value of  $S(e_{ik})$  for all  $e_{ik} \in E$  in the simulations is originally chosen from a uniform distribution over  $(0, 1)$ , to represent the presence of channels with different QoS levels. In the event of an attack, the value of  $S(e_{ik})$  and  $S(e_{ki})$  scores decreases (being zero the minimum), where  $i$  is the attacked node and  $k$  refers to all neighbors of  $i$  such that there exists  $e_{ik} \in E$  (since each connection is bidirectional). This decrease is represented by  $\delta$ . Since the attacker can leverage stealthy techniques to go unnoticed without affecting the communications, this value is also chosen uniformly at random from  $(0, 1)$ .

Algorithm 9 describes the proposed APT life cycle. For all the attack stages in the provided *attackSet* parameter, the security of agents and the QoS score of the links is reevaluated, as

**Algorithm 9** Simple APT life cycle

---

**output:**  $X$  representing the final opinion value for all agents,  $S$  representing the QoS scores of links  
**local:** Graph  $G(V, E)$  representing the network, where  $V = V_{IT} \cup V_{OT} \cup V_{FW}$   
**input:**  $attackSet \leftarrow attackStage_{APT_x}$ , representing the APT chain of attacks

$x \leftarrow zeros(|V|)$  (initial opinion vector)  
 $\{attack \leftarrow first\ attack\ from\ attackSet\}$   
**while**  $attackSet \neq \emptyset$  **do**  
     $x(attackNode) \leftarrow U(0, 1), \delta \leftarrow U(0, 1)$   
    **for** neighbour **in** neighbours(attackNode) **do**  
         $S(attackNode, neighbour) \leftarrow S(attackNode, neighbour) - \delta$   
         $S(neighbour, attackNode) \leftarrow S(neighbour, attackNode) - \delta$   
    **end for**

$X \leftarrow COMPUTEOPINIONDYNAMICS(x, S)$   
 $attackSet \leftarrow attackSet \setminus attack$   
**end while**

---

described before. Firstly, the attacked node (specified with  $attackNode$ ) is assigned with a random value of anomaly (i.e., the opinion of its agent) in the uniform (0,1) distribution. Then, each of its ingoing and outgoing links are updated with a diminished QoS score, according to the value of  $\delta$ . Afterwards, Opinion Dynamics is executed to aggregate all opinions and calculate their final values, which eases the identification of zones under the effect of the APT. Finally, this information can be input to the routing protocol.

**Reliable Message Delivery**

Once the attack model has been defined, we can execute the defender's code based on the routing protocol in presence of an APT to firstly show that messages are successfully delivered in a way that the probability of traversing a compromised node (i.e., with an opinion value greater than zero) is lower than using the previously proposed approach. To simulate this, a set of 100 different messages are randomly generated, whose sender and recipient belong to the graph  $G(V, E)$ , making sure that more than one path exists between both nodes. Half of these messages are control commands (i.e., sent from the IT section to one device in the lowest levels of the infrastructure), while the other half are data packets, generated in the production chain and dispatched to the IT subnetwork. Therefore, messages are delivered in both ways based on the industrial topology defined in Section 5.2.2.

To assess the level of security experienced by the response technique and consequently compare it with other solutions, we define the *compromise level* indicator for each of the messages sent. This holds the sum of anomaly values (i.e., opinions calculated with the Opinion Dynamics algorithm, represented with  $X$  in Algorithm 9), which are measured by the set of nodes that

compose the path described by the message, in the route from the recipient to the destination. The greater this value is, the highest probability for the message to be intercepted will be. For a given number  $N$  of messages transmitted, we can determine the *average compromise level* as

$$\frac{\sum_{i=1}^N \sum_{j=1}^{|R|} X_j}{N} \quad (5.4)$$

where  $X_j$  is the opinion of agent  $j$ ,  $1 \leq j \leq |R|$ , and  $R$  is the set of nodes that each message  $i$  traverses. This overall value is calculated for our custom routing protocol and will be compared with two other approaches: on the one hand, **(a) the previously proposed mechanism in Section 5.1**, that is based on the Dijkstra's shortest-path algorithm, without considering the opinions of the agents; on the other hand, **(b) the Dijkstra's path-finding algorithm parametrized with the opinion of agents** as weights for the search of the optimal path (i.e., the route with a minimal compromise level). In other words, for the computation of the path from sender to recipient in  $G(V, E)$ , (a) uses a weighting function  $W$  for each edge  $e_{ij} \in$  such that  $W(e_{ij}) = 1$  if  $e_{ij}$  simply exists (so that the destination is reached in the minimum number of hops). As for (b),  $W(e_{ij}) = X_j$ , hence prioritizing not to hop to a compromised node. Our aim is to show how **(c) our approach based on Bellman-Ford algorithm**, that uses the weighting function defined in Equation 5.3, achieves better security (i.e., the value of compromise level) than (a), with closer results to (b).

In this experiment (carried out in Matlab), we have generated a random industrial network of 50, 100 and 200 nodes following the topology described in Section 5.2.2 (where the two halves of nodes are respectively used for the IT and OT subnetworks and an extra firewall node is used to merge them). Over these topologies, we have simulated the effect of an APT (according to Algorithm 9) composed by 50, 100 and 200 attack actions, respectively. We have represented the overall behavior of Stuxnet (which was deeply addressed in Chapter 2) at a basic level: the APT begins by compromising one node from the IT network (originally using malicious USB flash drives) and then spreads through the entire subnetwork until it finally breaks into the OT section, where the threat propagates until it infects the target device (the uranium enriching centrifuges).

By making sure the number of attacks reaches the number of nodes, we represent the most critical scenario when the APT takes over the entire network, thereby showing the effectiveness of the algorithm at all times (although this validation process could be further optimized if attacker and defender were part of a dynamically confronted competition with specific action rules, by means of game theory). After each attack phase, the Opinion Dynamics algorithm is executed and the set of 100 random messages are delivered, putting into play the three aforementioned routing algorithms. Finally, the average of compromise is calculated. The plot in Figure 5.6 shows the evolution of this value over the entire set of attacks for the three assessed solutions.

As we can see, the Dijkstra's algorithm that uses the opinion of agents as weights to compute the path serves as the baseline of the minimum compromise level that can be achieved. However, our approach based on Bellman-Ford algorithm presents a similar result with a slight increment

of anomaly experienced, that still remains far from the high value experienced by the Dijkstra's scheme proposed before, as we wanted to demonstrate. We will now prove that our approach also provides better QoS requirements.

### Quality of Service Experienced

After analyzing how our solution effectively experiences a lower level of compromise when routing the messages, it is also necessary to prove that the paths generated by the protocol also achieve an adequate quality of service, which is the main aim of this section. This would ensure a fast and reliable communication, especially necessary when the computed paths impose several hops to reach the recipient as a consequence of avoiding the effect of the attack.

Following the previous methodology, we aim to deliver a set of 100 messages over the network  $G(V, E)$  in such a way that the number of hops is minimized and the QoS experienced is maximized. This time, we define the *QoS level* indicator for each message sent as the sum of individual QoS scores for all the successive edges that belong to the path (as explained in Section 5.2.1) divided by the number of hops that this message performs. The greater this value is, the better QoS with a lower number of nodes traversed will be. Given  $N$  messages transmitted, we can determine the *average QoS level* as

$$\frac{\sum_{i=1}^N \frac{\sum_{j=1}^{|R|} S(e^j)}{hops_i}}{N} \quad (5.5)$$

where  $S$  is the QoS score function from Equation 5.2,  $e^j$  refers to edges from the route  $R$  which is taken by message  $i$ , and  $hops_i$  is the number of intermediate hops. This average QoS value is calculated for our routing approach in presence of APT using the same topology and attack scenarios as in the previous test case, and is compared with the two other approaches: **(a) the previously proposed mechanism in Section 5.1**, that is based on the Dijkstra's shortest-path algorithm without accounting for any QoS implications; and **(b) the Dijkstra's path-finding algorithm parametrized with the QoS score of the edges** as weights for the search of the optimal path (i.e., the route with a maximum quality). Thus, (a) uses an  $W$  weighting function for each edge  $e_{ij}$  such that  $W(e_{ij}) = 1$  if  $e_{ij}$  simply exists, while in (b) it uses  $W(e_{ij}) = 1/S(e_{ij})$ , hence prioritizing the path with maximum QoS. In this case, our aim is now to show how **(c) our approach based on Bellman-Ford algorithm**, that uses the weighting function defined in Equation 5.3, achieves a better QoS level than (a), with closer results to (b).

The plot in Figure 5.7 represents the evolution in the average QoS levels. As the previous test case, the QoS-aware Distance Vector Routing presents a QoS level per hop ratio similar to the Dijkstra's algorithm weighted with the QoS scores. As we can see, the three routing approaches have their QoS levels diminished as the APT evolves (due to the attacks and consequent decrease of the  $S$  scores, as explained in Algorithm 9), but our approach shows a higher QoS level, close to the one experienced by the optimal Dijkstra's solution. Therefore, we have demonstrated our



reliable routing approach behaves in a nearly optimal way, more efficiently than the original response technique. Figures 5.6 and 5.7 prove that QoS- or security anomaly-based routing alone are not sufficient, since both criteria must be complied to ensure a delivery of messages balanced with a decent level of security and QoS. In addition, table-driven routing algorithms like DVR with the Bellman-Ford algorithm also ensure an ad-hoc selection of routes without any central entities involved in the communications, which can help achieving a higher level of security while alleviating the large amount of traffic that route updates like the original protocol can imply.

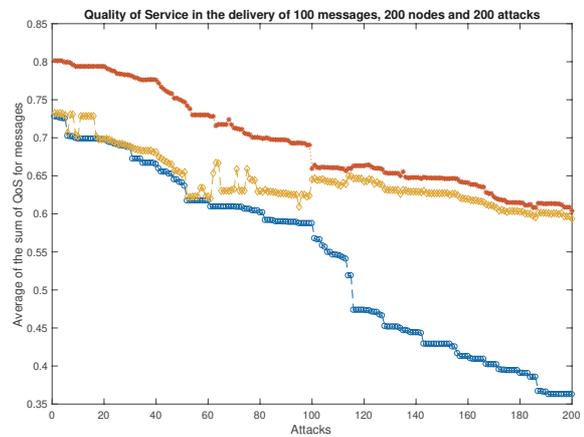
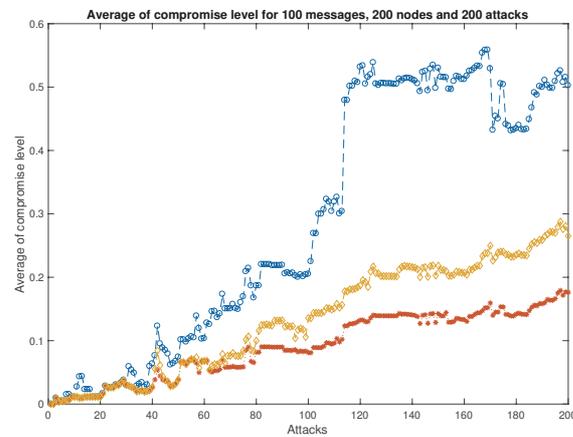
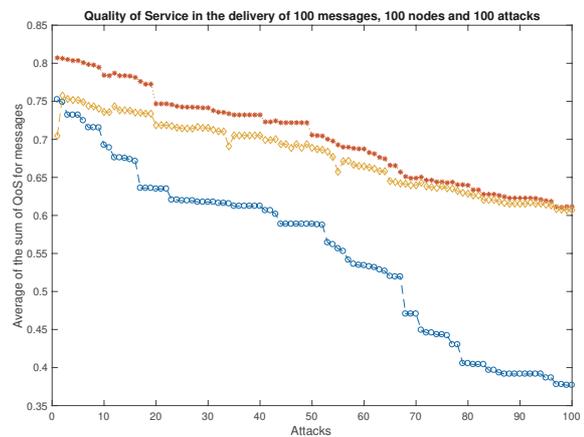
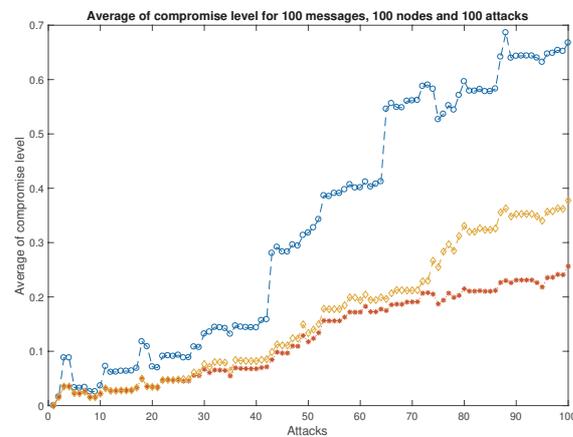
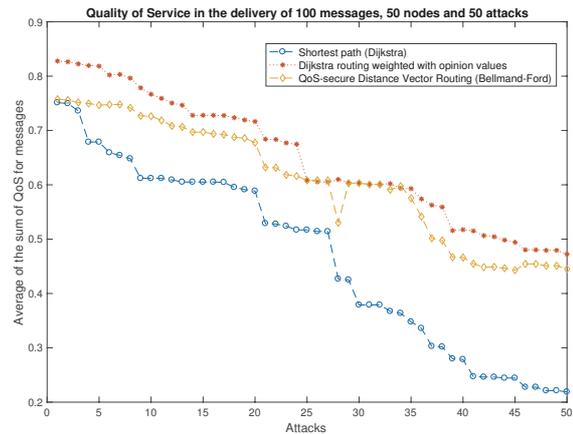
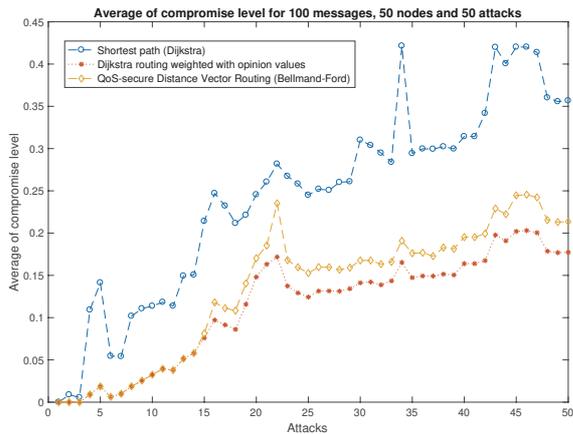


Figure 5.6: Average compromise level

Figure 5.7: Average QoS level

### 5.3 Applicability of Opinion Dynamics in the Industrial Internet of Things

As described in the introduction of this work, the interconnected things in the IIoT mainly consist in specific devices such as sensors, actuators or controllers that altogether enhance the

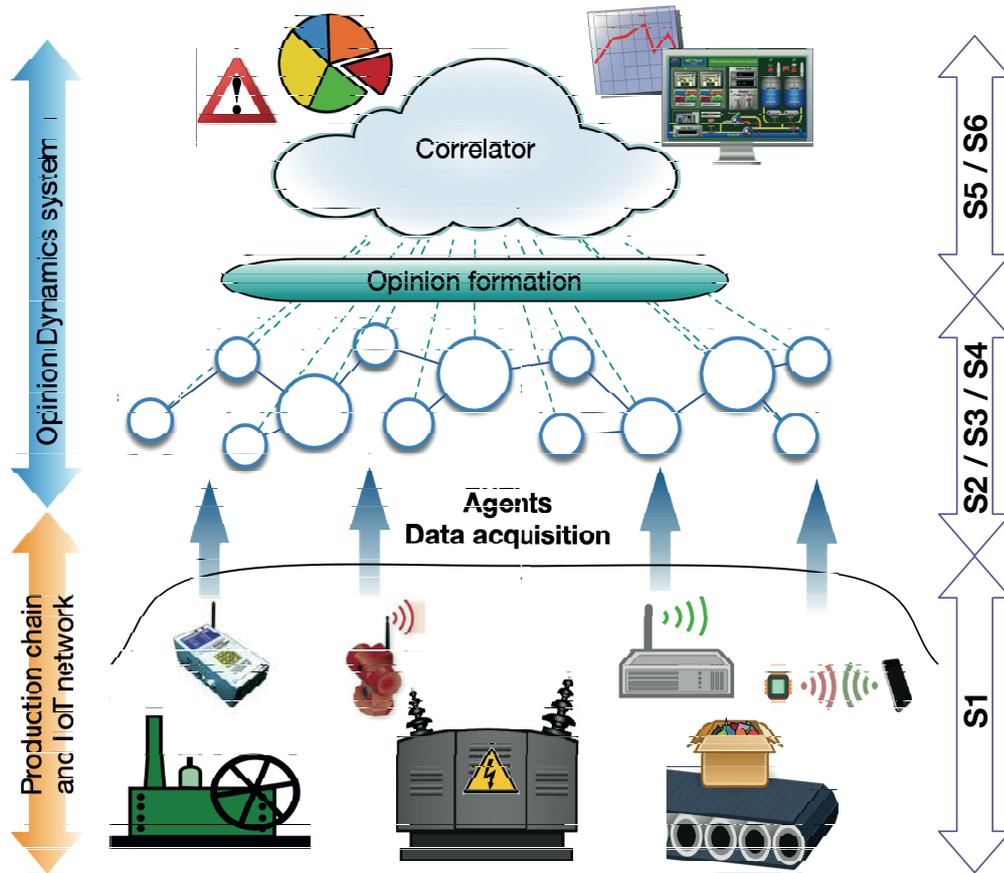


Figure 5.8: Stages of the Opinion Dynamics framework in a IIoT network

productivity in multiple sectors. The other side of this increased connectivity comes with the exposure of systems that were traditionally isolated to cyber attackers. This is reflected in the increasing number of attacks that these environments can suffer, such as unauthorized accesses (e.g., spoofing or phishing), data manipulation (e.g., man-in-the-middle attacks, packet injection), DoS attacks and other kinds of malware. These were thoroughly addressed in Section 2.2.1.

As a consequence of the convergence between the IoT and the industrial world, these attacks are similar to the ones in common IT systems, but their consequences in the IIoT are much more critical. Here, they can endanger the safety of resources and generate serious operation disruptions. Additionally, countermeasures that are usually deployed in IT environments (e.g., network segmentation, embedded security, encrypted communications, access control policies) require a greater effort in an industrial scenario, in which we have a non-interruptive operation restriction. Even so, these protection solutions could fail to prevent certain threats, such as insider attacks or DDoS attacks. In this sense, intrusion detection systems are a defense to prevent against unauthorized accesses, by finding patterns in the network data that does not match the expected behavior.

Given these problems, this section addresses the applicability of the APT traceability framework in IIoT ecosystems, specifically using the Opinion Dynamics correlation. The procedure carried out by the elements of this framework is presented in Figure 5.8, and its comprised by six stages. In stage 1, *data retrieval setup*, the system extracts the outputs of multiple anomaly detection mechanisms, vulnerability scanners or SIEM systems. In stage 2, *agents creation*, all data associated to a particular entity or device is assigned to its corresponding virtual agent. Note that raw data not extracted from existing IDSs, such as network traffic, can be used to obtain additional features (e.g., traffic volume, type of connections established) in stage 3, *Feature extraction*.

In stage 4, *feature selection and opinion formation*, each agent  $i$  combines all available data into an opinion  $x_i(t)$ , which shows the opinion (i.e., anomaly value) of the agent at a given time  $t$  (i.e., the security state of its monitored node, measured from 0 to 1). For this task, different models can be applied to weigh each feature depending on the current security scenario and the anomalies sensed. The evolution of these opinions over time is considered in stage 5, *correlation of opinions*. In this phase, all opinions evolve by taking into consideration the opinions of the surrounding agents  $x_j(t)$  and a weight  $w_{ij}$ . In order to facilitate this process, in the current incarnation of this framework, such correlation is executed in a central system (i.e., the centralized implementation of agents described in Section 4.2.1). As explained in Section 4.3.2, all opinions evolve using the following expression:

$$x_i(t + 1) = w_{i1}x_1(t) + w_{i2}x_2(t) + \dots + w_{in}x_n(t)$$

As a result of this correlation, it is possible to extract additional indicators in stage 6, *computation of indicators*. For example, all opinions can be grouped into clusters at any given time, providing a representation of the segments of the network that are being affected by existing attacks. Moreover, a global health indicator can also be calculated from the aggregation of all opinions. This opinion model can be enhanced by taking into account other parameters such as the criticality of the monitored resource, its historical events, or the persistence of the detected attacks, as discussed in Section 4.6.

Nevertheless, further research is necessary to fully realize the APT traceability framework in the IIoT domain, since there are several constraints and open questions to be solved. For example, whether using a centralized entity as implementation model is a feasible solution in all scenarios, or how to precisely instantiate these agents (e.g., IDSs, anomaly detection mechanisms) on a physical infrastructure whose criticality may restrict the modifications of hardware and software. Additionally, the potential overhead introduced in the communications, or the provisioning of parallel network interfaces to gather and analyze network traffic are other open issues that we aim to resolve in this section. More specifically, we will study the precise instantiation of the

algorithm, making more emphasis on the earliest stages – as they revolve around the integration of the algorithm with the IIoT network at low level.

### 5.3.1 Data retrieval

To start devising the integration of the framework over an IIoT scenario, the main question that arises is the nature of the information that can be collected by the detection system. As stated previously, we must provide the agents (regardless of where they are executed) with data of interest about the state of the resource they are monitoring, as to finally output a single – but aggregated – value of anomaly, that represents its opinion (stage 4). This process requires of data that is retrieved in stage 1, either from raw information extracted from the low layer and high layer protocols or from outputs of IDS solutions such as the ones described in Section 3.3. Here, we will especially focus on the former, as the existence of IIoT IDSs already proves their feasibility as inputs to the framework. In general, the information that can be processed by agents include, but are not limited to:

- **Network parameters:** involves two kinds of information, related to the topology and the state of the network, to infer the presence of anomalies via traffic analysis (by comparing the current value with the one learned in normal conditions):
  1. A physical network mapping that contains every pair of devices connected through a communication channel (in form of a graph, with the address of every node within the topology). This can be easily determined from the number of packets per protocol and recipient, which helps to tag frequent and non-frequent communications.
  2. QoS indicators: they inform about the reliability of connections by means of metrics like the delay time from one node to another, the bandwidth experienced and the packet loss ratio in connection-oriented protocols.
- **Communication information:** it implies the analysis of the payload contained within the exchanged packets and their frequency, which includes low-level commands issued from one source to its destination (e.g., control commands to actuators), as well as quantitative values from operations (e.g., readings from sensors). The former allows to detect suspicious actions potentially performed by compromised devices, while the latter permits to create a statistic model to later identify deviations in the values exchanged.

Going back to the early stages of the algorithm, the method for extracting these features from the traffic in a IIoT network is highly dependent on the wireless transmission channel used, its particular deployment architecture, and the application endpoint where data is consumed (which is presumed to be the central correlator). The aim with stage 1 is to seamlessly gather the aforementioned network information without interfering with the operations of the production chain (i.e., additional computation and delays) and, whenever possible, without introducing extra

physical equipment. This imposes several challenges, such as inferring a low-level network mapping out of the application data received by upper layer protocols (e.g., when only a gateway is visible for the industrial segment as an interface to the IIoT subnetwork) or estimate indicators through a parallel communication channel when the primary one is inaccessible (e.g., in third-party cellular networks).

Consequently, we must start by studying the amount and quality of data that can be potentially collected from the IIoT network given a specific configuration. For the sake of clarity, we define the concept of OT cell as a subsection of the entire industrial infrastructure where the same underlying wireless technology is implemented. Thus, according to the classification of lower layer protocols described in Section 1.1.3, we can draw some conclusions about the network parameters that can be obtained:

- **WPAN networks.** Both classic Bluetooth and the low-energy specification (the latter featuring the creation of a large-scale mesh of devices) support connectivity at IP-level in certain nodes within a network, acting as bridges between the industrial domain and the sensors at field level. As for IEEE 802.15.4 devices, gateways (e.g., coordinators in a Zigbee network) often centralize the retrieval of data from the lower layers of the industrial architecture. Therefore, the network-related information that is possible to extract in a OT cell of this kind is the one retrieved by the gateway that interconnects it with the upper levels of the infrastructure. This usually implies that the original information exchanged by sensors/actuators using these lower layer protocols is translated by the gateway into common industrial standards such as ModbusTCP, thereby losing granularity when studying the precise topology and QoS indicators. Consequently, we have three alternatives: (1) to deploy a capillary network that captures and relays the missing information through an auxiliary network interface (introducing hardware in exchange); (2) to manually provide the network mapping information at low level and establish the relationship with high level packets (lacking the QoS information); (3) to rely on this aggregated data and carry out a deep analysis of high level packets to infer the network mapping.
- **Wireless Area Networks (WLAN).** IEEE 802.11 standards, and in particular the latest 802.11ah standard, facilitate the creation of IIoT networks where a large number of devices need to cover wider areas. In contrast with WPAN networks, this is achieved with a higher power consumption, which enables the use of the IP protocol in all devices to cover areas of up to 1000m in a single hop. In addition, Relay Access Points are used to extend the connectivity to Access Points (APs), that transparently deliver the field level information to the industrial network, without any routing between the endpoint and the gateway. From the data acquisition perspective, this means that the network mapping and QoS indicators are easily obtained by capturing and analyzing the exchanged traffic packets.

- **Cellular networks.** When collecting low-level information in Cellular Networks, the amount of packets that can be captured decreases dramatically due to the presence of a public telecommunication network that processes all the traffic before it is consumed in the industrial network. Thus, it is not possible to obtain QoS data while packets are relayed through the multiple hops of the external infrastructure. Plus, the network mapping must be inferred at logical level, by capturing application level traffic and accounting for every source-destination pair within the industrial premises. This scarce amount of information increases when an edge paradigm is leveraged (e.g., fog computing or mobile edge computing) or when some of the cellular network infrastructure assets are controlled privately by the company, instead of an external provider.

Wireless transmission channel	Network parameters accesible	
	Network mapping	QoS
WPAN (IEEE 802.15.4, Bluetooth)	Through an additional capillary network, analysing high-level data from the gateway or manually	From the IT/OT network to the gateway only
WLAN (IEEE 802.11)	Yes, all data	Yes, all indicators
Cellular Networks and LPWAN	Logical network mapping, unless external telecomm. infrastructure or edge network resources are monitored	end-to-end indicators, unless external telecomm. infrastructure or edge network resources are monitored

Table 5.2: Network parameters collected from the different IIoT cells

Table 5.2 summarizes the different methods for collecting low-level network information in each IIoT cell. Still, stage 1 does not depend only on the information provided by lower layer protocols – it also revolves around gathering information about the communications at application level, as explained before. This can be classified into two classes: information about the production chain from the field devices, and control commands issued from the IT section to the industrial process. As for the former, the process of extracting the measured data from sensors is relatively straightforward, depending on the upper layer protocol used to exchange data:

- In asynchronous message protocols and publish-subscribe mechanisms such as MQTT or AMQP, the entity in charge of running the detection algorithm should be registered as subscriber to receive the measurements from the broker (i.e., the intermediate gateway).

- In RESTful architectures like CoAP or HTTP, the sensors readings would be accessed by means of an API (published by a CoAP server executed on an intermediate gateway or embedded in the own device on the field).
- In frameworks such as OPC UA and OneM2M, the retrieval of data requires additional analysis of how it is generated and consumed by endpoints, since they respond to abstract specifications of communication interfaces between services and components that are integrated in specific domains. It usually implies reading values from a common server that exposes a friendly API under a unified data model.

It is worth noting these communication channels very frequently use encryption measures to ensure the confidentiality of data (e.g., CoAP is built on top of DTLS). This makes it necessary that the entity that retrieves data from devices and executes the detection algorithm is allowed to access the exchanged data and comply with the system access control policy.

On the other hand, we also should be able to retrieve the precise set of commands that are issued from the managerial level of the industrial network, as explained before. According to the architecture of a IIoT-based control system, this implies filtering the operations executed by a PLC, which is hierarchically placed on top of an IIoT cell and ultimately issues commands to sensors/actuators (potentially using intermediate IIoT gateways). These devices can operate with a large range of protocols, ranging from open source standards like ModbusTCP or Ethernet/IP to private alternatives such as S7 from Siemens. In this case, accessing to the commands executed requires the development of dissectors for the particular protocol, which exceeds the scope of this thesis. However, as there are numerous solutions available in the market that especially focus on the analysis of these standards [218], it is possible to use external IDS results as inputs for our system.

### 5.3.2 Correlation of Anomalies

In this section, we introduce the design of the rest of the stages of the Opinion Dynamics algorithm to satisfy the APT traceability framework in IIoT networks. Note that we do not analyze stages 5 and 6, as these stages are independent from the underlying infrastructure once all necessary information (e.g., opinions) is available. For this particular instantiation, we will make use of the information extracted in Section 5.3.1, without resorting to external systems (i.e., existing IDS systems). Note that, due to the nature of the framework, those IDSs can be integrated anytime.

The virtual agents created in stage 2 deal with the processing of data retrieved in stage 1 and the features extracted from stage 3. From a physical point of view, this firstly means that the central correlator that executes the Opinion Dynamics System must establish a communication channel with every IIoT cell that is being monitored, so as to gather the network parameters (e.g., a link to the gateway in a WPAN or to the AP in a WLAN network). Likewise, it must be able to access the interfaces where data is published (e.g., the API in a CoAP based network).

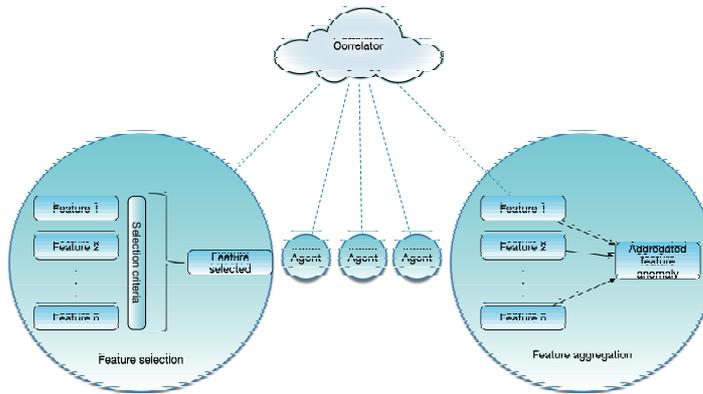
Then, from a logical perspective, this information in bulk is divided and assigned to virtual agents created by the correlator.

As deeply explained in Section 4.2.1, these agents are threads in charge of individually monitoring the security of an IIoT device within the topology to subsequently derive an opinion, following a 1:1 relationship between devices and agents. Equivalently, an agent receives the traffic (containing data and commands) that is exchanged by its assigned device, as well as the QoS indicators of every connection that it shares with the rest of neighbours. At this point, the physical network mapping conducted in stage 1 is essential for the central correlator to make such assignment of information. Nevertheless, as discussed before, the knowledge about the physical topology is not always accurate, due to the presence of intermediate gateways that aggregate data from a mesh of constrained devices and hinder the retrieval of network parameters. In this case, when the actual mapping cannot be determined by any of the methods presented in Table 5.2, we can assume the existence of agents that encompass a set of multiple devices.

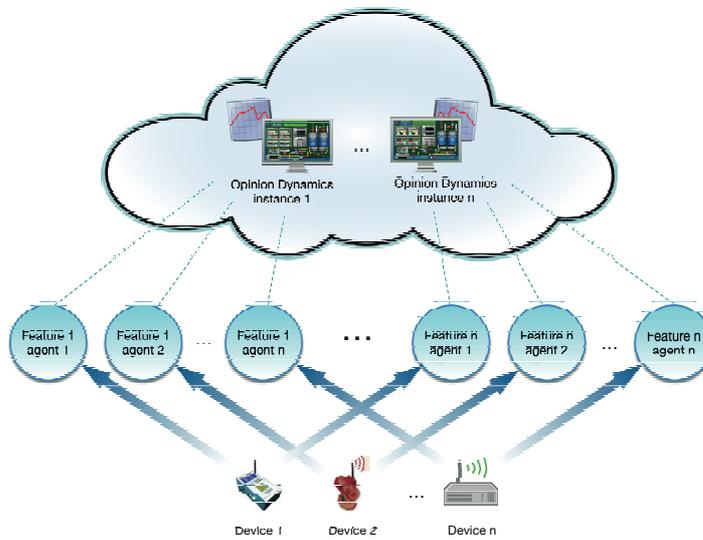
Once agents are created and provided with the information that they need to process, they perform an extraction and selection of features from that data in stage 3. These features refer to variations in certain magnitudes or indicators, which evidence anomalies suffered as a consequence of an attack. Some examples of features applicable to IIoT networks are:

- Number of connections established and devices accessed.
- Traffic load (total number of packets exchanged).
- Type of communication protocols used.
- Delay experienced in every communication channel.
- Ratio of lost/corrupted packets.
- Frequency and type of commands issued.
- Precise data values transmitted by sensors.

These features are monitored periodically (as often as the Opinion Dynamics is executed to visualize the latest changes in the network). A model is created to represent the behavior of each one so that it is updated in every period. Even though diverse alternatives could be proposed for formalizing this model, here we conceptually propose a simple but accurate approach, which is internally used in commercial IDSs: in the case of quantitative values (e.g., number of packets), the average is calculated. As for discrete features (e.g., devices accessed or protocols used), the model is represented with the set of occurrences for each value (e.g., number of packets sent to a given recipient or using a certain protocol) and their corresponding average. Either way, the values obtained for each feature are compared in each period with the existing model, which is assumed to reflect the behavior of the system in normal conditions (therefore, a initial phase of training



(a) Feature selection and aggregation



(b) Multiple Opinion Dynamics instances for features

Figure 5.9: Alternatives for the opinion formation in stage 4

is assumed). As a result, the standard deviation provides a value of anomaly for quantitative features. In discrete ones, the value of anomaly can be determined by analyzing the individual deviation in the number of occurrences. This way, the extraction of features would be complete for each agent.

All of these features are closely related to intrinsic network aspects of the devices monitored. A future work could involve the analysis of host-based parameters in the own IIoT devices as a source of anomaly for the opinion computation. For instance, the usage of CPU and memory, the processes running, and others. This would require the integration of capillary networks that retrieve such information from the OT cells or using external detection systems. This is possible due to the adaptable nature of the framework, which is open to include all kinds of features.

The opinion formation in stage 4 is the last stage before the correlation of anomalies and analysis of detection results. The opinion of each agent is formed at this point by deriving a single value from the set of anomalies sensed in each feature, which implies making a selection or aggregation. Diverse policies could be applied and compared, being the easiest to *select the feature* whose anomaly value is the maximum as the opinion for a particular agent. This would make the overall results of the Opinion Dynamics system very sensitive to changes, since a singular feature from the complete set of indicators measured by an agent could influence a whole neighbourhood of agents and raise risk alarms indicating the presence of a threat. Still, this approach could be recommendable in highly critical infrastructures where a fine grained auditing is needed. An alternative to selection is the *aggregation of features*, using the average of anomalies sensed for all the indicators considered (as long as they are not zero), for instance. However, the drawback of this approach is that greater anomalies measured in important features would be occulted to the correlator due to the aggregation with lower anomalies in other features. In this case, a weighted average of features would be interesting.

Lastly, there is one more way to implement this stage and avoid the loss of detail as a consequence of a selection or aggregation. It consists in *conducting a Opinion Dynamics correlation per feature considered*, in such a way that multiple instances of the detection algorithm are executed in the centralized entity, where each one concerns on a specific indicator. In that case, the correlator would take the anomalies in each feature as individual opinions for all the Opinion Dynamics instances (equivalently, each device would have an agent per feature monitored). As a result, it would be possible for a security administrator to visualize the state of connections, delays, protocols, etc. with a deeper level of detail. All these three alternatives are summarized in Figure 5.9 and shown in the next section through a simple case study.

After the formation of opinions in all agents of the network, they can be correlated and analyzed in stages 5 and 6 using the Opinion Dynamics algorithm to visualize the clusters of agents that expose the same degree of anomaly measured in their surroundings. This information is useful for computing health indicators for diverse areas and carry out a precise analysis of the historical data to draw conclusions about the attack pattern and predict future actions, as explained in Section 4.6.

#### 5.3.3 Case Study

After the study of the applicability of the Opinion Dynamics in the IIoT, this section focuses on showing the benefits of a conceptual deployment of this approach by means of a theoretical study case. Additionally, the three aforementioned alternatives for conducting stage 4 are discussed. In order to achieve these goals, we will follow the same methodology of Section 4.1.3 using graph theory.

The formalization of the network is explained in the following. Firstly, we define a graph that represents the physical interconnection of the Opinion Dynamics system with the multiple

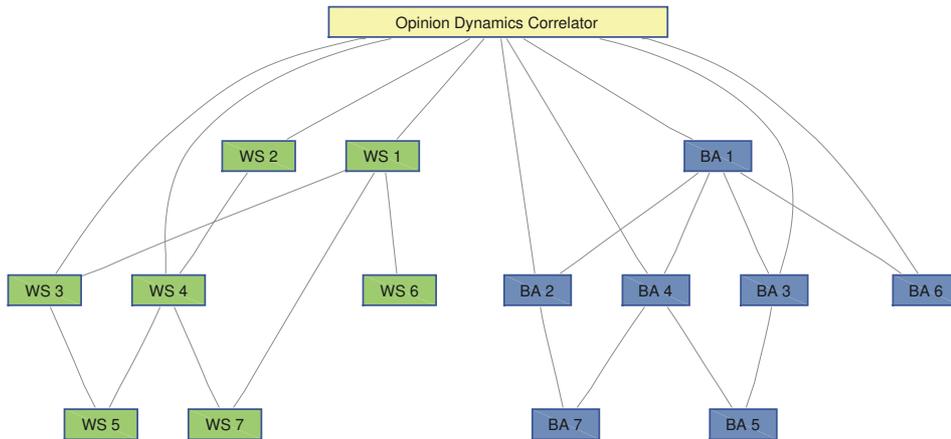


Figure 5.10: Example of network composed by two IIoT cells, using the Watts–Strogatz (WS) and Barabási–Albert (BA) model

IIoT cells that are present in the infrastructure, from which data is retrieved. For this purpose, we leverage the Watts-Strogatz [296] and the Barabási-Albert model [297]. Both distributions permit to simulate the topology of an IIoT cell, being the former used for producing graphs with small-world properties [298] and the latter for generating random scale-free networks [299], such as the connection of devices on the Internet. Here, we generate two simple cells of seven devices, which are accessed by a central correlator through the nodes which hierarchically have more connectivity (the Power Dominating Set [241], as in [300]), in order to simulate the presence of gateways, as explained in previous sections. The resulting network is depicted in Figure 5.10, that illustrates the implementation of the Opinion Dynamics correlator and its connection to the rest of nodes, which are labelled in each IIoT cell according to the model used.

Therefore, in this case study we assume the existence of a central correlator that is able to gather the network parameters and the communication information from all devices across each IIoT cell, using the strategies described in Section 4.2.1. Afterwards, the virtual agents located in this central correlator will be able to extract features and subsequently form their opinions. In order to show the impact of an attack over their computation, we use the same methodology as in Section 4.1.3, to formalize the attacker model of APTs in a sequence of steps. Each step has its own detection probability, which is quantitatively reflected in each agent to simulate a certain degree of anomaly measured. In our case, we provide these agents with a minimum set of anomaly detection rules based on two features: (1) the delay in their communication channels, and (2) the data values transmitted through those links.

In this case study, we will perpetrate a simple two-step APT attack against the IIoT cell based on the Watts-Strogatz model. These two steps are as follows: an initial intrusion against node 2, and a lateral movement towards node 4. In this basic example, if we consider that this

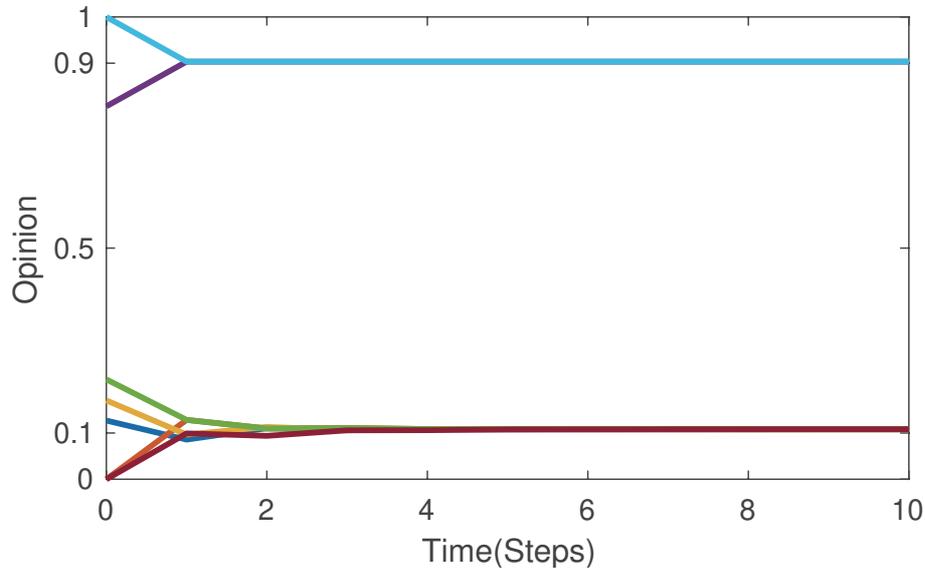


Figure 5.11: Opinion Dynamics clusters after a lateral movement in the IIoT cell

propagation makes use of a covert channel attack (which usually leverages delays introduced arbitrarily in the packet transmissions [301]), then each affected agent should raise a level of anomaly with respect to that feature. This would serve as input to ultimately execute the Opinion Dynamics algorithm and narrow down the attack.

Figure 5.11 plots the result of the Opinion Dynamics correlation between the seven agents that belong to the Watt-Strogatz cell. The lines represent every agent opinion, that ultimately form two consensus after executing the algorithm with 10 iterations, as explained in Section 4.3.2. This means that the network is divided into two clusters of nodes that suffer two grades of anomalies: one group of five agents (that sense a 10% of anomaly) and another one of two agents with 90% that correspond to the nodes involved in the lateral movement. Here, stage 4 has been carried out by *selecting the feature* whose anomaly value is higher, that in the case of node 2 and 3 is the delay. As for the rest of agents, the level of anomaly around 10% appears as consequence of a negligible variation on their data values transmitted. In case that *feature aggregation* was used instead of selection, an average of the anomalies in both features would be shown on the figure, which only serves as indicator that a greater-than-zero anomaly is occurring. Otherwise, if an *individual Opinion Dynamics instance* were used for each feature, the bottom of the plot in Figure 5.11 would not appear in the delay one (since those nodes do not show any variation of delay), whereas the top of the plot would not appear in the instance that concerns on the data variation (and opinions of nodes 2 and 3 would also be merged into the bottom cluster due to a low level of variation).

The next step of the framework execution would be to keep track of the multiple APT anomalies over time, associate them with actual attack phases and create a map with the

complete threat evolution throughout the network. This will be illustrated with a real setup in Chapter 6. Altogether, this brief description of threat detection exhibits how a security administrator could benefit from different correlation configurations to trace down the implicated nodes of an attack and accurately filter the anomalies suffered across the topology at all levels. This helps to identify the origin of the infection while anticipating further actions to introduce effective response procedures.

## 5.4 Applicability of Opinion Dynamics in the Smart Grid

### 5.4.1 Resilient Architecture for Fault Detection

The traditional architecture of the electricity grid has evolved in great measure since its original conception where the production and distribution of energy were supervised by a centralized system. With the introduction of Internet communication technologies in this scheme, there has been a shift towards a more interactive, interconnected and dynamic grid model of the 21st century, known as the Smart Grid. Its main benefit is the two-way flow of information, through which the user (i.e., by means of a smart meter installed in the household) and the utility company can communicate, making it possible to perform a fine-grain consumption metering, whose information is accessible to both of them [302]. This allows the user to participate in programs that aim to reduce electricity use when energy prices rise, and also allows him/her to sell the electricity generated at home (e.g., using solar panels). The utility company can also take advantage of this technology to improve demand response, by managing the generation and delivery of electricity in real time, so that grid operators can rapidly anticipate high peaks of demand and avoid power outages.

This metering model is put into practice through the Advanced Metering Infrastructure (AMI). This comprises all the elements that collect and transfer the consumption data measured in the home domain through many aggregation points until it reaches the utility provider end, where the information is analyzed for billing and control purposes, by means of the so-called Meter Data Management Systems (MDMS).

This data acquisition process requires both industrial and information technology equipment. On the one hand, the industrial network is conformed by the SCADA systems that are leveraged to remotely access the devices that sense the energy flow of many consumers in real time. These include, for example, the RTUs and PLCs, that are present in the substations spread over the WAN (Wide Area Network) or the Smart Grid. On the other hand, support for the MDMS procedures by interconnecting these industrial assets with external networks (e.g., Internet) and innovative technologies (e.g., cloud computing) to undergo further data analysis and support demand response.

This growing interconnection of SCADA systems (which traditionally work in isolation) has increased the number of cybersecurity threats in this context [303], favoring the appearance of

sophisticated attacks which aim to stealthily compromise nodes within the control network over a long period of time, as is the case with APTs. The presence of these attacks can damage the infrastructure and jeopardize the availability of resources, which translates into the inability to hold the power supply and potential blackouts in the grid [304]. By the same token, safety measures must also be introduced to preserve the availability of the power supply against high peaks of demand (that may also be provoked on purpose), hence avoiding outages.

For the aforementioned reasons, we firstly present the design and implementation of (1) a defense mechanism based on the APT traceability framework to detect changes in the industrial network arising as a consequence of these attacks. In addition to ensuring security, here we also address the safety of the Smart Grid resources by implementing (2) a load balancing model that permits a successful energy supply for the entire grid taking into consideration the prediction of future consumption. Both safety and security measures are included in a novel architecture to be easily integrated in the current Smart Grid conceptual model [305].

### **Architecture and Initial Assumptions**

The architecture of our approach is presented in this section and has two main purposes: (i) to predict high electricity peaks in comparison with the recent demand to uniformly distribute the energy supply to the consumption areas, and (ii) protect the control from external attacks (e.g., APTs). To achieve both goals, and therefore, the contributions of the work presented here, an architecture of two main networks is modelled: an energy network ( $N^e$ ) and a communication network ( $N^c$ ). These two networks contain five independent but strongly interconnected subnetworks, which are shown in Figure 5.12. Each subnetwork contains a set of Internet-enabled nodes (e.g., meter concentrators, gateways, RTUs, etc.) capable of interconnecting by themselves with other subnetworks. As for the energy network, the following subnetworks have been defined:

$N_1^e$  illustrates the customer's premises, subdivided into several power distribution areas or communities. In this case, each area characterizes a sub-part of a population, demanding energy according to its needs, requirements and life quality.

$N_2^e$  represents the spinal column of the entire energy generation and distribution infrastructure, which remains in a fixed and static deployment and configuration state.

In practice, electricity generators in  $N_2^e$  are interconnected in the power grid with the consumption areas in  $N_1^e$  through rigid transmission and distribution lines. Besides these energy subnetworks, we also deal with communication subnetworks that firstly transfer the energy usage data from the consumers to the provider and secondly transmit the control commands from the utility to adjust the generators according to the demand. In this sense, we define:

$N_1^c$  represents the set of smart meters that collect the measured energy usage data in the home domain.

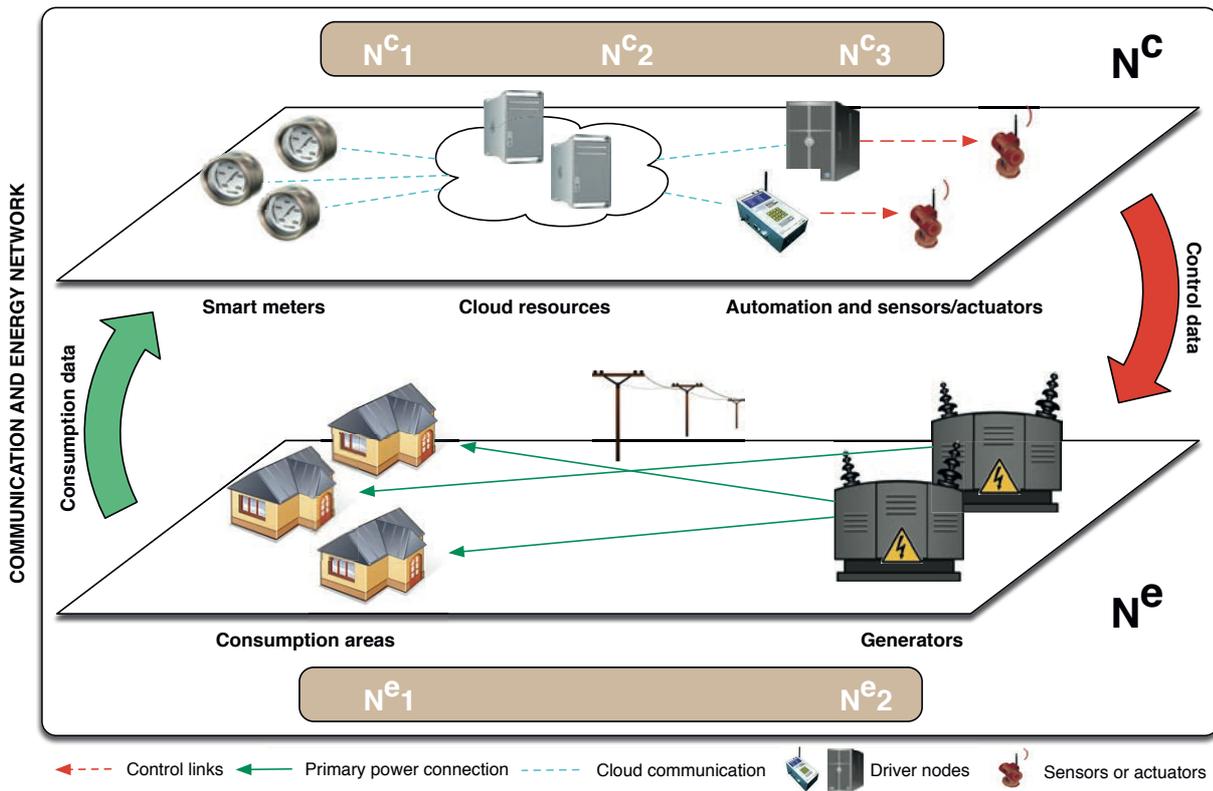


Figure 5.12: Five subnetworks-based architecture

$N_2^c$  corresponds to the cloud computing-based communication system to centralize all the computation and the forecasting process in nodes with high capacity to estimate new and nearby states, such as servers or proxies. In this way, it is possible to decouple the control processes and the demand management from additional computational processes that are required for the prediction.

$N_3^c$  embodies all the control and automation processes, required to protect the most critical underlying systems, such energy distribution and transmission substations. In this context, different cyber-physical elements are characterized such as acquisition and supervision elements working as driver nodes (e.g., RTUs, PLCs or gateways), and observation and control elements serving as sensory and reactive devices (e.g., sensors and actuators).

In real world, cloud resources belonging to  $N_2^c$  aggregate the information received from the users (via their smart meters embedded in  $N_1^c$ ) and compute an estimation of future consumption. According to this forecast, the generators of the production system are programmed by means of the actuators placed in the  $N_3^c$  subnetwork, which finally provide the electricity supply back to the consumption areas.

However, the conceptual construction of each of these subnetworks further entails working with aspects associated with graph theory and other concepts, related to structural controllability [306] and dominance [307]. For example, components of the subnetwork  $\mathbf{N}_1^e$  and  $\mathbf{N}_2^e$  are modelled on the basis of a random pattern, where the greater part of  $\mathbf{N}_1^e$  is permanently linked to  $\mathbf{N}_2^e$  elements (due to the fixed deployment of the energy distribution infrastructure) whereas a few nodes of  $\mathbf{N}_2^e$  are permanently connected to elements of each area in  $\mathbf{N}_1^e$  and driver nodes in  $\mathbf{N}_3^c$ .  $\mathbf{N}_3^c$ , to the contrary, follow specific network constructions centered on power-law distributions of type  $y \propto x^{-\alpha}$ , introduced in Section 4.1.2. This constraint is due to the structural features of real control infrastructures, which are based on multiple interconnected substations with a few industrial nodes (e.g, RTUs, sensors, actuators). This conceptually follows a hierarchical network architecture based on nodes with high degree (i.e., the number of edges incident on the node) connected to nodes with lower degree; similar characteristics to the power-law distributions as stated in [308] and [243]. The authors in [243], additionally, justify why other models are not applicable for power grids, such as the small-world distributions. According to them, the conditions given by, for example, Watts and Strogatz [245] are not satisfied by Power Grid samples due to physical and economic issues.

$\mathbf{N}_2^e$ , in turn, is based on specific grid distributions of type IEEE 118-bus or IEEE 300-bus as specified in [309], where we extract a subpart of these models to lead the practical case studies and the experimental results presented later.

To formalize the problem, we characterize two graphs, one related to  $N^e$  and another one to  $N^c$ . For  $N^e$ , let  $\mathcal{G}^e(V^e, E^e)$  be a directed bipartite graph, such that  $V^e$  is the union of the nodes in  $\mathbf{N}_1^e$  and  $\mathbf{N}_2^e$ , and the set of  $n$  customer areas in  $\mathbf{N}_1^e$  are connected to  $m$  grid generators of  $\mathbf{N}_2^e$  through grid connections in  $E^e$ . For the resilience and load balancing, we assume that each area is associated with  $\delta$  generators, such that  $\delta \geq 2$ . Within  $N^c$ , we consider  $\mathbf{N}_3^c$  to analyze the adversarial influence on the operational processes. Let  $\mathcal{G}_3^c(V_3^c, E_3^c)$  be a directed graph, containing the minimum set of driver nodes (referred to here as  $D_N$ ) capable of injecting control signals into the rest of the elements in  $V_3^c$ , also denoted here as the set of observed nodes (the set  $O$ ), such that  $\mathbf{D}_N$  and  $\mathbf{O} \subseteq V_3^c$ , all of them connected through communication links in  $E_3^c$ .

Under these conditions, several threat assumptions should be considered during the modeling and simulation of study cases. Firstly, the threats to be analyzed in this section are concentrated in  $\mathcal{G}^e(V^e, E^e)$  and  $\mathcal{G}_3^c(V_3^c, E_3^c)$ , where the adversarial model follows a weak approach, in which it is also assumed the attacker has high mobility in both subnetworks (to perform attacks against the power supply and the control network, respectively). The threats can be multiple and varied, where the adversary may target nodes or edges, and depending on the network, the interests may be very different. An attack in  $\mathbf{N}_1^e$  may, for example, focuses on producing concurrently anomalous deviations in the real demand and potentially overloading the power grid, misusing the energy during peak times. Contrarily, an attack in  $\mathcal{G}_3^c(V_3^c, E_3^c)$  may mean the constant removing of a few random communication links in specific nodes, simulating a denial of service. In this case,

the adversary's goal would be to alter the structural controllability to strategically unprotect the control itself and the functionality of  $\mathbf{N}_2^e$ .

Given this and the interconnected nature of the electrical systems and the technologies for the control and automation in real time, two independent, but narrowly related, approaches are presented here. These are intended to protect the following: (1) the processes of production and distribution of energy and (2) the control processes in response to unexpected changes which may also have a (mild, severe or irreparable) rebound effect on the dependent subnetworks (e.g., outages in  $\mathbf{N}_1^e$ , overloading in  $\mathbf{N}_2^e$ ) [310].

### Consumption Prediction and Load Balancing

Taking into account the aforementioned architecture, the first task required for the cloud infrastructure in  $\mathbf{N}_2^c$  is the ability to provide load balancing support to the generators according to the demand, for an effective electricity supply. Specifically, the main concern is the anticipation of upcoming peaks of demand, which could also be caused on purpose to cause blackouts in certain areas of the grid. By possessing this knowledge in advance it is possible to rapidly distribute the existing demand, at a given moment, among all the generators available in the grid (located in  $\mathbf{N}_2^c$ ), so that the affected consumption areas can keep receiving the requested energy and the continuity of the service is ensured.

In order to test the proposed load balancing algorithm in practical terms, it is desirable to firstly devise a way to simulate the generation of consumption data in real time. We intend to imitate the demand response under normal conditions and in the presence of anomalies (by introducing eventual outages in the data), with the aim of performing predictions that serve as input for the load distribution. This way, we can check the effectiveness of the algorithm against peaks and adverse conditions in a timely manner. For the sake of veracity when designing the generation of the bulk data that is used to check the accuracy of predictions, we have based our work on the datasets provided by the European Network of Transmission System Operators for Electricity (ENTSO-E) [3]. This organization represents 43 electricity Transmission System Operators (TSOs) from 36 countries across Europe, and provides hourly load values of all those countries at monthly intervals. Specifically, we have designed a custom dataset comprising all the hourly consumption values (in MW) of Spain from 2015, the last year for which data is available. If we show all these samples in a window of 24 hours, we obtain the graph in Figure 5.13.

As we can see in the figure, all the daily consumption values over the 365 days are plotted, resulting in a curve where most of the electricity demand is concentrated in the evening and decreases during the night. Based on this information, we use the actual data to define a mathematical function (henceforth the  $F$  function) that automatically generates consumption values indefinitely. For that, we perform a non-linear regression using the Gauss-Newton algorithm that finds a function of the type  $y = A\sin(Bx + C) + D$  that conforms to a set of data points

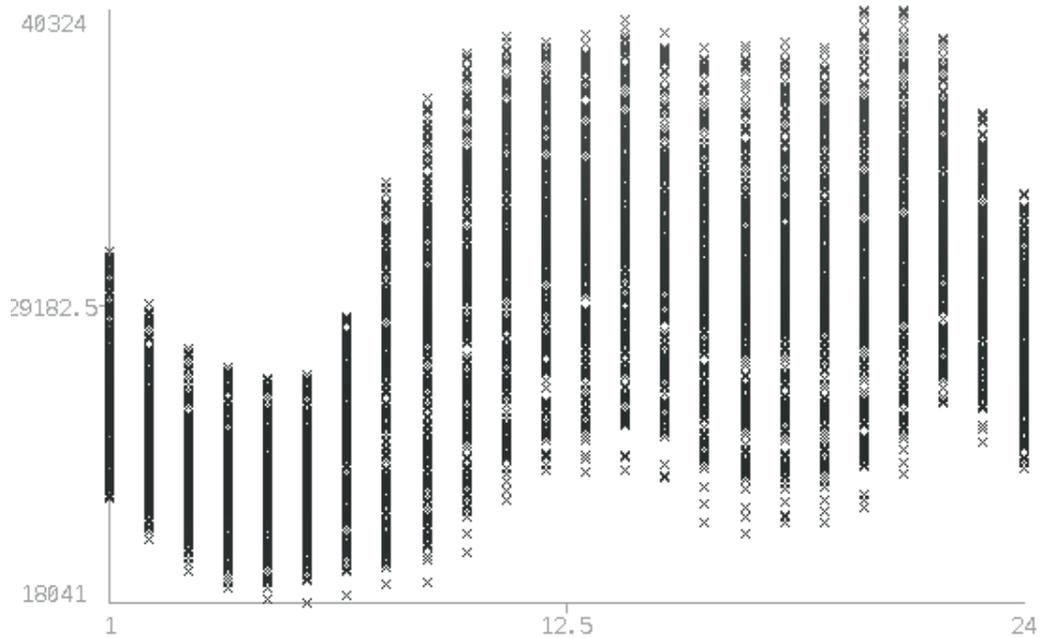


Figure 5.13: Hourly load values of Spain in 2015 [3]

$(x_i, y_i)$ . For the sake of clarity and the purpose of showing the efficiency of the prediction and load balancing method, we assume that the resulting consumption value only depends on the day of the week and the time of the day at which we want to predict the usage value (as independent variables of the function). Additionally, the month could be considered to analyze the influence of seasons. However, to provide a degree of randomness in the data and avoid returning the same value for a given set of input arguments (i.e., day of the week and hour), we consider adding certain deviations, whose value is arbitrarily chosen from a uniform distribution  $U(-\lambda, \lambda)$ , where  $\lambda$  represents the maximum divergence value. In addition, we have included the possibility of experiencing a peak of consumption (i.e., a considerable increase in certain values) under a probability  $\gamma$ . We must also mention that the original electricity values from the datasets have been divided by 100 to represent the conceptual consumption of a single area or province in Spain. By doing this, the demand of multiple consumption areas over the grid is simulated, which is accomplished through the execution of the aforementioned  $F$  function, in parallel, for several instances. Apart from this function to simulate the consumption, we must find a way to predict future values based on previous behavior. Altogether, this information will serve as input for the load balancing algorithm executed in the  $\mathbf{N}_2^c$  systems, that is finally responsible for the prevision of the energy supply for all areas within the grid.

Here, the prediction of the energy usage between neighborhoods is based on time series forecasting [311]. Contrary to traditional machine learning methods, which also work with multiple datasets but treat all the observations equally, time series adds an explicit order dependence to all of them: the *time dimension*. This gives higher importance to the last observations rather than

all data available, which is valuable for prediction. In addition, the analysis of time series can also determine seasonal patterns, trends or the relationship with external factors. In our case, the aim is to forecast future values of a time series, that is, the one described by the consumption curve. Specifically, we use the statistical model ARIMA, which stands for *AutoRegressive Integrated Moving Average*, and counts on three different components, expressed as  $ARIMA(p, d, q)$ :

- **Autoregression (AR):** use of a dependent relationship between an observation and a number of lagged observations, represented by the  $p$  parameter.
- **Integrated (I):** in order to make the time series stationary, it differentiates between raw observations (e.g., subtracting an observation from an observation at the previous time step). The number of times that the observations are differentiated is represented by  $d$ .
- **Moving Average (MA):** use of a dependency between an observation and a residual error from a moving average model. The size of the moving average window is represented by  $q$ .

These parameters  $(p, d, q)$  are characterized according to the general ARIMA model:

$$Y_t = -(\Delta^d Y_t - Y_t) + \phi_0 + \sum_{i=1}^p \phi_i \Delta^d Y_{t-i} - \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t \quad (5.6)$$

where  $\phi_1, \dots, \phi_p$  are the parameters of the autoregressive part of the model and  $\theta_1, \dots, \theta_q$  belong to MA, and the rest of parameters are part of the integration filter. Lastly,  $\varepsilon$  adds an error margin. The parametrization and accuracy of the ARIMA model for our purposes are discussed later, specifically in Section 5.4.1. The result of applying this model provides a set of future energy readings, taking into account the last consumption reports. As explained, once we have this information, the last step for load balancing consists in uniformly distributing the available electricity supplied by the generation resources in the grid among all the consumption areas at a given moment (which is represented with the graph  $\mathcal{G}^e(V^e, E^e)$ ), taking into account the forecasted value of the amount of requested energy by each of these areas.

In more detail, for the design of the load balancing algorithm, we have the following constraints. Let us assume a set of generators  $G$  of  $\mathbf{N}_2^e$  that supply electricity for a set of areas  $A$ . Each generator  $i$  has a maximum load denoted by  $g_i$ , and each area  $j$  demands  $a_j$  units of energy, having  $1 \leq i \leq |G|$  and  $1 \leq j \leq |A|$ . As initial conditions, we accept that:

- **C1:** there does not exist any area  $j$  whose  $a_j$  is higher than any  $g_i$ , for all  $i \in G$ . This ensures that every area can be supplied by at least one generator.
- **C2:** the sum of electricity requested by all the areas does not exceed the sum of electricity supplied by the generators; formally,  $\sum_{j=1}^{|A|} a_j \leq \sum_{i=1}^{|G|} g_i$ . This ensures that all areas can be provided with the requested energy.

Therefore, what we want to find is a relationship  $R \subseteq A \times G$  between areas and generators, such that each area is assigned with a generator and the sum of electricity requested by the

areas associated with a generator, does not exceed its capacity. This can be modelled as a search algorithm, since we explore a set of candidate solutions in the form of a tree, beginning with the initial one (an area is assigned to a generator) and gradually adding associations in the search for a valid solution, which is when all areas are assigned with a generator and **C1** and **C2** are consistently satisfied.

More specifically, we have designed a novel algorithm that makes use of backtracking, which is widely used for constraint satisfaction problems [312]. It incrementally builds candidates in the solution, and discards each partial candidate as soon as it determines that it does not comply with the proposed conditions, which makes it impossible for the candidate to be completed as a valid solution. The resultant technique is explained in Algorithm 10.

---

**Algorithm 10** Load Balancing (A, G)

---

**output**  $(R = \{(a_j, g_i)\}$  where  $1 \leq i \leq |G|$  and  $1 \leq j \leq |A|$ )

$R \leftarrow \{\}$

$R \leftarrow \text{SOLVELOADBALANCING}(A, G, R)$

**function** SOLVELOADBALANCING(A, G, R)

**if**  $|R| = |A|$  **then**

$Found \leftarrow True$  **return**  $R^a$

**else**

$Found \leftarrow False$

$j \leftarrow 1$

**while** *not Found* and area  $a_j$  not assigned and  $j \leq |A|$  **do**

$i \leftarrow 1$

**while** *not Found* and  $i \leq |G|$  **do**

**if** energy assigned to generator  $i + a_j \leq g_i$  **then**

$R' \leftarrow R \cup (a_j, g_i)$

SOLVELOADBALANCING(A, G, R')

**end if**

$i \leftarrow i + 1$

**end while**

$j \leftarrow j + 1$

**end while**

**end if**

**end function**

---

<sup>a</sup>A solution where an assignation has been found

---

In this case, a partial candidate represents a relationship R where not all areas are assigned to a generator. As described, the algorithm begins by assigning one random area to one random generator, and keeps iterating in the search for a valid solution, assigning new areas to generators if their capacity still allows it and recursively calling the function (which is modelled by the inner loop of the algorithm). Otherwise, the partial candidate is discarded and another area is assigned in the first loop of the algorithm. Thus, the *Found* variable finally indicates whether, or not,

there is a feasible relationship between areas and generators that successfully distributes the energy, complying with **C1** and **C2** conditions.

### Fault Detection and Control Protection

Together with the prediction and load balancing algorithm that ensures the safety of the power supply infrastructure, the other task required for the resilient architecture consists in the security of the control elements belonging to  $\mathbf{N}_3^c$ , represented with the graph  $\mathcal{G}_3^c(V_3^c, E_3^c)$ . We aim to secure the structural controllability domain by applying a distributed decision algorithm that enables us to detect subtle changes in the underlying network, that may be the result of a stealth attack. If we assume a set of finite agents uniformly distributed over the industrial network (named driver nodes in Section 4.1.1), it is possible to execute cooperative algorithms that allow them to accurately identify which parts of the topology have suffered changes (i.e., playing the role of agents). This, in turn, is determined by exchanging information about their surroundings with each other. This information can be used to deploy effective recovery techniques to guarantee the continuity of the service, as we have demonstrated with the APT traceability framework. In this section, we propose a first approach for its applicability in a Smart Grid scenario, based on the detection of topological changes, following a similar model to the one described in Section 5.1.

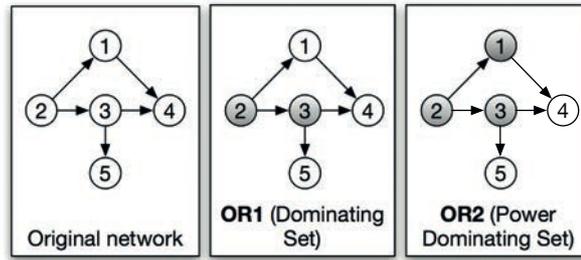


Figure 5.14: Observation rules for the election of the driver nodes

In this applicability analysis we have considered that the correlation is conducted by a subset of nodes within the topology (the driver nodes), implementing a distributed deployment of physical agents. More specifically, the set of driver nodes ( $\mathbf{D}_N \in V_3^c$ ) can be selected according to the following two rules, the **OR1** and **OR2** conditions, that were introduced in Section 4.1 and are represented in Figure 5.14. This means that every edge in  $E_3^c$  is adjacent to at least one member of  $\mathbf{D}_N$ .

Therefore, the detection algorithm between these agents is a light modification of the Opinion Dynamics algorithm described earlier, processed by each driver node  $n_d$  in  $\mathcal{G}_3^c(V_3^c, E_3^c)$  in discrete time. This approach creates a fragmentation of the affected zones within the network once the agents share information about their surrounding topology.

For the computation of Opinion Dynamics it is necessary to define a matrix  $W$  of size  $n \times n$  (with  $n = |V_3^c|$ ) holding the weights that represent the confidence between the agents'

opinions. Each agent assigns a weight to the rest of agents in its surroundings (particularly those nodes sharing a communication link) based on the closeness between their opinions. Altogether, the vector  $x_{nd}(t)$  holds the opinion of each driver node such that it is updated by  $x_{nd}(t+1) = W(t, x_{nd}(t))x_{nd}(t)$ , where  $t$  refers to another iteration of the algorithm. The logic of this equation is equal to  $x_{nd}(t+1) = w_{i1}x_{nd_1}(t) + w_{i2}x_{nd_2}(t) + \dots + w_{in}x_{nd_n}(t)$  such that  $w_{ij} = 1$ . Each opinion is originally calculated according to the new state of the network with respect to the original topology, which is computed with the difference in the node *betweenness centrality*. This way of representing structural behaviors was introduced in 4.1.1 and characterizes the principal control loads in  $\mathbf{N}_3^c$  [313], assuming that the main network control dynamics flow through the shortest paths. Therefore, any topological variation impacts on  $BC$  and subsequently on the new upgrading of  $x_{nd}(t+1)$  in time  $t+1$ . Once we execute the Opinion Dynamics with  $t$  tending to infinity (i.e., a high number of steps), it is possible to visualize the consensus between clusters of agents about topological changes on different parts of the network. Opinions  $\simeq 1$  mark topological changes within  $\mathbf{N}_3^c$  that are generally located in the surroundings of those local driver nodes that detect the deviation. This also means that a persistent, yet subtle change, over time, with values close to or exceeding 0.5 can mean the approximation of a structural change.

To prove the effectiveness when detecting such topological changes, we must simulate the action of a stealth attack, taking its nature into consideration. Particularly, these mutations appear as a consequence of the lateral movements taken to find new victim nodes and hence gain influence within the network. These attacks have to be planned strategically instead of leading arbitrary attacks, where the target must be focused on the control and its dynamics. Based on the general attack behavior described in Section 5.1, we have defined three different attack models:

**STG1:** the attacker focuses on an arbitrarily chosen node within the network and performs a change on any of its adjacent edges, to subsequently move to a neighbor node in a random way.

**STG2:** the threat is concentrated on those driver hubs with the highest degree  $d^+$  and  $d^-$ , where the attack aims to randomly remove a few edges.

**STG3:** the adversary is able to attack the node with the highest influence over the control by simply observing the traffic and its bandwidth. Through graph theory, this representation is possible through the highest *edge betweenness centrality* of the neighborhood, as specified in Section 5.1.

Taking into account these three threats, Algorithm 11 outlines the life cycle described by a stealth intrusion like this. It takes the original network described with  $\mathcal{G}_3^c(V_3^c, E_3^c)$  and performs a succession of individual attacks against the edges  $E_3^c$  (i.e., either the addition or removal of incoming or outgoing edges), resulting in the modified network represented with  $\mathcal{G}'_3(V_3^c, E'_3)$ . After each edge modification, the attacker propagates to an adjacent node in accordance with

one of the strategies presented before. At this point, the Opinion Dynamics algorithm can be executed to detect the portions of the network that are affected by the attack.

---

**Algorithm 11** Stealth attacks life cycle

---

**output:**  $\mathcal{G}'_3$  representing the resulting matrix  $M$   
**local:**  $\mathcal{G}_3^c(V_3^c, E_3^c)$ ,  $numOfAttacks$ ,  $STG_x$   
 $attackedNode \leftarrow \text{random } v_i \in V_3^c$ ;  $\mathcal{G}'_3 \leftarrow \mathcal{G}_3^c$

**for**  $i:=1$  **to**  $numOfAttacks$  **step** 1 **do**  
     $attack \leftarrow \text{randomAttack over } attackedNode$  (edge addition or removal)  
    update  $\mathcal{G}'_3$  based on attack  
    **if**  $STG_x = 1$  **then**  
         $attackedNode \leftarrow \text{random } v_i \in V_3^c$   
    **else if**  $STG_x = 2$  **then**  
         $attackedNode \leftarrow \text{NEIGHBOURWITHHIGHESTDEGREE}(M, attackedNode)$   
    **else if**  $STG_x = 3$  **then**  
         $attackedNode \leftarrow \text{NEIGHBOURWITHHIGHESTBETWEENNESS}(M, attackedNode)$   
    **end if**  
**end for**

---

## Experimental Results and Discussions

After successfully designing mechanisms to firstly ensure the safety of the grid and also the security of the control elements involved, our aim is to test these services in practice.

To start with, we have to implement the  $F$  function in charge of generating the consumption plot that in conjunction with the information provided by the prediction process, serves as input to the load balancing algorithm. As previously described, we have leveraged the annual consumption dataset in Spain as of 2015 to adjust a nonlinear correlation of the data to create the  $F$  function. This function simulates the consumption for a specified *hour* and a *dayOfTheWeek*, over which we have also added some extent of randomness  $\lambda$  (here we assume  $\lambda = 15$ ) and a potential *peak* (a value of 50 has been considered) in the energy usage under a given probability  $\gamma$ . The result is the following expression:

$$F = 100 * \cos(hour/3.82 + \pi/3) + 100 - 8 * dayOfTheWeek + \lambda + peak \quad (5.7)$$

Where the output value is expressed in MW, and holds the value of consumption for certain regions within the grid. For example, Figure 5.15 shows the result of executing the  $F$  function for an entire week (i.e., showing the evolution over its 168 hours), with a peak probability of 5%. Taking a close look, we can rapidly see the two peaks produced on Monday and Friday at night.

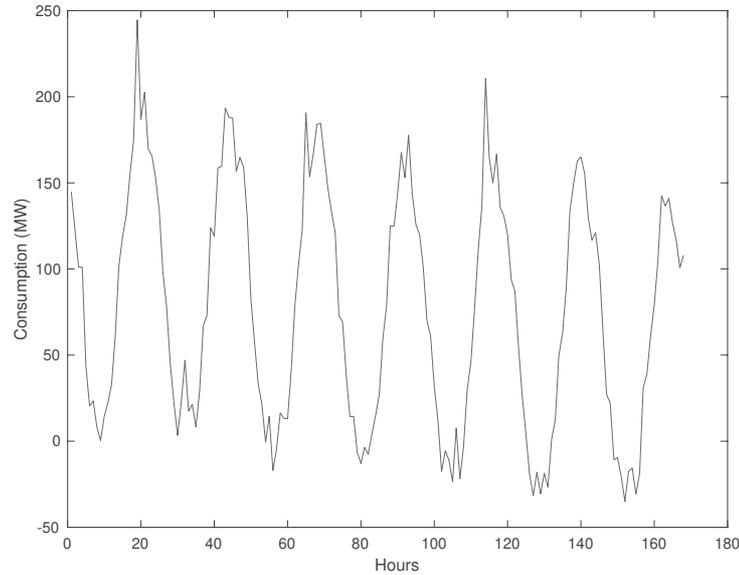


Figure 5.15: Weekly consumption generated by  $F$  function

It is also clear that the overall progression evolves towards a lower consumption as the weekend approaches.

Once we have modelled the  $F$  function and we are able to successively generate consumption values over a time period, we move on to parametrize the ARIMA statistical model so as to treat the consumption output as a time series and perform the forecasting. To find the optimal value for the  $p$ ,  $d$ , and  $q$  parameters, it is necessary to follow a formal methodology that estimates each one by examining the AR or MA behavior of the series and testing with initial values to subsequently analyze how the model fits the original data [314]. For this purpose, the Simple and Partial Autocorrelation functions (AFC and PACF, respectively) are used. Once the appropriateness of the model has been compared, its residual errors are checked with the Akaike Information Criteria (AIC). For our particular case of forecasting the consumption time series, we have automated this process through the R *forecast* package [315], which enables the estimation of its coefficients and also gives a ratio of likelihood. For example, if we gather the consumption values of ten days, it determines that the  $ARIMA(3,0,1)$  model is suitable to fit the information, the value of which is computed as follows (taking into account Equation 5.6):

$$Y_t = a_1 Y_{t-1} + a_2 Y_{t-2} + a_3 Y_{t-3} + b_1 \varepsilon_{t-1} + \varepsilon_t \quad (5.8)$$

After defining the model, it is possible to perform the prediction of upcoming days. Figure 5.16 represents the forecast of two more days after a given period of time, which shows the accuracy of the ARIMA when predicting the consumption curve, that follows the expected progression.

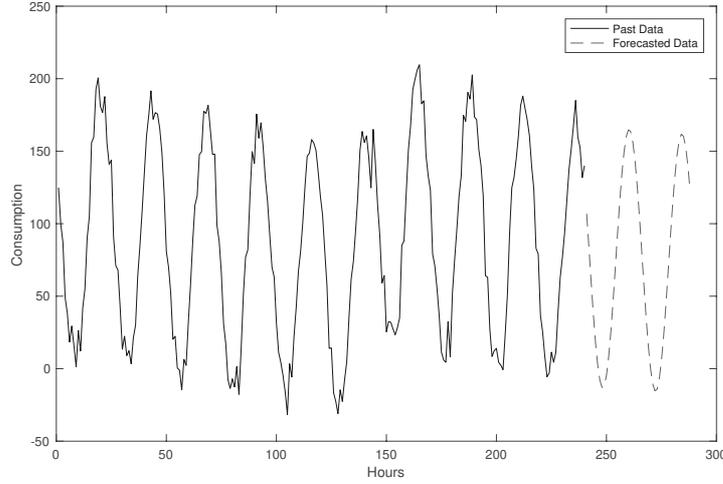


Figure 5.16: Forecast after 10 days using ARIMA

Once we have the information about the future status of the grid at our disposal, we are in a position to execute the load balancing algorithm that uniformly distributes the electricity demand among all the generators available. Specifically, the energy usage prediction for all the individual areas spread over the Smart Grid provide sufficient input to the utility to carry out demand response. Recalling the concepts of the proposed architecture previously introduced,  $\mathbf{N}_2^e$  represents the energy distribution infrastructure, composed by the generators and substations that supply the electricity to the consumption points (e.g., neighborhoods and electric vehicle charge points). These assets are interconnected following the network described by the graph  $\mathcal{G}^e(V^e, E^e)$ , where we assume there are  $\theta$  areas demanding energy to  $\delta$  generators.

In the interest of veracity and taking into account that the aforementioned network remains rigid in its topology and configuration state, we have considered the IEEE-14 and IEEE-57 bus systems to carry out simulations based on a real-grid test case [316]. Both of them consist of a simple approximation of the American Electric Power system as of the early 1960s. The first system has 11 loads (assumed to be the areas of consumption for our purposes) connected to 5 generators, whereas the second model has 7 generators and 42 loads. A test case has been defined for each one, with as many areas ( $\theta$ ) and generators ( $\delta$ ) as each system respectively defines. We have supposed that every generator  $i$  in the  $G$  set has a maximum load  $g_i$  that is randomly selected in a defined interval, and each area  $j$  demands  $a_j$  units of energy whose value is, at most, the maximum value of capacity for a single generator. Taking these parameters into consideration, Figure 5.17 shows the simulation of the load balancing algorithm for the IEEE-14 and IEEE-57 systems, where we can see how the consumption areas are accommodated to the available generators. To simplify, we have considered a maximum of capacity per generator of 10MW and 15MW, respectively.

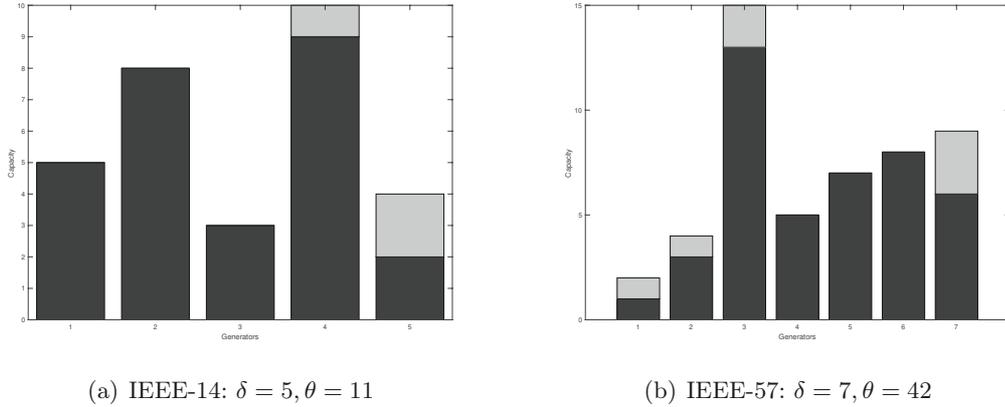


Figure 5.17: Load balancing for the two proposed systems

So far, we have put into practice the mechanism that preserves the availability of the AMI infrastructure in its safety dimension. As for security, we now show the effectiveness of the intrusion detection technique based on Opinion Dynamics. For this, we have randomly created a network of a power-law distribution composed by 100 nodes, where we have conducted a set of 50 topological attacks as described in Algorithm 11. If we run the Opinion Dynamics algorithm over the set of 70 agents (which are driver nodes of  $\mathcal{G}_3^c$ ), we can check how the opinions evolve to reach a consensus and create different clusters within the network. More specifically, Figure 5.18 shows how the total number of agents of the network are divided into substantial sets depending on the degree of change, for the three attack strategies that we define in Section 5.4.1.

In these plots, each line represents the change in the opinion of the corresponding agent when the algorithm is executed over 50 steps ( $t = 50$  in the Opinion Dynamics algorithm). Altogether, the presence of big clusters of opinions means a confident consensus of agents about a change experienced in a particular area, whose level of criticality is higher as it approaches 1. This is particularly evident in the **STG3** test case, where most of the agents agree on a topological attack in a specific part of the network, with approximately 60% of change. This behavior occurs due to the attack model chosen: the attack in **STG3** always propagates to nodes with higher influence on the control (i.e., a higher *betweenness centrality*, this is, the driver nodes), which makes it easier for the agents to locate the subtle changes (and is also the most realistic pattern, since the attacker commonly aims to gain the control of the network). This result is somewhat similar to **STG2**, because it is expected that those nodes with a higher degree are precisely the ones that have greater hierarchy over the network. However, since **STG1** focuses on propagating the attack in a random way, it is harder for the agents to reach a consensus on the portions of the network that are affected, resulting in a fragmentation of multiple opinions. On the whole, this constitutes a valuable insight into deploying accurate response techniques to overcome the effects of one of these threats.

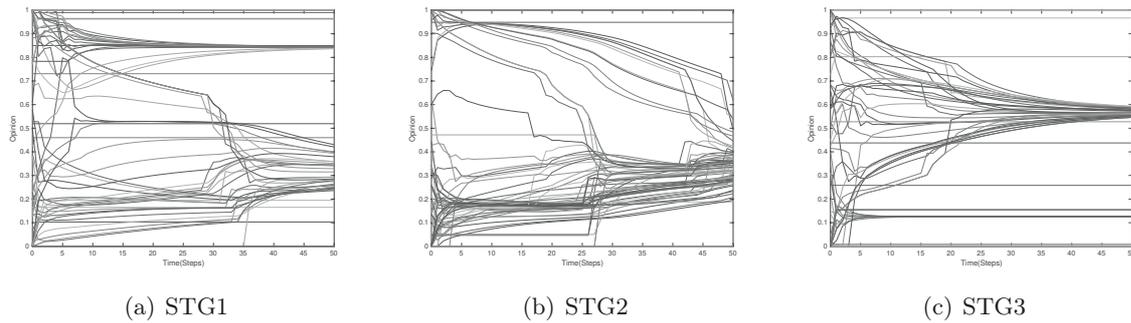


Figure 5.18: Opinion dynamics after 50 attacks

### 5.4.2 Context-awareness Manager for Authorization Policies

With the introduction of the latest information, communication and operational technologies, the Smart Grid allows the utilities to accurately monitor energy consumption so they can adjust generation and delivery in near real time. It also helps users get detailed consumption reports which are useful to save money by adapting power usage to the price fluctuation. However, the power grid is also exposed to multiple cybersecurity threats and privacy issues inherited from the ICT sector, which might end up becoming part of an APT. This section highlights the contributions of SealedGRID [317], an EU H2020 project which addresses the protection of the SG against these and other sophisticated attacks, providing a scalable, highly trusted, and interoperable SG security platform. It is applicable to modern industrial networks as well as traditional control infrastructures like SCADA and telemetry systems, abiding the existing standardization work.

One of the main cybersecurity challenges with the Smart Grid field nowadays is the need to implement an Access Control Management Service to control the information within the grid. These services are essential to manage permissions of users, peripheral devices or programs when they request to use certain resources within the infrastructure. The integration of IT technologies and especially the cloud hinders the application of conventional access control models in industrial systems (and particularly in the Smart Grid), for different reasons. These can be summarized in the sharing of information among heterogeneous entities with different degrees of sensitivity, performance and regulations. In this complex scenario, access control mechanisms deployed (either in field devices, PLCs or cloud resources) aim to restrict what each entity should be able to access and the connections that can be accepted, having the ability to deal with a diversity of devices [318]. Current solutions are still in their infancy, due to the need for a dynamic and fine-grained mechanism that deals with several users and constrained resources. Therefore, it becomes mandatory to analyze the full range of requirements that access control presents in the upcoming scenario, in order to accurately tailor the available models and propose new approaches that meet these conditions.

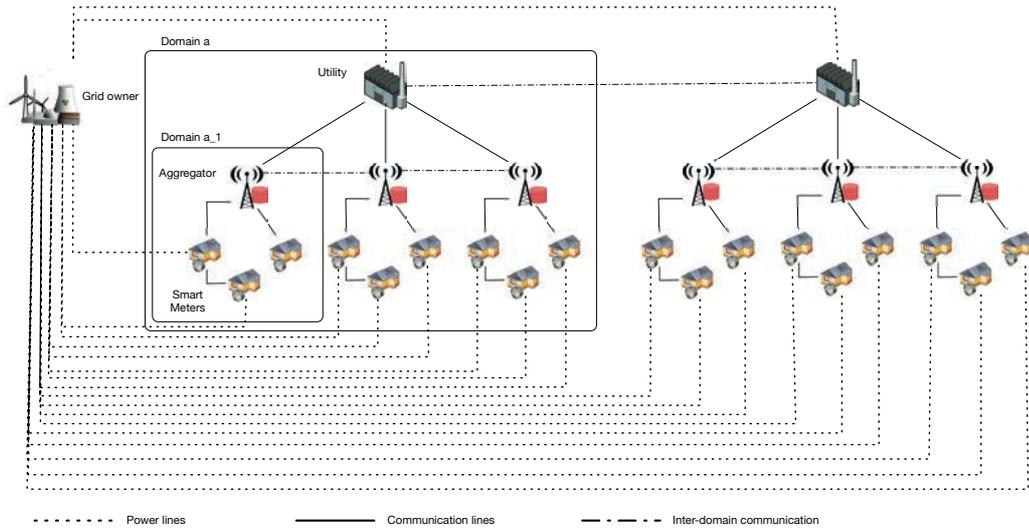


Figure 5.19: Overview of the SealedGRID infrastructure

At the same time, it is also crucial to pair this control with the continuous assessment of the network in terms of security, as to permit or deny the use of certain services in case of risk. This is usually enabled by **context-awareness mechanisms**, which retrieve data about the production chain in real time (e.g., network events, alarms, raw traffic). Here we will show how the traceability system based on Opinion Dynamics can be applied to detect attacks against the grid infrastructure and keep track of the such incidents over time, as to assist the authorization systems in a holistic manner.

### Authorization in the Smart Grid: Background and Terminology

Firstly, we need to lay the base for the subsequent sections, by proposing a common terminology and set of requirements for the design and implementation of an access control mechanism that takes into account the security of the power grid in real time. More specifically, we will establish assumptions with respect to the types of devices, actors and resources involved in the SealedGRID architecture:

- The **Smart meter** is responsible for collecting electricity consumption readings.
- The **Aggregators** are intermediate nodes between the collector and the smart meters, which sum the individual readings received by the meters and transmit the result to the collector.
- The **Utility** accumulates high-frequency aggregated values. It can either use these values as is for demand response (e.g., control the electricity consumption in a specific area) or sum them for billing purposes.

These entities are represented in Figure 5.19. In SealedGRID, with respect to the internal architecture of these entities, they embody different internal modules in charge of performing the cryptographic operations (i.e., key management, authentication) as well as the authorization procedures, which are coupled with privacy preserving techniques that ensure the security of the overlaying applications running on top of these devices.

We also need to lay the base for the different components involved in the authorization process. In the Smart Grid scenario, when different domains are interconnected to each other and collaborate, it is common to apply authorization frameworks based on the presence of Policy Information Points (PIPs), Policy Enforcement Points (PEPs) and Policy Decision Points (PDPs). These are entities that uptake different responsibilities on the authorization procedure (i.e., the decision of whether granting access to a resource that has been requested):

- **Policy Information Points (PIPs):** these are processes that are strategically located everywhere across the Smart Grid (both in the household and the aggregators and utilities) to gather as much information as possible for computing the access decision on the PDP. This information is extracted continuously by the context-awareness module, which will be later explained. In practice, all the sealedGRID components will be considered as PIPs, since they provide information with the aforementioned module.
- **Policy Enforcement Points (PEPs):** these are the processes that perform the requests to the rest of the SealedGRID devices when required. More specifically, these requests are relayed to the corresponding PDP in charge of controlling the access to the protected resources, together with the information gathered by local PIPs.
- **Policy Decision Points (PDPs):** these entities finally take the decision of whether permitting or denying the access whose request has been received, applying the defined control-access policy. At the same time, it also manages the authorization process.

For our particular concern, we will use a hierarchical architecture for the design and implementation of the authorization components (i.e., the PIPs, PEPs and PDPs). In that architecture, there are multiple roles (e.g., users, operators, security administrators) spread over the topology, which are studied when designing the actual control access policy. For our analysis, we can assume that such policy is implemented through a hybrid access control mechanism based on RBAC (Role-Based Access Control) and ABAC (Attribute-Based Access Control). For the actual formalization of the access-control rules, the IEC 62351 standard can be used, in order to follow a common framework of policies applied in the SG context. This is a reference in the sector to address the security of industrial networks [79]. It is composed of eleven parts, where part 8 is especially applied to control access mechanisms.

Here, we particularly focus on the assignation of PIPs, PEPs and PDPs to the infrastructure components. On the one hand, all elements of this infrastructure can be considered as a PIPs

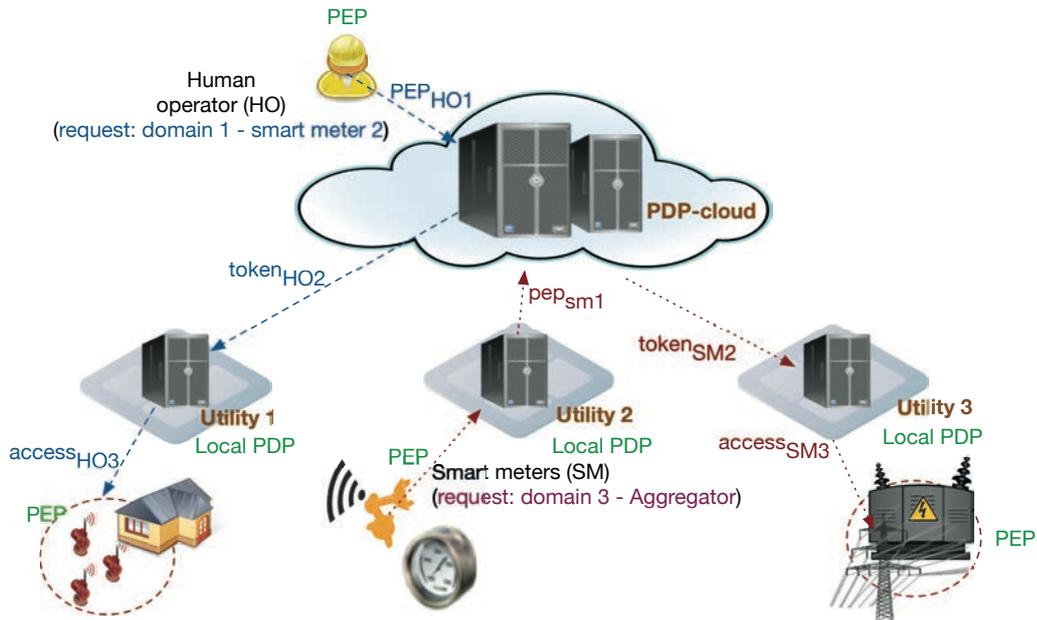


Figure 5.20: Hierarchical architecture of the PEP and PDP entities

and PEPs, since all the SealedGRID devices embody a context-awareness module that provides information at all times to ease the access control decision. At the same time, all the devices can potentially submit an access request to the rest of elements within the same domain or others. For instance, a smart meter could potentially access its local aggregator and then this device could access its associated utility or other aggregators.

As for the decision points, different issues must be addressed. On the one hand, we aim to enable multi-domain scenario where multiple utilities actively collaborate in a certain region for the management of the power supply, which results in a federated grid that involves several partners. In other words, we assume the existence of more than one utility interconnected. From the authorization perspective, this leads to the necessity of creating a global access control policy that applies rules for the secure and interoperable access between resources that belong to different domains. As a consequence, the authorization component has to accommodate two classes of PDPs: one global PDP to the entire system and another localized in the utility that allows it to locally apply the policy to the regions that it controls.

At this point, it is the subject of study to implement the global PDP using a cloud computing infrastructure. The reason is that we need to centralize the definition and readjustment of a global access control policy to the whole set of utilities underneath. This way, by placing PDPs on the individual utilities in a local way, we reduce the overhead introduced in the decision computation, since all requests that involve the local access to resources within a domain can be effectively resolved by the delegated utility. The use of utilities as intermediate PDPs with the global one thereby allows them to periodically update their policy rules by fetching the new changes from the

cloud. In addition, this procedure can be also carried out in a domain level, by placing low-level PDPs in the precise aggregators when the requests concern devices in a localized area. For this purpose, some computation nodes in the edge of the network or the fog computation technology can be leveraged.

Therefore, we distinguish between two types of PDPs: a global PDP in the cloud (referred to as ‘PDP-cloud’), and local PDP (denoted by ‘local PDPs’). Both are also illustrated in Figure 5.20, which represents the hierarchical architecture of these PDP entities at all levels and shows how remote stakeholders can gain access to resources by using PEP instances through the PDPs places in the domain or leveraging the PDP-cloud. Altogether, this design simplifies the centralized actions in the cloud and any occurrence of bottlenecks between domains.

For the interest of our analysis with respect to the use of Opinion Dynamics as a context awareness mechanism in this scenario, we now describe the operational architecture of the Global Policy Decision Point, that takes care of the security of all elements within the power grid.

### Global Policy Decision Point (PDP-cloud)

As explained earlier, the global PDP is shaped in the cloud to mainly conduct two specific tasks with respect to the authorization, which impose special requirements in terms of computation:

1. **To receive information of the context from each PIP deployed in the Smart Grid infrastructure.** This means that the PDP-Cloud offers an overview of the security state of the whole system at all times. Also, it allows to execute further accountability and auditing procedures, in such a way that the current policies are continuously assessed for the entire set of domains in order to readjust them in real time depending on special conditions of security and network overload.
2. **To define the global policy and roll out updates to the entire set of domains.** Eventually, the PDP-cloud performs individual access decisions that involve the use of resources between different utilities.

As for the architectural design of this PDP entity, it is composed by two chief components, which are also depicted in detail in Figure 5.21:

**PDP manager:** its main operation is to validate the authentication tokens provided by each entity and perform the access decision based on the received request, the defined policy rules and the information of the context (provided by the context awareness manager). In specific, it comprises the following internal modules:

- **Authentication module:** this procedure is required to validate the identity of the entities that submit the access request. It involves not only SealedGRID devices (that may perform a local authentication), but also human operators, engineers or customers using mobile

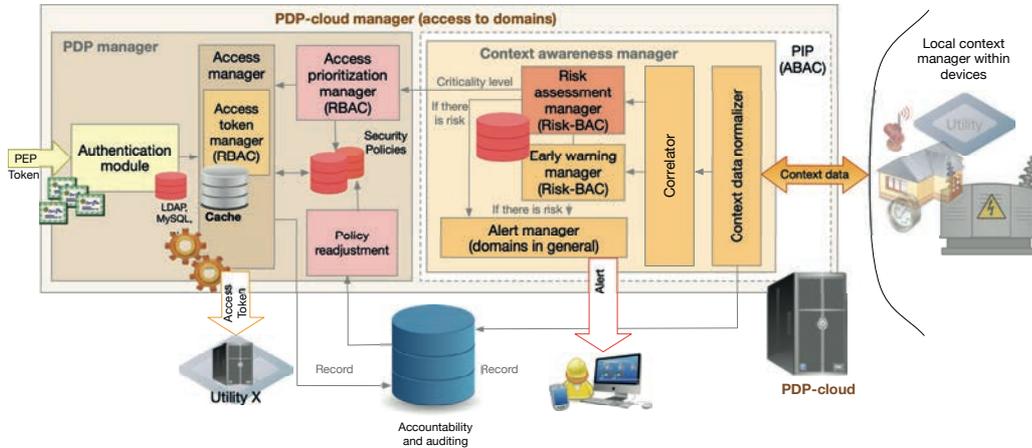


Figure 5.21: Architecture of components of the PDP cloud

devices, for which the use of some authentication mechanisms (e.g., OpenID protocol) might be required. In general, this module accepts token that have previously signed with the appropriate certificates of authority. Once they have been validated, the token is processed with the access manager.

- **Access manager:** it is the core of the PDP manager, where the access decision is computed taking into consideration the access token received and the information of the context (which informs about the security state of the elements involved in the requested access). These tokens contain information about the previous authentication process and specific information generated by the PEP request; at least, the identity of the resource requested and its corresponding domain, together with the action to be performed. Based on this information and the policy defined (including roles and permissions), the access decision is finally computed.
- **Access prioritization manager:** it is activated with higher priority to compute rapid access decisions in the event that potential threats are detected by the context awareness manager. For this reason, it receives feedback from the risk assessment manager. In any other case, it provides input to the access manager by serving information about the security state of the network in the zones concerned by the requested access, in order to compute a decision.
- **Policy readjustment:** this module performs advanced analytics procedures to gain insight from the previous access requests and data retrieved by the distributed context-awareness modules within the Smart Grid infrastructure. Its aim is to redefine the policy rules based on environmental and security conditions that may fluctuate over time. In this sense, we address in the next section the possibility of integrating further auditing procedures, possibly

using some DLT such as a blockchain network. It is important to note that the local PDP lacks this policy readjustment module, as it is the responsibility of the PDP-cloud to define the policy rules and notify the local PDPs about potential updates. Other than that, the architecture of the local PDP is essentially the same, and the only difference lies in the decisions of the latter being hierarchically lower within the grid.

**Context awareness manager (PIP):** it has the responsibility to retrieve and gather the information provided by the distributed elements of the Smart Grid network, provided by their embedded context-awareness modules (that implement the PIP functionality for all the SealedGRID devices). The context awareness manager is structured into different modules:

- **Context data normalizer:** this element is in charge of gathering the data from the local context awareness modules embedded on the SealedGRID devices, and normalize this information to extract multiple indicators and filter out noise. The aim is to provide profitable, useful and accessible information in real time to inform the rest of the PDP modules about the security state of the network.
- **Risk assessment manager:** as early introduced, this module computes health indicators to identify potential anomalies caused by sophisticated attacks, which are measured by the context awareness modules of each device. This is represented with global and local health indicators.
- **Early warning manager and alert manager:** their aim is to analyze the most critical threats detected to rapidly alert the operators and activate protection mechanisms that successfully reduce the impact on the system. At the same time, this information is loaded in a database for future risk assessments, in which a set of parameters are evaluated, such as the frequency of attacks, the criticality of the affected resources, etc.

In order to better explain the dynamic interaction between the modules of these two chief components (PDP manager and context awareness manager), Figure 5.22 shows the sequence diagram that represents the authorization flow in a general basis. Firstly, the remote PEP generates a token that is signed with its own certificate of authority (i.e., using federated login), which is validated once it reaches the PDP (through the Authentication module), and then the token can be processed by the access manager. In order for it to compute a decision, it firstly checks whether the request matches the system access control policy; at this point, if the permissions are insufficient, the access cannot be granted and the decision token is returned to the PEP. Otherwise, the security state of the network is checked for the resources to be accessed, for which the context awareness manager is leveraged. More specifically, the Access manager queries the Access prioritization manager, which is permanently fed with the information provided by the Risk assessment, which is always assessing the security state of the network in real time.

As depicted in Figure 5.21, this module receives input from the correlator, which analyses the information provided by the local PIPs installed in every SealedGRID device. For this task, the context awareness manager makes use of the Opinion Dynamics correlation algorithm studied in this document. Back in the PDP manager, once the Access manager has the security-related information about the concerned resources, it finally computes an access decision. This decision is notified to the PEP by means of a token. Also, as an additional step, this decision is stored in a historical database for accountability purposes, which helps to debug and enhance the current security access control policy through advanced readjustment mechanisms, as explained in the next section. Before that, we give some details on how to implement the Opinion Dynamics approach in this Smart Grid environment.

#### **Development of the context awareness manager**

Based on the previous description, the PDPs grant access to the grid resources depending on both static conditions (i.e., the access control policies) and the security state of the assets whose access is requested. This information is provided at all times by a context-awareness module embedded in every component of the grid infrastructure. This module hence implements the PIPs, which permanently provide useful information for computing the access decision on the PDPs (i.e., through the Risk Assessment manager). This information must comprise information about the quality of service and the security status of the component in question (i.e., a smart meter, aggregator or utility), which is also known as its context. In other words, they play the role of the agents in charge of assessing the anomalies, as introduced in the traceability framework of Section 4.2.1. In the following, we summarize the main set of aspects that should be measured by the proposed context-awareness module embedded in every SealedGRID device:

- **Current operation of the device:** it implies accounting for the behavior of the analysed component in real time. For example, keeping track of the number of energy usage readings measured by a smart meter, the set of households controlled by an aggregator (together with their current demand for control procedures), and detailed information about the utility and all its branches, from a technical perspective (including every update on their systems, databases or contractual arrangements that could be minimally required to correctly tailor the access control mechanism). This can also be used for auditing purposes and conduct a readjustment of policies over time based on data about the past, as addressed in further sections.
- **Quality of service information:** it involves examining the throughput of the concerned component at all levels:
  - **Host-based information:** it encompasses the usage of computational resources (CPU/RAM memory), electricity consumption, installation date, firmware version, etc.

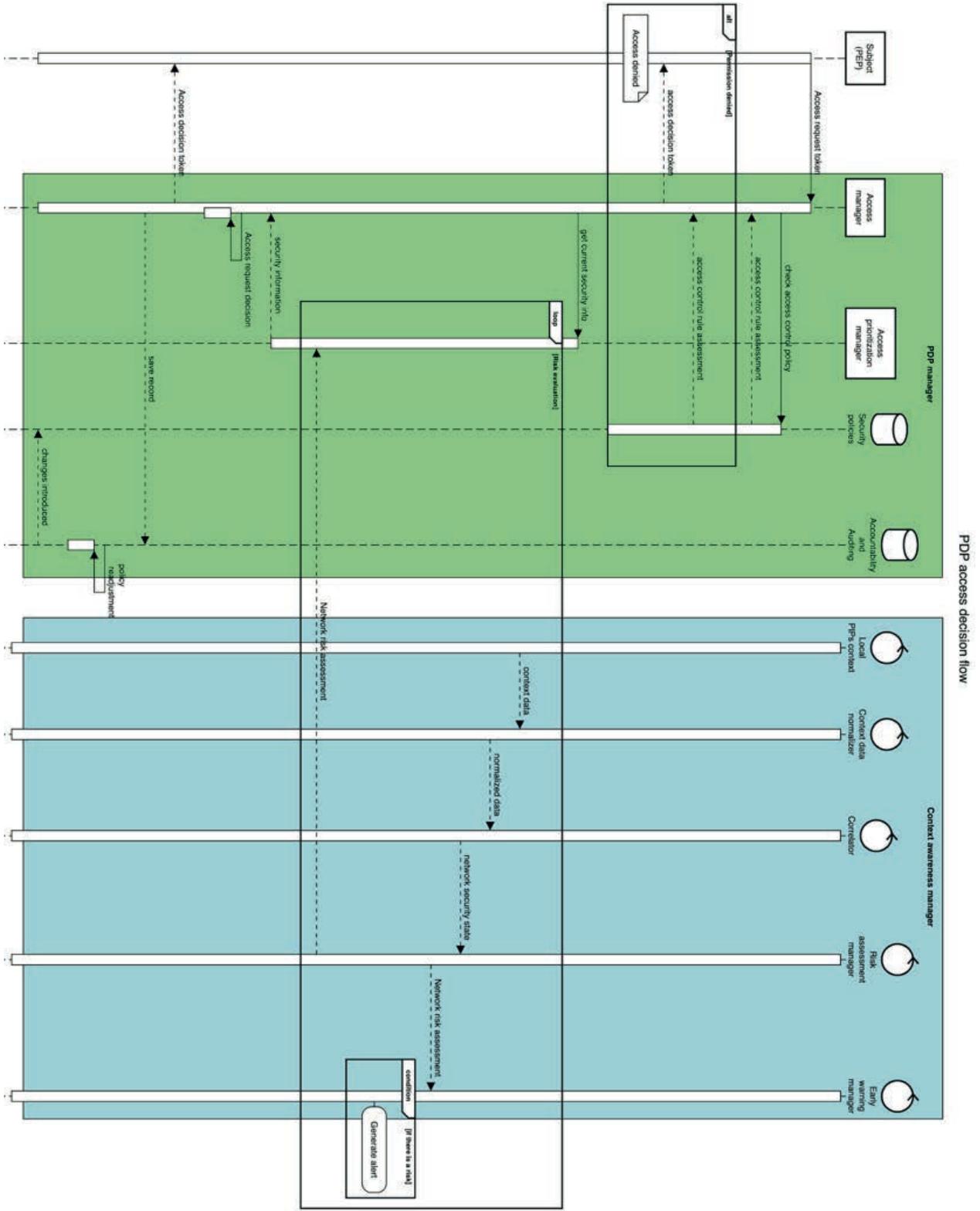


Figure 5.22: Sequence diagram for the authorization flow on the PDP-cloud

- **Network-related aspects:** communication protocol supported, types of commands used, connections opened with other smart meters / aggregators in multiple regions, etc. together with the QoS of every established communication channel (e.g., bandwidth, delays, packet loss ratio)
- **Security state information in real time:** at this point, numerous solutions can be put in place to assess the security of the component and detect potential threats, whose information must be also retrieved by the context-awareness mechanism. It includes the use of firewalls, antivirus, etc. and the parametrization of the security techniques in the communications and exchanged data: encryption algorithms, key size, etc. Additionally, it is especially interesting to extract profitable information about the real-time security of the components by means of IDSs, which could be integrated in the grid to potentially find anomalies with respect to their expected behavior.
- **Intrinsic information about the SealedGRID embedded modules:** in other words, it is meta-information about the own modules that are executed alongside the context-awareness mechanism on the same SealedGRID component. This may include, e.g., the supervision of key management procedures, the aggregation of consumption values for privacy purposes, etc.

Altogether, the goal of gathering information with this degree of heterogeneity is to create a virtual representation of the whole grid infrastructure, containing as much knowledge as possible for the different scenarios. It results in profitable information returned to the decision points, since they have enough awareness as to perform complex access decisions based on dynamic criteria. In practice, the particular implementation of a context awareness mechanism that fulfills these requirements with respect to the heterogeneity of information spread over a fully distributed network is not trivial whatsoever, as already discussed in Section 5.4.1. Here, we leverage Opinion Dynamics to identify and trace sophisticated threats across a large infrastructure, by tracking and correlating anomalies sensed by a plethora of ad-hoc analysis techniques and external intrusion detection systems. These patterns can be visualized graphically and mapped into the different regions of the network, as Figure 5.23 shows. The formal definition and features of this approach have been deeply explained in Chapter 4.

Concerning the actual implementation of this approach on the grid proposed in the context of SealedGRID, the context awareness mechanism based on Opinion Dynamics must execute a set of phases, which involve the aforementioned modules of the context awareness manager (i.e., the Context data normalizer, the correlator, and the Risk Assessment manager). These phases are equivalent to the steps needed to uptake the network information acquisition in the APT traceability framework, explained in Section 4.2.1:

- **Information gathering and data retrieval:** first of all, every PIP individually measures information of all kinds on the device it is monitoring (as explained before, regarding

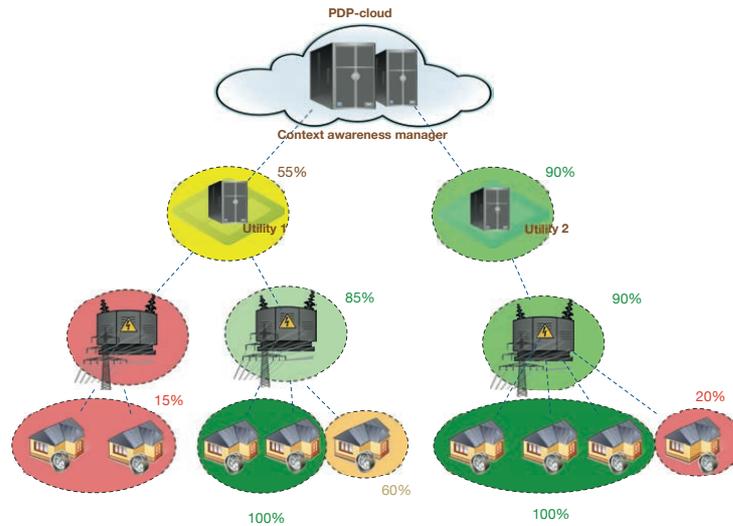


Figure 5.23: Network fragmentation due to Opinion Dynamics anomalies

the current device operation, QoS, security state and intrinsic information about the rest of SealedGRID modules). This data is aggregated in a bundle that is retrieved by the immediate PDP that periodically requests it to perform access control decisions, following the authorization flow.

- **Data recollection and normalization:** once at the PDP context awareness manager, the bulk data received by the complete set of devices that it is monitoring is normalized (i.e., the noise is filtered and the set of aspects measured are accurately selected) and stored for auditing purposes (i.e., the analysis of behaviors and patterns that lead to potential changes on the access control policies).
- **Detection:** after the data has been gathered, the information is analysed with the Opinion Dynamics algorithm, to detect intrusions and security issues that may pose a risk to the organization or the grid infrastructure. For this goal, the bulk information can be divided into different agents that are logically created (by means of parallel threads) or physically deployed, to individually process the data from their respective physical devices. In specific, the different aspects that are originally measured by the distributed PIPs are compared with their expected behavior, using the accountability database. The ultimate aim for every agent is to extract a single value that represents the degree of anomaly detected in the context of its device (i.e., the agent opinion). For this task, additional IDSs and further analysis procedures can be integrated to assess the different aspects measured (e.g., QoS, security).
- **Correlation:** finally, all anomalies detected by the set of logical agents are finally correlated using Opinion Dynamics. This results in a clusterization of opinions depending on the

affected zones, which permits to compute different indicators of health, at different levels (i.e., divided into zones, concerning the global infrastructure, etc.), as discussed in Chapter 4.

### 5.4.3 Readjustment of Intelligent Authorization Policies

From the previous section, we deduce that it is of paramount importance to introduce fine-grain control over the custody of sensitive data along the grid, while ensuring the democratization of the available infrastructure, by means of advanced authorization policies. The main challenge here is the overwhelming complexity when managing the flow of data between all parties involved. This heterogeneity of data is increased with the concept of microgrids, and the possibility that users also become prosumers; that is, that they can consume and also sell electricity to the grid. Added to this is the problem of privacy when analysing consumption data on a large scale (using Big Data), so that it is possible to extract information on consumption habits that is particularly useful for third parties. The latter is especially aggravated in federated SG environments, where several electricity companies actively collaborate to manage resources in different geographical areas.

On top of that, progress in telecommunications does not stop, and the industry is not unaware of this progress. New communication technologies such as 5G and innovative computing paradigms on the edge of the network (such as fog computing) are on the horizon [319]. Apart from them, there are other disruptive technologies that we already live with, such as the Internet of Things or the blockchain. At present, some of these mechanisms are already integrated in various industrial sectors (including the Smart Grid), in what is already known as Industry 4.0.

Despite these issues, far from adding more complexity to the Smart Grid infrastructure, these technologies can solve several of the problems we currently face. This ranges from increased process automation to secure data transmission and storage at all levels, accompanied by almost instantaneous transmission and analysis, with very high efficiency. Only in this way can we understand future industrial scenarios where data ubiquity is achieved as well as optimal interaction between all participants in the production chain.

However, these mechanisms can only be enabled by continuously monitoring all assets in terms of cybersecurity, in order to anticipate risks, generate evidence transparently and ensure the democratization of the available resources. As introduced in the previous section, all these measures must be governed by advanced authorization policies capable of adapting to lively environments with ever-changing technology and actors. By introducing ubiquitous processes that permanently conduct context-awareness analysis over the entire infrastructure, the authorization components are accurately fed with operational and cybersecurity inputs to improve decision making and later enhance the established policies [320]. As explained in Section 5.4.2, these processes act as the PIPs that are assigned with different tasks. Firstly, they retrieve contextual information from a layer of devices to be monitored, which includes low-level operational inputs

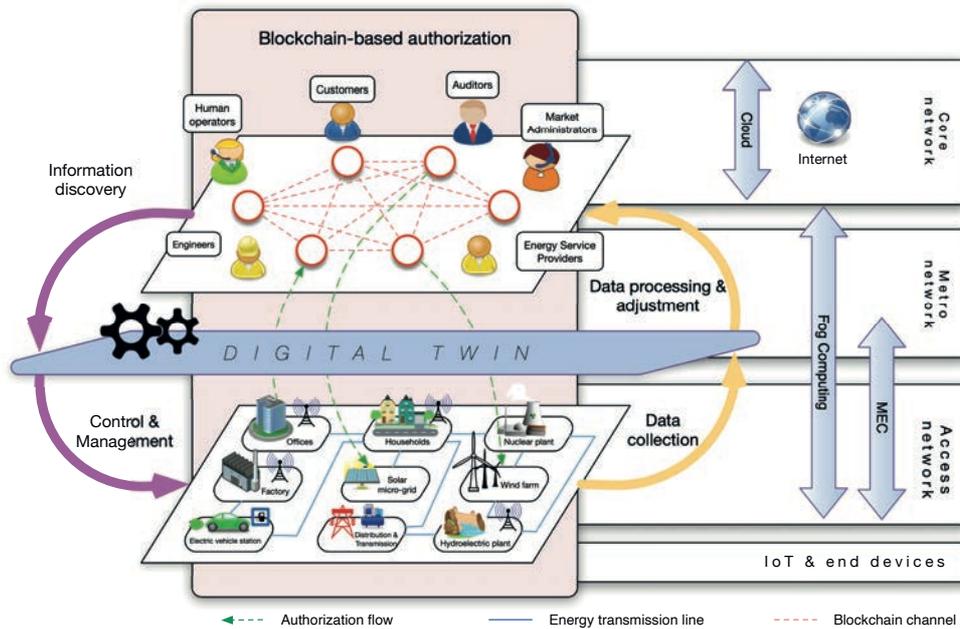


Figure 5.24: Smart Grid architecture for accommodating DTS in the long term

(e.g., energy readings, pricing data), host-based or network-related information (e.g., protocol used, commands issued, bandwidth). Together with cybersecurity analysis executed ad-hoc or leveraging external mechanisms (e.g., intrusion detection systems), this information helps to create a virtual representation of the assets in a certain region. This enables an accurate assessment of the current state of devices and the prediction of consequences derived from potential actions, approaching the concept of Digital Twin.

A Digital Twin (DT) can be defined as a representation of a physical asset in virtual space, enabled by a synchronized data acquisition about its structure, functionality and behavior. By analyzing and simulating virtual states of such entity, it is possible to undertake real-time monitoring and predictions, optimize processes and improve decision making [321]. Therefore, they emerge as a solution to guide access control, by coordinating all security services within the SG network in a holistic manner. In particular, these can transparently implement the PDP components in charge of autonomously apply authorization at a regional or local level, carrying out the context-awareness processes.

In the long-term roadmap of the Smart Grid, the gradual integration of Artificial Intelligence (AI) solutions (like those implemented by Digital Twins) will be decisive for the ultimate goal of having an autonomous, fully decentralized power grid. This is consistent with recent industrial views that suggest that the landscape of DT evolution will fulfill a three-stage process: from mere monitoring systems with limited analysis capabilities nowadays, going through semantic platforms featuring prediction and optimisation over the next few years, until the future implementation of fully semantic, self-learning, socio-technical platforms [322].

Figure 5.24 shows the characterization of DTs in the upcoming SG architecture. This hyper-connected and totally decentralized model assumes that the barrier between energy generation, transmission and distribution assets are blurred, and all processes coexist with micro-grids and EV infrastructures. Based on any application scenario, these resources are collaboratively used and the information is compartmentalized and securely accessible by the corresponding stakeholders, who have flexible control over the legislation, energy management and data acquisition. This is achieved with techniques based on Network Function Virtualization (NFV) and Software Defined Networking (SDN), with little or no change to the physical infrastructure: firstly, a 5G-based communication would allow the instantaneous connection of millions of IoT devices with a distributed peer-to-peer (P2P) automation, aided by Mobile Edge Computing (MEC) technologies. These bring the cloud computation closer to the proximity of users (i.e., the edge of the network) to carry out data analytics, in orchestration with fog computing to deploy scalable services across multiple domains. In a higher layer of abstraction over these physical devices, there would be a blockchain-based authorization system to manipulate data while ensuring its security at rest.

At this point, DLT solutions (such as a blockchain) can vouch for the data ownership and provenance between all partners within a federated Smart Grid. By integrating a blockchain network, access registers can be securely analyzed by external auditors to submit potential policy updates to devices and components involved, favouring the creation of access control schemes governed by Smart Contracts [323]. This way, these structures would be able to handle the access to information and resource trading in communities, thereby avoiding the need for additional and coupled PDP computation nodes.

Transversely to the end devices and the blockchain infrastructure deployed over the future grid, the presence of DTs must be holistic to achieve a symbiosis between physical assets and their virtualized entities. This means the authorization and the energy management processes must be integrated around the DT agents to implement a fully distributed automation. This way, they play the proactive role of controlling resources over the grid, compared to the passive behavior (i.e., monitoring) presented in Section 5.4.2. This functionality is enabled by orchestrating MEC, fog and cloud services at multiple architecture layers, which enables sensing of the physical and to have full interaction with the blockchain and the production line. The functionality loop between both worlds is provided by the data that connects them, so that the DT agents act as transparent but operational proxies in this duality, as represented in Figure 5.24. It comprises four phases that are executed permanently in high-frequency intervals:

1. **Data collection:** energy usage data and control information is retrieved in the proximity of IoT and end devices, leveraging the B5G infrastructure to carry out context-awareness procedures.

2. **Data processing and adjustment:** as data is aggregated, further analysis and detection is performed, to subsequently store such information in the ledger and execute additional maintenance tasks to inform potentially affected stakeholders.
3. **Information discovery:** the DT subscribes to events on the blockchain that are related to its monitoring area (e.g., pricing information, demand response) in order to accelerate decision-making and anticipate potential security issues that may render changes in the access to resources.
4. **Control and management:** as aforementioned, the DT agents that are hierarchically spread over the SG infrastructure have full autonomy to manage its corresponding assets, without the need of a vertical and centralized control.

Aside from sensors information and control commands, authorization requests and responses pass through the intermediate DT agents located in the edge. These submit transactions to the ledger and relay the outputs back to the field devices using 5G communication, based on the existing policy and the cybersecurity state assessed by the agents (e.g., using the Opinion Dynamics traceability model). Likewise, they can propose amendments to the access control scheme based on repetitive behaviors and past perceptions.

The access control and authorization systems will acquire a greater influence from artificial intelligence in the new DT prototypes for the Smart Grid. The provisioning of traditional policy schemes in industrial sectors requires an initial static procedure to analyse the regulations applied, engineer the roles involved, establish permissions and define rules for accessing resources and performing actions, considering precise constraints and relationships between assets. At the same time, these rules should be consistently declared to avoid conflicts, using an interoperable policy language such as XACML (eXtensible Access Control Markup Language) [324]. However, such mechanisms will have to face an unsteady environment where a huge set of actors fluctuate and the information flow is massive. In consequence, policies will have to be continuously assessed for the entire set of domains in order to readjust them in real time depending on a wide range of social, economic, and security conditions to guarantee the continuity of the network [325]. Additionally, although decentralised authorization systems are more flexible than centralized decision points in terms of efficiency, they are harder to manage.

Therefore, the administration of complex authorization systems is expected to progress towards more automated processes with scarce manual intervention, as Figure 5.25 shows. In this trend, we can classify the use of AI for intelligent authorization into two research lines:

- **Automatic policy alteration.** The aim is to gain insight from previous access requests and the overall behavior of the system, in order to refine existing rules. Data mining and classification algorithms are useful to identify discrepancies in policy specifications and infer new properties. Also, such evaluation can be combined with time-constrained delegation

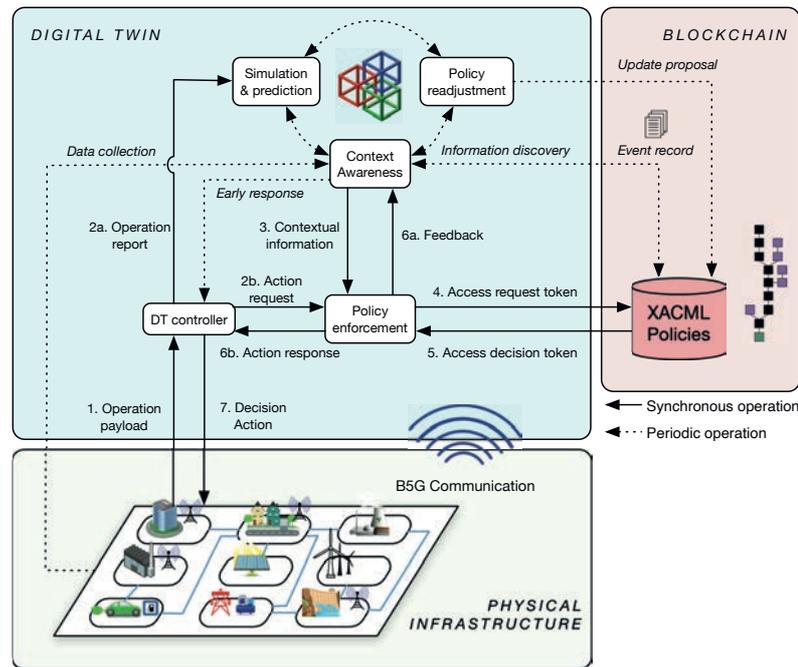


Figure 5.25: DT authorization workflow in the long term

models and domain-specific rules to derive authorizations in unforeseen scenarios [326]. Altogether, they can help to automate conflict resolution and role assignment, as well as to support implicit authorizations (i.e., accesses that are not explicitly specified or granted) [327]. Simultaneously, the complexity of the explicit authorization set is reduced.

The logic behind the analysis and improvement of security policies could ideally be implemented through smart contracts in the architecture shown in Figure 5.24. This way, the DT agents distributed over the architecture would be in charge of auditing the policy correctness, so that they would be able to submit transactions to propose upgrades to certain access control functions, which would be then approved by the consortium after being contrasted with other peer agents concerned.

- **Rule learning.** In this case, algorithms are trained to learn from data and infer policies rules from scratch. The most significant solutions here are about reinforcement learning. The traffic and events generated during the standard operation of the grid are studied to identify target resources and infer trust relationships between users and assets, based on the anomalies encountered [325]. Their counter-side is that the system is exposed to potential security threats, as the learning process takes place progressively while accesses are made and optimal policy rules are barely applied. As such, these solutions are more appropriate in later stages of the authorization life cycle, when a base authorization model is enforced. Other alternatives to guide the reinforcement learning consists in probabilistic policy reuse,

which balances among the application of the dynamically learnt policy, the exploration of random actions and the use of past policies.

Due to the differences between automatic alteration and rule learning, a dynamic authorization system that is collaboratively maintained and upgraded by a decentralized architecture of DTs must find a balance between these two approaches. Generally, the most common rationale will be to apply rule learning over a minimal set of policy regulations defined in the organization, to subsequently polish them with automatic alteration methods. Therefore, this process must be coupled with context awareness and auditing procedures to feedback the learning mechanisms and be fully integrated with the DT functionality loop described before.

## Chapter 6

# Verification and Validation

After showing the benefits of the APT traceability framework and applying the enabling solutions to some Industry 4.0 scenarios, we now have to validate all these findings. In this chapter, we perform the verification and validation of the framework defined, the correlation algorithms and the response techniques developed. Both processes are different in their definition and goal. According to the PMBOK guide, an standard adopted by the IEEE, *validation* refers to the assurance that the system meets the needs of the customer and other identified stakeholders. On the other hand, *verification* concerns the evaluation of whether the system complies with a requirement or specification.

In our case, the verification is conducted using theoretical demonstrations to elaborate the correctness proof of every approach presented in relationship with the detection and traceability of threats. This includes the clustering-based detection (introduced in Section 4.3.3) and the Opinion Dynamics-based technique to perform the traceability of APTs (presented in Section 4.6) and ensure the survivability of the network (Section 5.1), by means of induction. Additionally, the response against APTs using the information provided by the Opinion Dynamics solutions is assessed through game theory. On the other hand, we also validate our detection approach from a practical point of view, by implementing a proof of concept of this approach in a real testbed, that integrates several kinds of industrial devices and protocols.

### 6.1 Clustering-based Detection Approach through Induction

This section presents the correctness proof of the clustering-based detection presented in Section 4.3.3, both the location and accumulative approach. This problem is solved when these conditions are met:

1. The attacker is able to find an IT/OT device to compromise within the infrastructure.

2. The traceability solution is able to identify an affected node, thanks to the clustering mechanism and fulfilling the  $O_1$  output of the APT traceability framework, defined in Section 4.2.2.
3. The detection can continuously track the evolution of the APT and properly finish in a finite time (termination condition), complying with the  $O_2$  and  $O_3$  outputs.

The first requirement is satisfied under the assumption that the attacker breaks into the network and then moves throughout the topology following a finite path, according to the model explained in Section 4.1.3. An APT was defined as at least one sequence of attack stages against the network defined by  $G(V, E)$ . If we study each of these traces independently, and based on the distribution of  $G$ , the attacker can either *compromise* the current node  $v_i$  in the chain (as well as performing a *data exfiltration or destruction*) or propagate to another  $v_j \in V$ , whose graph is connected by the means of firewalls, according to the interconnection methodology illustrated in [82] and summarized in Section 4.1.1.

As for the second requirement, it is met with the correlation of anomalies generated by agents in each attack phase. As presented with the attacker model, the value of these anomalies are determined in a probabilistic manner, depending on two possible causes: (1) the severity of the attack suffered and the criticality of the concerned resource; or (2) an indirect effect caused by another attack in the vicinity of the monitored node. Either way, the  $O_1$  correlation helps to actually determine whether the attack has been effectively perpetrated against that node, or it belongs to another APT stage in its surroundings. This information is deduced from the combination of  $I_2$  (the contextual information) together with these anomalies (i.e.,  $I_1$ ), by using K-means to group these nodes and associate them with actual attacks.

We can easily demonstrate the third requirement (i.e., the termination of the approach) through induction. To do so, we specify the initial and final conditions as well as the base case:

**Precondition:** we assume the attacker models an APT against the network defined by graph  $G(V, E)$  where  $V \neq \emptyset$ , following the behaviour explained in Algorithm 3. On the other hand, the detection solution based on clustering can firstly sense the individual anomalies in every distributed agent, hence computing  $I_1$  and  $I_2$ .

**Postcondition:** the attacker reaches at least one node in  $G(V, E)$  and continues to execute all stages until  $attackSet = \emptyset$  in Algorithm 3. Over these steps, it is possible to visualize the threat evolution across the infrastructure, following the procedure described in Algorithm 2 in the case of accumulative clustering, and running K-means with both  $I_1$  and spatial information, in the case of location-based clustering.

**Case 1:** the adversary intrudes the network and takes control of the first node  $v_i \in V$ , and both clustering approaches cope with the scenario of grouping healthy nodes apart from

the attacked node. This is calculated by the K-means algorithm within a finite time, by iteratively assigning data items to clusters and recomputing the centroids.

**Case 2:** the adversary propagates from a device node  $v_i$  to another  $v_j$ , so that there exist  $(v_i, v_j) \in E$ . In this case, the correlation with K-means aims to group both affected nodes within the same cluster, which can be visualized graphically. As explained before, this is influenced by the attack notoriety and the closeness in the anomalies sensed by their respective agents (i.e., the threshold  $\epsilon$  in Algorithm 2), as well as extra information given by  $I_2$ .

**Induction:** if we assume the presence of  $k \geq 1$  APTs in the network, each one will consider Case 1 at the beginning and will separately consider Case 2 until  $attackSet = \emptyset$  for all  $k$ , ensuring the traceability of the threat and complying with the postcondition. Eventually, these APTs could affect the same subset of related nodes in  $G$ , which is addressed by the K-means to correlate the distribution of anomalies (again, attempting to distinguish between attacked nodes and devices that may sense side effects), in a finite time.

This way, we demonstrate the validity of the approach, since it finishes and it is able to trace the threats accordingly.

## 6.2 Opinion Dynamics-based Traceability through Induction

This section presents the correctness proof of the consensus-based detection and traceability problem for APTs, which was introduced and explained in Section 4.6. This problem is solved when the following conditions are met:

1. The attacker is able to find an IT/OT device in the system and attack it.
2. The detection system is able to trace the threat, thanks in part to the consensus (detection and traceability).
3. The system is able to properly finish in a finite time (termination).
4. The algorithm is capable of terminating and providing advanced detection at any moment (validity).

The first requirement is satisfied because we assume that the attacker is capable of (i) declaring the chain of attacks in advance, such as scanning, lateral movement, exfiltration or destruction (see Section 4.1.3), and (ii) identifying kinds of devices (e.g., IT/OT nodes and firewalls) by their functionalities. The modus operandi of the attacker is systematic except when the attacker needs to make a specific lateral movement, either through the selection of a new random neighbor node within the network or the selection of the neighbor with the highest betweenness. To comply with

the predefined attack patterns, the attacker firstly needs to identify the first target node, which generally belongs to IT network – evidently, this characteristic depends on the type of attacker (insider or outsider) and their skills. If the attacker is an outsider, his/her goal is to find a  $v_{IT_i} \in V_{IT}$  in order to penetrate by itself within the system, and to advance until reaching those nodes serving as firewalls such that  $v_{FW_i} \in V_{FW}$ . Once a  $v_{FW_i}$  is finally reached, the attacker tries to gain access in the operative network to compromise the most critical devices, i.e.,  $v_{OT_i} \in V_{OT}$ . If the attacker is an outsider, the compromises relies, in this case, on the pre-established APT threat chain; i.e., on *attackSet*.

The second requirement is also found due to the software prevention agents,  $a_i \in A$ , integrated as part of  $v_{IT_i}$ ,  $v_{FW_i}$  and  $v_{OT_i}$  of  $G(V, E)$ . These agents present capacities to detect anomalies and trace the intrusive presence by means of opinion dynamic parameters, the values of the which are attenuated according to time and aggressiveness of the threat (what we defined as the decay factor in Section 4.6). This attenuation, dependent on  $\Phi_i$ , does not means to completely forget an incident in past. But rather, in remembering the most significant aftermaths of the previous attacks in order to show the advance of the threat in real time, and therefore its traceability.

Through induction we demonstrate the third requirement, corresponding to termination of the approach. To do this, we specify the initial and final conditions together with the base case. Namely:

**Precondition:** by assumptions, we assume that the adversary is an advanced expert with skills to reach the IT-OT communication channels belonging to  $G(V, E)$ . However, this capacity depends on the set *attackSet* defined in Algorithm 3, which defines threat chain such that *attackSet*  $\neq \emptyset$ .

**Postcondition:** (i) the adversary reaches the network  $G(V, E)$  and compromises at least a node in  $V$  such that *attackSet* =  $\emptyset$  after the loop in Algorithm 3. And (ii) the system successful detects the threat such that  $\delta > 0$  and marks the traceability according to the real consensus state of  $G(V, E)$ , registered in the array vector  $x$ .

**Case 1:** *attackSet*  $\neq \emptyset$ , but  $| \textit{attackSet} | = 1$ . In this case, the attacker needs to launch the unique attack defined in *attackSet*. As mentioned, if the attack does not imply a lateral movement, the success of the threat is concentrated on just one node in  $V$ , since the following iteration of the loop implies that *attackSet*  $\leftarrow \textit{attackSet} \setminus \textit{attack}$ , and therefore *attackSet* =  $\emptyset$ . To the contrary, if the attack entails a lateral movement, then the attacker has to select a new neighbor node, either from a random or target point of view.

Any attack in  $V$  means an impact on the attacked node with a significant influence in its opinion dynamic (i.e.,  $x(\textit{attackednode})$ ). If, in addition, the decay factor is activated, the system weakens, but does not delete, the aggressiveness of the threat to stress the current trace of threat over the time. This computation is possible through  $\Phi_i$  in Algorithm 4. Once

$x$  is updated, the system computes the  $\delta$  value taking into account the weighted average of the Opinion Dynamics of the entire system (see Algorithm 5).

**Induction:** if we assume that we are in step  $k$  ( $k \geq 1$ ) of the loop where  $attackSet \neq \emptyset$ , then **Case 1** is going to be considered each time. When  $k = |attackSet|$ , the system computes **Case 1** and ends the detection algorithm with  $\delta > 0$  since  $attackSet = \emptyset$ , showing the traceability of the threat through  $x$  and complying with the postcondition.

Finally, the latter requirement is also satisfied since the algorithm finalizes and detects the threat through Opinion Dynamics (either individual or collective), and shows the traceability of the threat over the time.

### 6.3 Opinion Dynamics-based Survivability through Induction

The correctness proof of the message recovery problem presented in Section 5.1 is solved when the following requirements are satisfied: (1) the ratio of lost messages when facing an APT attack decreases when using the redundant topology; (2) the algorithm that crafts the set of redundant edges and sends the messages along the network is able to properly finish in a finite time (therefore arriving to the termination of the algorithm).

We can show the termination of the algorithm through induction, where we first define the initial and final conditions, and the base cases.

**Precondition.** We assume that the network described by  $G(V, E)$  is threatened by one or more attacks, probably causing the removal of available routes from the sender to the destination. In other words, there exists a share  $s$  belonging to a message  $m$  from sender  $v_1$  whose recipient is  $v_r$  for which it is not possible to find a sequence of vertices  $v_1, v_2, \dots, v_r$  such that  $(v_i, v_{i+1}) \in E, \forall i \in 1, \dots, r$ .

**Postcondition.** given the aforementioned message share and redundant network  $G'(V, E')$ , there exists a sequence of vertices  $v_1, v_2, \dots, v_r$  such that  $(v_i, v_{i+1}) \in E', \forall i \in 1, \dots, r$ . The availability of additional edges in  $E'$  is subject to the redundancy strategy selected. Either way, the new route is located by a pathfinding algorithm like *BFS* or *Dijkstra*.

**Case 1.** We have a message  $m$  that is divided into  $n$  shares, such that  $m = \{s_1, s_2, \dots, s_n\}$ . In the first step, share  $s_1$  is sent to vertex  $v_2$  through  $(v_1, v_2) \in E$ .

**Case 2.** In an intermediate step of the path from sender to destination, the share  $s_1$  traverses the node  $v_l$ , and the pathfinding algorithm is evaluated to check the availability of a route. According to Algorithm 8, three scenarios can be distinguished at this point:

-Recovery solution: it takes place when the destination is reachable only through the redundant topology  $G'$ , that is, there exists a route  $v_l, v_{l+1}, \dots, v_r$  where  $(v_l, v_{l+1}) \in E'$  but  $(v_l, v_{l+1}) \notin E$ .

-Privacy solution: it occurs when multiple routes are available to reach the recipient of the share, using either the original or the redundant topology. Namely, there exists, at least, a route  $v_l, v_m, \dots, v_r$  where  $(v_l, v_m) \in E$  and another one  $v_l, v_{l+1}, \dots, v_r$  such that  $(v_l, v_{l+1}) \in E'$  and  $(v_l, v_{l+1}) \notin E$ . In this case, the share hops to  $v_m$  or  $v_l$  arbitrarily, with the aim of making the route as confusing as possible, thereby dodging potentially compromised nodes over which the attacker expects the traffic to flow. Note that the network may experience some delays when delivering such shares (due to extra hops to reach the recipient), which could be the subject of further research. However, since we are considering a critical scenario, we prioritize availability rather than performance.

-Share loss: in the worst-case scenario, the redundant edges are not sufficient to find a path from  $v_l$  to  $v_r$  and the original path is no longer available due to the APT. In these circumstances, the share is lost and the algorithm terminates. Note, however, that the secret sharing scheme is resistant to share losses with a given threshold, so the rest of shares  $s_i$  with  $i \neq 1$  can still rebuild the message  $m$ . This depends on the  $n$  and  $k$  parameters: specifically, the message  $m$  successfully reaches its recipient with a probability  $\frac{k}{n}$ . In this regard, we must stress that the choice of  $n$  is based on the severity indicator  $\mu$ , as explained in Section 5.1.2.

**Induction** finally, after a finite number of steps where the different subcases of Case 2 have been applied (except for a secret loss), the node  $v_j$  before the last in the sequence holds the share  $s_1$  and there exists an edge  $(v_j, v_r)$ . The share is finally delivered and Algorithm 8 terminates, satisfying the postcondition of saving a portion of the messages from getting lost, ensuring the validity of our algorithm.

We can also give a brief analysis of the computational complexity of the response algorithm, which must be performed in two ways: for the secret sharing scheme and for the subsequent delivery of shares over the network by using a pathfinding algorithm. As for the former, processing a given message takes  $n$  steps, as many as the number of shares it has been split into (determined by  $\mu$ ), having  $O(n)$  complexity. With respect to the communication mechanism, the complexity must firstly consider the overhead invested by the pathfinding method, which in the case of BFS is  $O(n + e)$ , where  $n \approx |V|$  and  $e \approx |E|$ . Secondly, it also implies the complexity associated with the share delivery along the graph. Considering the worst-case scenario of the longest route, such a transmission has a cost of  $O(n - 1 + e)$ , since the share has to traverse all edges and every node in the network but the sender.

## 6.4 Opinion Dynamics-based Response through Game Theory

Among the novel mechanisms introduced in Chapter 4, Opinion Dynamics stands out as a multi-agent collaborative system that enables the traceability of the attack throughout its entire



life cycle, by means of a distributed anomaly correlation. In this section, we propose a theoretical but realistic scenario to prove the effectiveness of that approach under different types of attack model, using concepts supported by the structural controllability field [163] and game theory [328]. For that goal, we develop TI&TO, a two-player game where attacker and defender compete for the control of the resources within a modern industrial architecture. Both players have their own movements and associated scores, according to the behavior of an APT and a detection system based on Opinion Dynamics, respectively. This game is ultimately run in different simulations that aim to show the algorithm capabilities, while also suggesting the optimal configuration of the technique in conjunction with other defense solutions. Therefore, we can summarize our contributions as:

- Formal definition of the TI&TO game, specifying the game board, each player's goal and the score rules.
- Design of an attacker model in form of a set of stages that flexibly represents the phases of an APT, to represent the movements of the attacker, which are subject to a determined score.
- Design of a defender model based on the use of Opinion Dynamics and response techniques (i.e., local detection, redundant links, honeypots) to reduce the impact of the APT within the network, which also implies an associated score in the game.
- Experiments carried out to validate the algorithm and recommend the configuration of the defender that returns the best result.

In the context of industrial networks defense, researchers have been extensively exploring the applicability of game theory [328]. In these networks, it is common to cope with many levels of criticality, different network sizes, interconnectivity and access control policies. Therefore, decisions in terms of security frequently fluctuate, which is harder in Industry 4.0 scenarios, where many heterogeneous devices interact with each other and organizations exchange information using the cloud, fog computing or DLT structures. In this sense, game theory offers the capability of analyzing hundreds of scenarios, thereby enhancing the decision making. At the same time, it also allows to validate the effectiveness of a given technique (e.g., Opinion Dynamics in our case) if we analyze different strategies of use for all the scenarios examined.

Based on the information that each player has, there are different types of games: on the one hand, in a *perfect information* game both players are aware of the actions taken by their adversary at all times; on the other hand, a *complete information* game assumes that every player always knows the strategy and payoffs of the opponent. As explained further, the approach presented here (TI&TO) represents a two-player game with imperfect and incomplete information, since no player (i.e., attacker and defender) knows the location of the adversary within the network topology or his/her score. According to a second level of classification, this game can be considered

as dynamic and stochastic, as both players take their actions based on the state of the network and being exposed to events that affect them in a probabilistic way.

There are multiple researches in the literature that fall under these classifications. Concerning complete perfect information games, Lye et al. [329] proposes a two-player game that simulates the security of a network composed by four nodes that can be in 18 potential states, on which both players can take up to 3 actions, that are observable at all times by the opponent. With respect to complete imperfect information games, Nguyen et al. [330] propose ‘fictitious play (FP)’, a game that considers the network security as a sequence of nonzero-sum games where both players cannot make perfect observations of the adversary’s previous actions. Also, Patcha et al [331] propose an incomplete perfect information approach, for the detection of intrusions in mobile ad-hoc networks. Whereas the attacker’s objective is to send a malicious message and compromise a target node, the defender tries to detect it using a host-based IDS. Another related work based on imperfect information is [332], where van Dijk et al propose a simple game where two players compete for the stealthy control of a resource without knowing the actual identity of the owner until a player actually moves.

Many of these solutions have been successfully applied to the detection of threats. However, most of the models are based on either static games or dealing with perfect and complete information, aiming to find an optimal strategy when a steady state of the game is reached (being the Nash equilibrium the most famous one) [328]. In contrast, a real control system faces a dynamic interaction game with incomplete and imperfect information about the attacker, and the proposed models of this category do not specify a realistic scenario with an extensive attack model [332] [333]. This lays the base and inspiration for the design and implementation of our proposed scheme. With TI&TO, we aim to get insight about how to effectively implement and configure a defense strategy based on the use of Opinion Dynamics, under such stochastic conditions.

### 6.4.1 Proposed Network Architecture

As defined in next subsection, TI&TO focuses on a game where both attacker and defender fight for the control of an infrastructure. The attacker tries to break into the network in a stealthy way by taking over as many nodes as to complete the predefined kill chain of a specific APT. With respect to the defender, he/she must recover those nodes until he/she completely eradicates the threat from the network. Thus, this network infrastructure plays the role of the game board, and must be designed realistically as to represent the topology of a modern industrial ecosystem.

For this reason, the network used in the game embodies cyber-physical resources of different nature, ranging from operational devices (e.g., sensors/actuators, PLCs, SCADA systems, etc.) to information technology devices from the managerial point of view (e.g., customer-end systems). Following the methodology of Section 4.1, the board will be an infrastructure composed by two sections with the same number of nodes: OT and IT, connected via firewalls to secure the traffic. As already stated, the network is represented with graph  $G(V, E)$ , so that  $V$  refers to

the nodes connected with each other based on links contained in the  $E$  set. Thus, OT and IT sections are represented with  $G(V_{OT}, E_{OT})$  and  $G(V_{IT}, E_{IT})$ , respectively (having  $V = V_{IT} \cup V_{OT}$  and  $E = E_{IT} \cup E_{OT}$ ). Likewise, both sections are randomly generated following a different network distribution to simulate different infrastructure setups. Whereas the IT section follows a small-world network distribution,  $G(V_{OT}, E_{OT})$  is based on a power-law distribution of type  $y \propto x^{-\alpha}$ .

Once generated, both sections are connected by means of a set of intermediate firewalls  $V_{FW}$ , so that  $V = V_{IT} \cup V_{OT} \cup V_{FW}$ , as specified in Section 4.1.1. In the IT section, we want devices to be able to access the OT section, since they are computationally capable nodes that commonly control the production chain from the corporate network. This means that all nodes in  $V_{IT}$  are connected to  $V_{FW}$ . However, on the OT side, only SCADA systems and other high-level servers can access external networks, whereas the majority of them are sensors, PLCs and devices with a restricted functionality. Consequently, the connected nodes will be those that have a maximum connectivity (i.e., dominance in graph theory) within the power-law distribution network of the OT section, given the concepts of structural controllability introduced in Section 4.1.1. Therefore, for our concerned network infrastructure, the PDS of the OT section will be connected to the firewalls that also connect to the IT nodes. In our simulations, we consider that 5% of the total number of nodes in  $V$  are firewalls, to restrict the traffic between both sections in a realistic way.

In order to characterize the types of nodes within the architecture and enrich the network model, it is also necessary to define some related concepts that will be useful to understand the game dynamics:

**Criticality of nodes.** We define the criticality of a resource as the risk subject to that type of device within the organization, and determines the impact of a given threat if the attack is perpetrated at that point. For example, the criticality of a sensor is negligible compared to that of the SCADA system, which implies dramatic consequences on the infrastructure in the event it is disrupted. Likewise, resources in the OT section are also deemed as more critical than the IT ones to ensure the continuity of the production chain. This will be also used by the defender to assess which nodes should be healed in order to minimize the impact of an APT.

We formally define this concept taking into account the graph  $G(V, E)$  introduced before. Firstly, let  $CRIT : V \mapsto \mathbb{R}(0, 1)$  be a function that assigns a criticality degree to all nodes of the network. To distinguish which devices present a higher hierarchy within the topology, we additionally leverage the concept of DS and PDS introduced in Section 4.1.1. At the same time, since the OT section is considered as especially critical, its devices will have to be associated with a higher value. As a result, we define  $\Psi$  as an *ordered set of criticality values of size  $d$* , where  $\Psi = \psi_1, \dots, \psi_d$  and  $\psi_i = [0, 1]$ , such that  $\forall \psi_i, \psi_i < \psi_{i+1}$ .

Once  $\Psi$  is defined, we can create a model that maps every element of the network (i.e., its nodes) to the elements of  $\Psi$ . This model, where  $d = 6$  and  $\Psi = \psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6$  to consider



$V_{IT} - DS_{IT} - PDS_{IT}$	$\psi_1$
$V_{OT} - DS_{OT} - PDS_{OT}$	$\psi_2$
$DS_{IT}$	$\psi_3$
$DS_{OT}$	$\psi_4$
$PDS_{IT}$	$\psi_5$
$PDS_{OT} \cup FW$	$\psi_6$

Table 6.1: Map of  $V$  to  $\Psi$ 

all elements of both network sections (i.e., the OT and IT section, including its nodes and the DS and PDS subsets), is likewise described in Table 6.1.

**Vulnerability of nodes.** Besides the criticality, the concept of vulnerability involves the ease of a node to be compromised by the attacker. In this case, we will assume that this value is opposed to the criticality, in the sense that field devices will be commonly equipped with lower security protection measures, whereas high-level systems that control the industrial process will embody advanced security services. Correspondingly, we can define  $VULN : V \mapsto \mathbb{R}(0, 1)$  as the function that assigns a vulnerability degree to all nodes of the network. In the same way as criticality,  $\Upsilon$  is an ordered set that represents the vulnerability of each node type, where  $\Upsilon = v_1, \dots, v_d$  and  $v_i = 1 - \psi_i$ . The particular instantiation of these values for the simulations is carried out when the network represented by  $G(V, E)$  is created, as explained in the experiments section.

**Redundancy of links.** In order for the OT subnetwork to be resilient against DoS attacks located on their links, and due to the criticality of its resources, we also consider that this section presents redundancy on its edges. This is a solution that was also proposed in Section 5.1 as a response technique to enable the reachability of messages across the network. In our case, with the use of auxiliary edges in  $E$  (referred to as  $E_R$ , so that  $E_R \subset E$ ), we ensure that the detection algorithm exchanges the opinion among agents even when some links are down as a consequence of an APT. This may occur in the game when the attacker attempts the defender to lose track of the anomalies in the affected nodes. This way, all nodes in  $V_{OT}$  count on an additional channel that interconnects them with another node, based on the strategy explained in [334]. It is worthy to note that these redundant edges are just logical connections that only serve to transfer the anomaly values between agents.

Based on these principles, Figure 6.1 conceptually shows an example of network topology, together with the integration of the Opinion Dynamics correlator. In the diagram, the redundant edges in the OT section are represented with dashed lines.

### 6.4.2 Rules and Scoring System

We now describe the game dynamics for both players and how each of their movements is measured in quantitative terms. Since the final objective of this research is to assess the effectiveness of

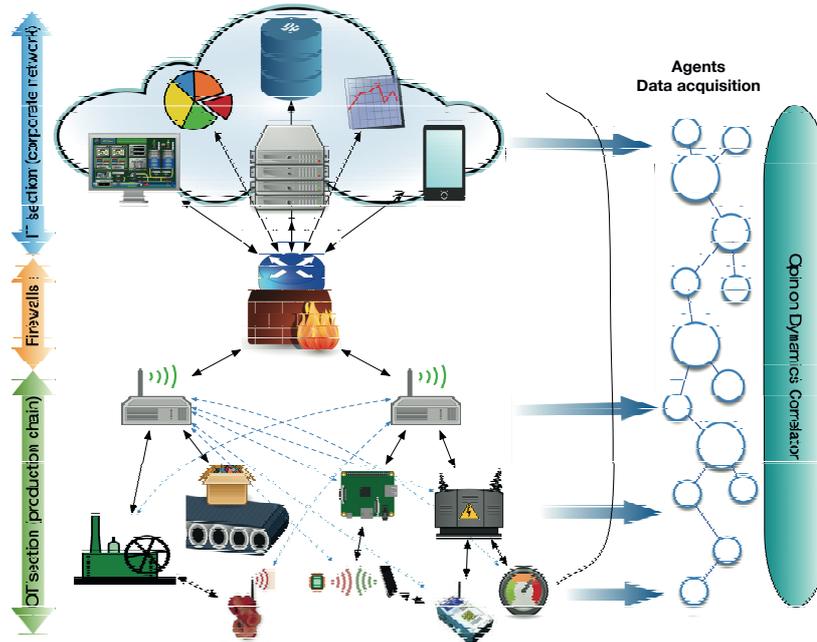


Figure 6.1: Example of network topology used in TI&amp;TO

the Opinion Dynamics, we aim to analyze the best behavior of the defender for a realistic attack model. Therefore, it becomes necessary to utilize a formal representation of the results while following a fair methodology for both players, which have equivalent costs and rewards assigned to their movements in the game.

We start by defining TI&TO in an informal way. As introduced before, both compete for the control of the game board. The base of the scoring system works as follows: *whereas the attacker earns points as it spreads the threat across the infrastructure, the defender increases the score when those infected nodes are recovered*. However, this is just the number of points scored, which serves as a reference of the throughput achieved by each player. There is a termination condition that regulates who wins a given game: *as for the attacker, the game is over when he/she manages to successfully complete all the phases of the APT kill chain*. Concerning the defender, *the victory is achieved when all nodes infected by the adversary return to their originally uncompromised state*. In the following, we give a formal definition of all the elements involved in TI&TO and the notation used along this manuscript:

**Players.** There are two players: the attacker and the defender. For simplicity, they are denoted by  $A$  and  $D$ , respectively.

**Time.** In our approach, time is split into discrete ticks for the interest of the analysis. The game begins at time  $t = 0$  and continues indefinitely as  $t \rightarrow \infty$ . At a given  $t$ ,  $A$  and then  $D$  has a turn to play. They act sequentially adopting a Stackelberg game [335], where the attacker is the leader and the defender acts depending on the resulting state of the board.

**Movement.** It is performed by  $A$  or  $D$  and changes the board at time  $t$  according to their respective attack and defense models. In brief, both players take actions to either take over healthy nodes of the network (in the case of the attacker) or heal a compromised node (by the defender). Therefore, every movement can alter the state of a node. It is denoted by  $M^p(t)$

**Node State.** It is a time-dependent variable  $N = N(t)$  that determines whether a node in  $V$  is compromised (i.e., the attacker has reached it) or remains safe from the APT. For a given node  $i$  (belonging to the IT or OT section),  $N_i(t)$  is equals to one if it is compromised at time  $t$ , and zero otherwise. We assume that  $\forall v \in V, N_v(0) = 0$ .

**Reward.** Every movement performed by  $A$  or  $D$  generates a reward depending on the ultimate goal that both of them chase, which determines the score. In this case,  $A$  receives one point when a new node is compromised, whereas  $D$  obtains the same reward once a previously compromised node has been successfully recovered. A reward for a player  $p$  at a time  $t$  is denoted by  $R^p(t)$ .

**Cost.** Besides a reward, every movement also implies a cost  $C$  for the player. This represents the fact that the attacker can exploit vulnerabilities that in turn may cause its detection, while the defender may stop the production chain to recover the security state of a critical resource. It is formalized with  $C^p(t)$ .

**Utility.** It is the total number of points scored by a player  $p$  at time  $t$ . It is calculated as the reward minus the cost of the movement made by  $p$ , which is denoted by  $U^p(t)$ . The overall goal for both players is to maximize the utility as  $t \rightarrow \infty$ , until the game is over.

**Strategy.** We define a strategy  $S$  for a player  $p$  as the sequence of movements  $M(t)$  along time for a given instance of game, represented by  $S^p = \{M^p(0), M^p(1), \dots, M^p(t)\}$ . As explained later on, this strategy changes as the game evolves: whereas the attacker seeks vulnerable nodes throughout the network while avoiding its detection, the defender follows an adaptive strategy based on the last movement of  $A$  (more specifically, on the new state of the affected nodes).

Although we consider the utility as a reference for the performance of both players in a given game instance, we define three different termination states:

( $TS_1$ ) **Attacker wins.** It is reached when he/she successfully completes all the movements of the strategy  $S^A$ , where  $S^A = \{M^A(0), M^A(1), \dots, M^A(n)\}$ . We assume there exists at least one last node  $v$  that is compromised, so that  $N_v(n) = 1$ .

( $TS_2$ ) **Defender wins.** It is accomplished when the defender manages to heal all nodes and hence eradicate the effect of the attacker over the entire network, before the succession



of movements in  $S^A$  are completed. In other words, for a given attacker strategy  $S^A = \{M^A(0), M^A(1), \dots, M^A(n)\}$ , there exists  $t' < n$  such that for all  $v \in V, N_v(t') = 0$ .

( $TS_3$ ) **Draw.** For the interest of the analysis, we define an additional third termination condition that occurs when the attacker completes the strategy  $S^A = \{M^A(0), M^A(1), \dots, M^A(n)\}$  but the defender also performs a last movement that ultimately heals all nodes. In this case, we have that for all  $v \in V, N_v(n) = 0$ . Even though this may be considered as an attacker win (since he/she succeeds in the disruption of resources), the defender still finds the trace to the threat in the end, which shows the accuracy of the detection technique going after the infection.

With this, the dynamics of the game and the basic rules have been presented. However, we have to describe the precise specification of the players' movements. While the intruder puts into practice a set of individual attack stages that represent an APT (i.e., a strategy of  $n$  movements), the defender leverages the Opinion Dynamics algorithm to flexibly adapt to the threat propagation over the network. In both cases, they can apply different actions to change the state of nodes and obtain a score based on different conditions.

### 6.4.3 Attack and Defense Models

As introduced before, we aim to find a formal representation of an APT for the attacker model. In this sense, the attacker model in TI&TO is inspired by the methodology of Section 4.1.3. After the extensive review of the most important APTs reported in recent years, we came to the conclusion that it is possible to specify one of these threats as a finite succession of attack stages perpetrated against an industrial control network defined by the graph  $G(V, E)$ , so that  $attackStages = \{attack\ stage_1, attack\ stage_2, \dots, attack\ stage_n\}$ . This way, each attack stage corresponds to a different movement performed by the attacker. In the following, we describe the different types of stages considered in the game and explain their effect on the board. Then, the reward and cost generated for this player are calculated. Lastly, the strategy creation is explained:

- **initialIntrusion**<sub>(IT,OT,FW)</sub>. After a phase of reconnaissance, the attacker breaks into the network through a 'patient zero'  $v_0 \in V$ , that can be a node from the IT or OT section. It is the first movement of the attacker ( $M^A(0)$ ), so that  $N_{v_0}(0) = 1$ .
- **LateralMovement**<sub>(IT,OT,FW)</sub>. Once a node  $v_i$  has been compromised, the adversary chooses a FW (if it is accessible), IT, or OT node  $v_j$  from the set  $neighbours(v_i)$  (i.e., those nodes for which there exists one edge  $e = (v_i, v_j)$  such that  $e \in E$ ). For the election of the node to take over, we assume that the attacker scans the network in the seek for the most vulnerable device (according to the  $VULN$  function). We assume  $A$  can compromise a node that has been previously healed by the defender, but its  $VULN$  value is then reduced by half.

- **LinkRemoval.** Once the attacker has perpetrated a lateral movement from  $v_i$  towards  $v_j$ , that communication channel can be disrupted to decoy the defender (and hence avoid the Opinion Dynamics detection). As a result, the defender cannot exchange the opinion of the agents assigned to  $v_i$  and  $v_j$ , since no anomaly information is transferred through that link, as explained in the next section.
- **Exfiltration of information and Destruction.** It represents the final movement of the attacker. The adversary destroys the node that has been previously compromised, after possibly extracting information that is sent to an external command&control network.

Each of these movements results in a different cost and reward for the attacker, who determines his or her utility after each turn of the game, so that the score can be compared with the defender. As for the reward, and aiming to hold the symmetry between both players, they will receive one point every time they gain control of a given node that previously belonged to the adversary. For the attacker, it means that there exists one node  $v \in V$  at a time  $t$  such that  $N_v(t-1) = 0$  and  $N_v(t) = 1$  after  $M^A(t)$ , resulting in  $R^A(t) = 1$ . For simplicity, we consider that all stages have the same reward.

With respect to the cost of every attack stage, we have to recall the Opinion Dynamics algorithm in relationship with the defender goals. We assume all the network resources are monitored by anomaly detection mechanisms, outputs of which are retrieved by a Opinion Dynamics correlation system. This allows the defender to potentially trace the movement of the attacker along the network, since the different attack stages will generate various security alerts that increase the probability of detection, which can be conceived as a cost. In Section 4.1.3, we proposed a taxonomy of detection probabilities in form of an ordered set associated with each attack stage. Following the same procedure, now we define  $\Theta$  as the ordered set of detection probabilities, where  $\Theta = \{\theta_1, \dots, \theta_n\}$  and  $\theta_i = [0, 1]$ , such that  $\forall \theta_i, \theta_i < \theta_{i+1}$ . This model, which is illustrated in Table 6.2, maps every attack stage to the elements of  $\Theta$  to represent their cost. There are multiple reasons behind this mapping, that are summarized as follows:

1. We assign the lowest level of detection probability ( $\theta_1$ ) only to the devices in the neighbourhood of the affected node in a lateral movement, since some discovery queries will normally raise subtle network alerts.

$initialIntrusion(v_0)$	$\theta_3$
$*LateralMovement_{IT,FW}(v_i \rightarrow v_j), neighbours(v_i)$	$\theta_4 \rightarrow \theta_2, \theta_1$
$*LateralMovement_{OT}(v_i \rightarrow v_j), neighbours(v_i)$	$\theta_5 \rightarrow \theta_2, \theta_1$
$*LinkRemoval(v_i \rightarrow v_j)$	$\theta_5 \rightarrow \theta_5$
$destruction(v_i)$	$\theta_6$

Table 6.2: Map of *attackStages* to  $\Theta$

2. The second lowest probability of detection ( $\theta_2$ ) is linked to the elements that are the target of a lateral movement, because these connections usually leverage stealthy techniques to go unnoticed.
3. An initial intrusion causes a mild detection probability  $\theta_3$ , since the attacker either makes use of zero-day vulnerabilities or social engineering techniques, which is a crucial stage for the attacker to be successful at breaking into the network through the ‘patient zero’.
4.  $\theta_4$  and  $\theta_5$  are assigned to devices (from the IT and OT section, respectively) causing the delivery of malware to establish a connection to an uncompromised node in a lateral movement. In specific, since the heterogeneity of traffic is lower and the criticality of the resources in that segment is greater, anomalies are likely to be detected when compared to the IT section. On the other hand,  $\theta_5$  is also assigned to the involved nodes in a link removal stage, since it is an evident anomaly sensed by both agents.
5. The highest probability of detection ( $\theta_6$ ) is assigned to the last stage of the APT, as it usually causes major disruption in the functionality of a device or the attacker manages to connect to an external network to exfiltrate information, which is easily detected.

The precise election of this taxonomy and quantitative instantiation of the  $\theta$  values is further explained in the experiments section.

As for the strategy applied for the attacker in TI&TO,  $S^A$  will vary depending on the state of surroundings nodes that are vulnerable at every time  $t$  of the game. The precise behavior to define the chain of attack stages is the following:  $S^A$  always starts with an *initialIntrusion*, which is randomly chosen from the IT or OT section (hence representing multiple kinds of APTs [227]). Then,  $A$  attempts to make a *LateralMovement<sub>FW</sub>* movement to compromise a firewall. This movement is straightforward on the IT section as every node is connected to them. However, in case of the OT section, the attacker needs to escalate over the hierarchy of nodes until reaching a PDS node and then the firewall, as explained in Section 6.4.1. Once there,  $A$  penetrates the other section, where we assume he/she must complete a minimum succession of  $\sigma = 3$  *LateralMovements* (choosing the most vulnerable nodes) before finally executing the *Destruction* of a resource. In that case, the game terminates complying with  $TS_1$  or  $TS_3$ , depending on the movements of  $D$ . In this sense, the defender can prevent this chain from completing if he/she detects the attacker and successfully eradicates the infection from all nodes (complying with  $TS_2$ ). In order for the attacker to avoid that situation, a *LinkRemoval* can be executed. In TI&TO,  $A$  makes this movement when the defender manages to heal  $h = 3$  nodes in a row, which represents the situation where  $D$  is close behind the attacker on the board, as explained in the next section.

This procedure to define the attacker strategy as the game evolves is formalized in Algorithm 12. Note that the attacker can always follow this chain of stages as long as he/she posses at least

one node. In case one is healed, another node is chosen and the APT continues. Otherwise, if the defender manages to heal all victim nodes, the game ends complying with  $TS_2$  or  $TS_3$ .

As discussed before, the ultimate goal of this section is the analysis of the Opinion Dynamics technique against the effects of a realistically-defined APT. As such, we assume that the set of movements that the defender can leverage is summarized in the execution of the algorithm at every turn of the game, followed by an optional node reparation, as described in Section 6.4.2. Therefore, the defender adopts a dynamic behavior which allows us to analyze the effectiveness of different protection strategies.

---

**Algorithm 12** Attacker strategy creation
 

---

**output:**  $S^A$  representing the attacker strategy  
**local:** Graph  $G(V, E)$  representing the network, where  $V = V_{IT} \cup V_{OT} \cup V_{FW}$ ,  $gameState = 0$  representing initial game state

```

 $S^A \leftarrow \{\}$ ,  $Victims \leftarrow \{\}$ ,  $numSteps \leftarrow 0$ 
 $attackedNode \leftarrow$  random node in  $V_{IT} \cup V_{OT}$ 
 $S^A \leftarrow S^A \cup initialIntrusion(attackedNode)$ ,  $Victims \leftarrow Victims \cup attackedNode$ 
while  $gameState == 0$  do
  if defender healed  $h$  nodes in a row and  $numSteps < \sigma$  then
     $S^A \leftarrow S^A \cup LinkRemoval$ 
  else if  $attackedNode$  is in first section attacked then
     $S^A \leftarrow S^A \cup LateralMovement_{FW}(nextAttackedNode)$ 
     $Victims \leftarrow Victims \cup nextAttackedNode$ 
     $attackedNode \leftarrow nextAttackedNode$ 
  else if  $attackedNode$  is in second section attacked and  $numSteps < \sigma$  then
     $S^A \leftarrow S^A \cup LateralMovement_{(IT,OT)}(nextAttackedNode)$ 
     $Victims \leftarrow Victims \cup nextAttackedNode$ 
     $attackedNode \leftarrow nextAttackedNode$ ,  $numSteps \leftarrow numSteps + 1$ 
  else
     $S^A \leftarrow S^A \cup Destruction(attackedNode)$ ,  $gameState \leftarrow TS_1$ 
  end if

  if defender healed  $attackedNode$  then
     $Victims \leftarrow Victims \setminus attackedNode$ ,  $numSteps \leftarrow 0$ 
    if  $Victims$  is empty then
      if  $gameState == TS_1$  then  $gameState == TS_3$ 
      else
         $gameState \leftarrow TS_2$ 
      end if
    else
       $attackedNode \leftarrow$  random node in  $Victims$ 
    end if
  end if
end while

```

---

We start with the basics. As mentioned in Section 6.4.2, the defender aims to locate the attacker position across the whole network, keeping track of the anomalies suffered and their persistence over each area of the network as the game evolves. This is enabled by the Opinion Dynamics traceability, as proposed in [300]. Thus, the status of the network is checked by the defender at each turn: then, the most affected node is selected and, based on the severity of the anomaly, he/she finally decides to heal the node. Depending on the accuracy of this action, the defender receives a determined utility. This process, which is henceforth referred to as ‘reparation’, is described in Algorithm 13. It is repeated successively in each turn of the defender, until all compromised nodes are repaired, complying with the defender-win condition (so that the complexity of the defensive approach is linear) or the attacker completes its set of attack stages. There are some aspects to point out here. Firstly, the defender can decide whether to repair the most affected node or stay idle during each turn, which depends on a predefined threshold. Namely, if the opinion given by the agent that monitors that node surpasses it, then the defender opts to heal it. After executing the experiments, and since Opinion Dynamics is calculated as a sum of weighted sum of opinions, this threshold is set to 0.5, which returns the best outcome for the defender.

On the other hand, the reward is one as long as the defender succeeds at healing a node that was in fact compromised; otherwise, the reward is zero. With respect to the cost, it is equivalent to the criticality of the node that is healed (regulated with the *CRIT* function of Section 6.4.1), in such a way that high-level resources are subject to a potential stop in the production chain and usually need a greater effort in terms of security.

---

**Algorithm 13** Reparation of nodes at time  $t$

---

**output:**  $U^D(t)$  representing the utility  
**local:** Graph  $G(V, E)$  representing the network, where  $V = V_{IT} \cup V_{OT} \cup V_{FW}$   
**input:**  $X$  representing the opinion vector of the network agents

```

candidateNode  $\leftarrow$  node in  $V$  with maximum  $x(t)$ 
OldNodeState  $\leftarrow$   $N_{candidateNode}(t)$ , healThreshold  $\leftarrow$  0.5
if  $x_{candidateNode} > healThreshold$  then
    REPAIRNODE(candidateNode)
end if
if OldNodeState == 1 then
     $N_{candidateNode}(t) \leftarrow 0$ ,  $R^D(t) \leftarrow 1$ 
else
     $N_{candidateNode}(t) \leftarrow 0$ ,  $R^D(t) \leftarrow 0$ 
end if
 $C^D(t) \leftarrow CRIT(candidateNode)$ ,  $U^D(t) \leftarrow R^D(t) - C^D(t)$ 

```

---

The reparation procedure is the main movement of the defender. However, this reparation strategy can also be influenced by three different configurations:

- **Local Opinion Dynamics.** In practice, a global correlation of the Opinion Dynamics agents in a synchronous way may not be feasible in a real industrial environment. Concretely, we aim to demonstrate that the execution of the aforementioned correlation, but considering a subset of nodes of the original network, is effective enough for the defender. Let  $G'(V', E')$  be the subgraph of  $G(V, E)$  so that  $V' \subset V$  and  $E' \subset E$ . This subgraph is built including a *candidateNode* and all its child nodes within graph  $G$  located at a distance of certain number of hops (in our tests, a distance of one or two hops will be used). The graph  $G'$  is used for the computation of the Opinion Dynamics, as usually performed in the original approach. The first election of *candidateNode* is established after  $M^A(0)$ , considering the highest anomaly measured by the agents over the network. Afterwards, the defender is able to locally compute the correlation and heal nodes in subsequent movements. Thus, at every turn, the *candidateNode* is updated to the node in  $V'$  with the greatest opinion, which implies moving the Opinion Dynamics detection zone.
- **Redundancy of links.** In Section 6.4.3, the link removal stage was introduced, that allows the attacker to potentially remove links from the topology that make the defender lose track of the threat position, by fooling the local Opinion Dynamics. At this point, we must recall the subset of redundant links  $E_R \subset E$  introduced in Section 6.4.1. These channels will be used by the defender whenever the attacker destroys a link in  $E$ , so that opinions will be transmitted using those links only in that case. Despite this may seem as an advantage for the defender, those links can randomly cover pairs of nodes that may not be affected by a link removal. Additionally, the disruption of a link from  $v_i$  to  $v_j$  in  $E'$  does not make  $v_j$  inaccessible for the local Opinion Dynamics at all times, since there could be a third node  $v_k$  covered by the defender that has another connection  $(v_k, v_j) \in E'$ .
- **Honeypots.** For the interest of the analysis, the defender lastly features the possibility of establishing honeypots. It implies modifying the network from the beginning to assign the role of honeypot to specific nodes, which will be randomly chosen in the simulations. These are used as a bait to lure the attacker to compromise them by exposing a higher degree of vulnerability (which was regulated with the *VULN* function of Section 6.4.1). If the attacker attempts to compromise it, then a higher anomaly will be generated by that agent, which would help the defender to rapidly find the position of the threat, eradicate the threat at a given turn  $t$  and hence update the area of the local Opinion Dynamics detection. For our tests, 5% of the total number of nodes have been considered as honeypots, which is a minimal value to show the effectiveness of this response technique.

Table 6.3 summarizes the set of movements eligible for each player, indicating their reward and cost. In the following, we run simulations with different configurations for the defender to assess the Opinion Dynamics detection technique.



Player	Movements	Reward	Cost
Attacker	Initial Intrusion	1	$\theta_3$
	Lateral Movement ( $v_i \rightarrow v_j$ )	1	$\theta_4$ or $\theta_5 + \theta_1 *  \text{neighbours}(v_i) $
	Link Removal ( $v_i \rightarrow v_j$ )	1	$2 * \theta_5$
	Destruction ( $v_i$ )	1	$\theta_6$
Defender	Node reparation ( $v_i$ )	1	$CRIT(v_i)$

Table 6.3: Summary of movements leveraged by attacker and defender

#### 6.4.4 Simulations and Results

Once both attacker and defender have been described, this section presents the results of playing games under different parameters of TI&TO. As explained, the aim of these experiments is to find the best strategy for the defender given an APT perpetrated by attacker.

In specific, four test cases of games are conducted to assess incremental configurations for the defender' strategy: (1) a local Opinion Dynamics detection around 1 hop of distance from the observed node; (2) local detection with 2 hops of distance; (3) the addition of redundant edges in  $V_{OT}$ ; and (4) the integration of honeypots within the topology. On the other hand, the attacker follows the model explained in Section 6.4.3. Each test case is composed by 10 sets of 100 games, where each set is based on a new generated board, following the network architecture introduced in Section 6.4.1. At the same time, different sizes of network are considered in each test case: 100, 200 and 500 nodes.

Considering a realistic scenario and according to the methodology explained before, we have assigned values for the detection probabilities represented with  $\Theta$ , together with those of  $\Psi$  and  $\Upsilon$  sets, which regulate the criticality and vulnerability of resources in our simulations. This instantiation of values is shown in Table 6.4. For the interest of realism and to represent a certain level of randomness in the accuracy of the detection mechanisms that every agent embodies, these values will also include a random deviation in the experiments, with a maximum value of  $\pm 0.1$ .

$i$	1	2	3	4	5	6
$\psi_i$	0.2	0.3	0.4	0.5	0.6	0.8
$v_i$	0.8	0.7	0.6	0.5	0.4	0.2
$\theta_i$	0.1	0.3	0.4	0.5	0.6	0.9

Table 6.4: Instances of the  $\Psi, \Upsilon, \Theta$  ordered sets used in the simulations

For each board and game set, the percentage of victories achieved by each player (in addition to the ratio of draws) is calculated. These are shown in form of a boxplot, where each box represents the quartiles for each player given the different configurations of size in each case. Different conclusions can be drawn from these simulations, which are discussed in the following.

**Test Case 1: local Op. Dynamics with 1 hop, no redundancy, no honeypots.**

In this case (and related to Figure 6.2), the attacker clearly experiences a high rate of victories as he/she easily escapes from the defender detection, which only encompasses one hop of distance from the affected node. Therefore, the best-case scenario for  $D$  occurs when he/she just manages to follow the infection until it is eradicated in the last turn, resulting in a draw.

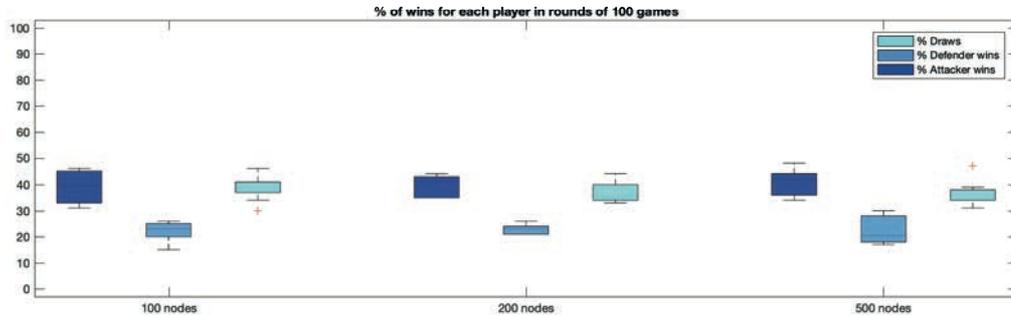


Figure 6.2: Test-case 1: Percentage of victories and draws

**Test Case 2: local Op. Dynamics with 2 hops, no redundancy, no honeypots.** With the introduction of more nodes covered by the local detection (whose number is approximately squared with respect to Test case 1), the percentage of defender wins increases significantly, which shows the importance of applying Opinion Dynamics on a wide area, as shown in Figure 6.3. However, the number of attacker victories and draws still remains moderate, since the defender has not sufficient accuracy as to keep track of  $A$  when the removal of links is performed and the detection is eluded.

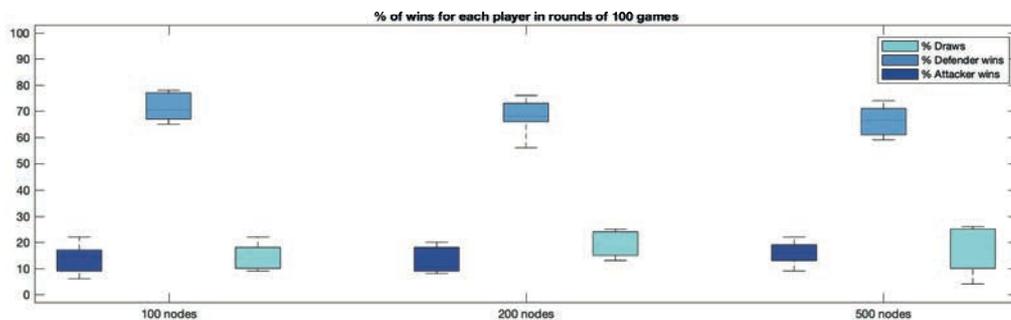


Figure 6.3: Test-case 2: Percentage of victories and draws

**Test Case 3: local Op. Dynamics with 2 hops, redundancy, no honeypots.** The implementation of more defensive aids results in a higher number of wins for the defender (see Figure 6.4). Here, the redundancy makes  $D$  able to trace most of the attacker movements, including when that player wants to get rid of the detection, which is more evident in smaller networks. And yet, the defender must successfully heal all the compromised nodes across the network that

may continue the attack and be far away from the current detection focus, which still returns a mild number of attacker victories and draws.

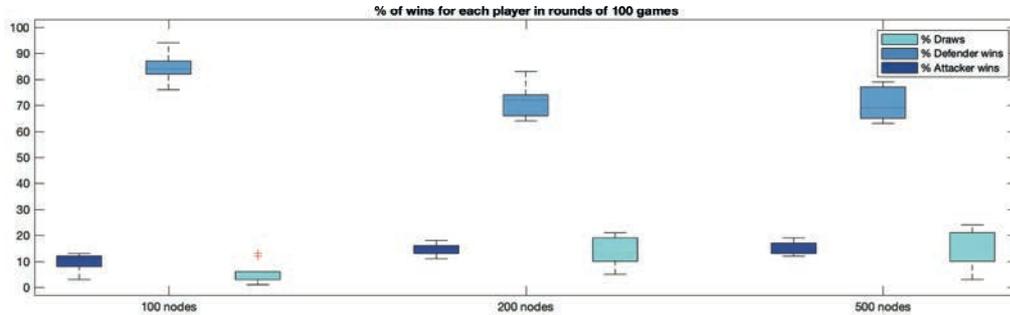


Figure 6.4: Test-case 3: Percentage of victories and draws

**Test Case 4: local Op. Dynamics with 2 hops, redundancy, honeypots.** Lastly, the addition of honeypots are a secure way for the defender to ensure the highest number of victories, as shown in Figure 6.5. The presence of these devices triggers severe anomalies when the attacker tries to compromise them. They are sensed by the defender to rapidly locate the current affected node, as long as  $D$  covers a wide area that contains the position of the attacker at that time. In this case, the use of these two tools (besides the redundancy) are enough as to win most of the games. The rationale behind this result is simple: when the attacker attempts to compromise one of this fake nodes, a great anomaly is generated which is detected by the defender, as long as he or she manages to cover a wide area that contains the current position of the attacker (i.e., when 2 or more hops of distance are leveraged by the local Opinion Dynamics). This behavior is shown in Figure 6.6. In this network, the attacker traverses the nodes and then they are immediately healed (they are labeled with an ‘X’ when they are attacked and ‘H’ when they are healed, along with the anomaly measured by Opinion Dynamics). In the last movement, the attacker attempts to compromise a honeypot (depicted with a diamond shape) and the defender manages to locate and eradicate the infection. Since the defender does not possess any other compromised node, the game is over.

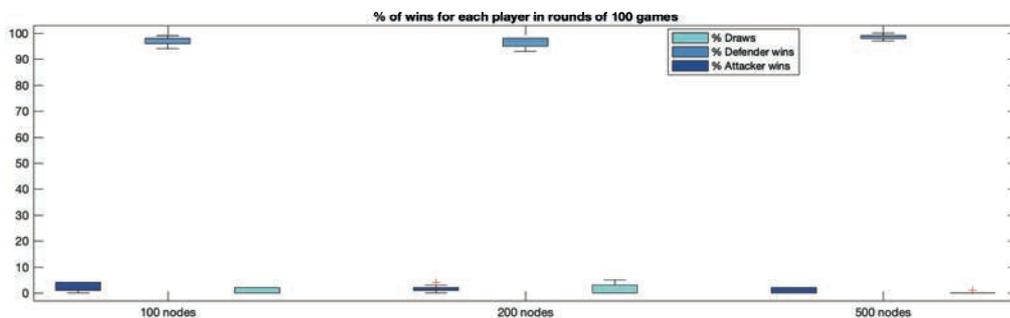


Figure 6.5: Test-case 4: Percentage of victories and draws



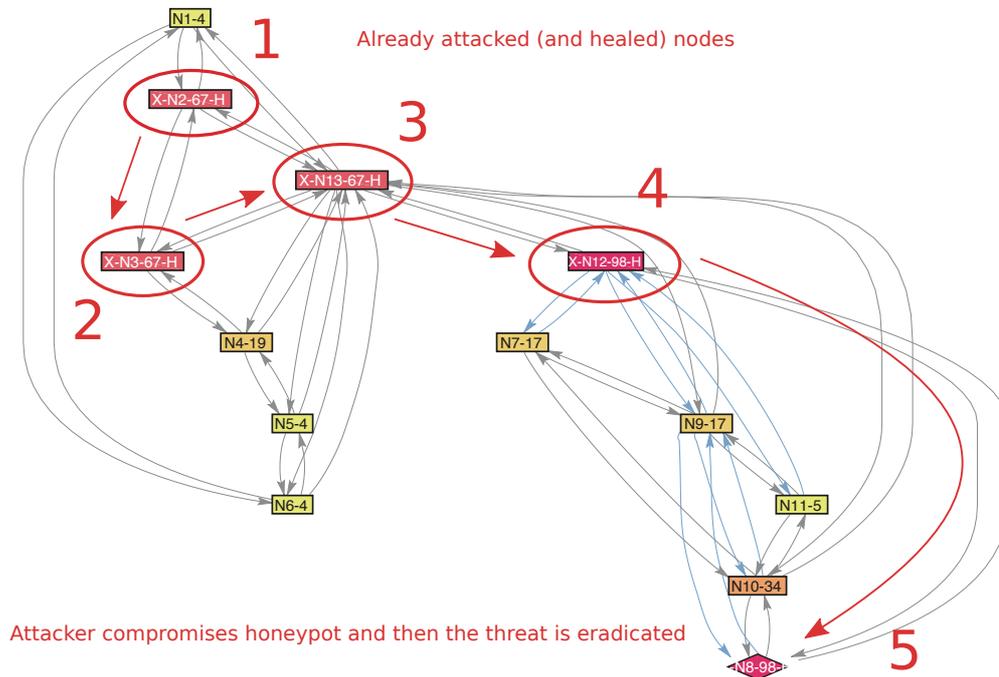


Figure 6.6: Example of defender-win after the attacker compromises a honeypot

In general, we can deduce that solely by implementing Opinion Dynamics, the defender can benefit from its detection to reduce the impact of the attacker over the network. The protection improves with the introduction of additional measures such as redundancy or honeypots, and the same results are obtained for different sizes of network.

We can also draw some analysis on the overall score in these test cases. Figure 6.7 plots the average score of the defender and attacker for the four test cases presented before. At a glance, we can see how  $D$  shows a superior throughput in all cases, and a slightly higher score when using low-size networks, since he/she experiences greater accuracy in the reparation of nodes. Also, the score decreases as test cases implement additional defense measures. On the one hand, the attacker generates more anomalies (and hence more costs) due to the link removal attacks in the attempt to dodge the detection. On the other hand, the defender has more candidates to heal due to the increased number of anomalies, and does not always have a high accuracy in choosing them.

To sum up, by means of game theory we have demonstrated that local Opinion Dynamics is still valid for catching the compromised nodes of the attacker when it is applied with a minimally wide detection area (i.e., two hops of distance from the observed node) and it is paired with effective response techniques (i.e., where honeypots pose an effective measure) that precisely make use of the provided detection information. The game approach itself is validated from a theoretical point of view in the next section.

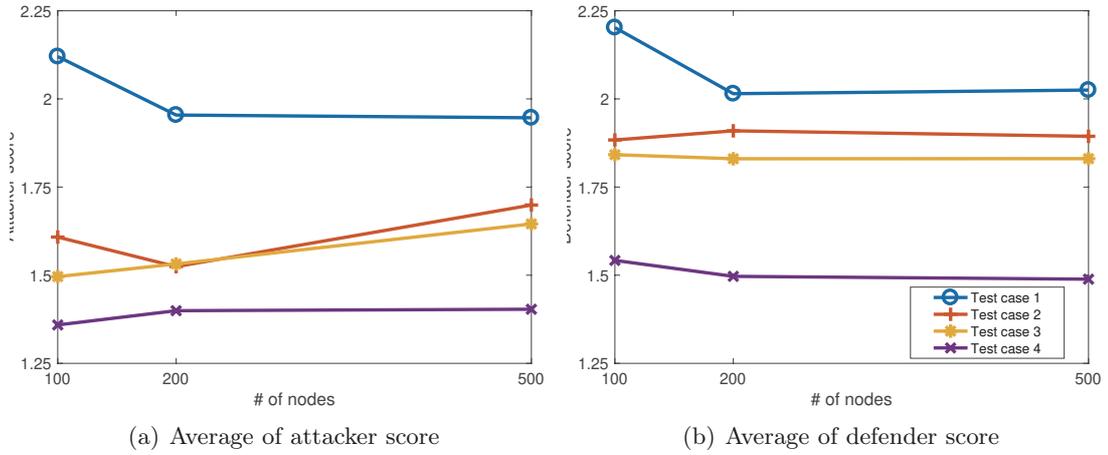


Figure 6.7: Percentage of victories for each player in each test case

### 6.4.5 Theoretical Demonstration

This section presents the correctness proof of TI&TO for the different cases that may occur during a certain game instance. This problem is solved when these conditions are met:

1. The attacker can find an IT/OT device to compromise within the infrastructure.
2. The defender is able to trace the threat and heal a node, thanks to the Opinion Dynamics detection.
3. The game system is able to properly finish in a finite time (termination condition).

The first requirement is satisfied since we assume that the attacker can perform different attack stages to define his/her strategy over the game board (assuming  $V \neq \emptyset$ ), such as lateral movements, links removal or destruction. The modus operandi of the attacker is systematic, beginning with a random node  $v_0 \in V_{IT} \cup V_{OT}$  at  $t = 0$  which is compromised (see Algorithm 12). Then,  $A$  penetrates the infrastructure to ultimately gain control of the operational or corporate network, where a certain node is finally disrupted ( $V_{OT}$ ) after a set of  $\sigma$  lateral movements. In an intermediate time  $t$  of the game, the attacker can execute a new stage as long as there is at least one node  $v_a$  such that  $N_{v_a}(t) = 1$ , which becomes the new *attackedNode* in Algorithm 12. When the state of all nodes is set to zero, the game terminates.

The second requirement is also met with the inclusion of intrusion detection solutions on every agent  $a_i \in A$  that facilitate the correlation of events. With the local execution of the Opinion Dynamics correlation from  $t = 1$  on the node that presents the greatest anomaly (using one or two hops of distance), we ensure that the agents associated with the resulting subgraph of nodes will have an opinion  $x_i(t) \geq 0$ . According to Algorithm 13, this means that  $D$  will heal the node with the maximum opinion if that value surpasses the threshold (0.5, as explained in Section

6.4.3), setting its state back to zero and updating the detection area. Otherwise, he/she will remain idle during that turn.

We can demonstrate the third requirement (corresponding to the termination of the approach) through induction. More precisely, we specify the initial conditions and the base case, namely:

**Precondition:** we assume the attacker models an APT perpetrated against the infrastructure defined by graph  $G(V, E)$  where  $V \neq \emptyset$ , following the strategy explained in Algorithm 12. On the other side, the defender leverages Opinion Dynamics to visualize the threat evolution across the infrastructure and eventually repair nodes, following the procedure described in Algorithm 13.

**Postcondition:** the attacker reaches the network  $G(V, E)$  and compromises at least one node in  $V$  such that  $S^A \neq \emptyset$  and continues to compromise more devices in the loop in Algorithm 12, to achieve  $numSteps = \sigma$ . Player  $D$  executes Opinion Dynamics to detect and heal the most affected nodes after executing the correlation. The game evolves until any of the termination states (see Section 6.4.2) are reached.

**Case 1:**  $numSteps = \sigma$ , but  $gameState$  is still set to zero. In this case, player  $A$  has successfully traversed the network having  $Victims \neq \emptyset$ . Therefore, he/she needs to launch the Destruction movement over the *attackedNode*. This makes  $gameState$  comply with  $TS_1$  termination condition temporarily until the defender moves. If  $D$  manages to heal *attackedNode* and  $Victims = \emptyset$ , then the game also terminates, with  $TS_3$ .

**Case 2:**  $numSteps < \sigma$ . In this case, the next stage in  $S^A$  implies a lateral movement. If the attacker is still in the first section where the first intrusion took place (whether IT or OT), he/she must locate a firewall to perpetrate the other section before increasing  $numSteps$ . After this, the defender can make his/her movement and potentially heal a node, which can make the attacker remove a link in the following iteration. If the node healed is *attackedNode*, the attacker must choose another node in  $Victims$ , resetting  $numSteps = 0$ . In the event that  $Victims = \emptyset$ , then the game terminates with state  $TS_2$ .

**Induction:** if we assume that we are in step  $t$  ( $t \geq 1$ ) in the loop in Algorithm 12, then Case 1 is going to be considered until  $A$  completes his/her strategy ( $TS_1$  or  $TS_3$ ). In any other case, Case 2 applies until achieving  $numSteps = \sigma$  (hence applying Case 1 again) or  $Victims = \emptyset$ . In this last case, the game finishes with  $TS_2$ .

## 6.5 Validation in a Testbed

In this section we will go beyond the theoretical experiments described in the previous section, and provide the experimental results of a proof of concept implementation of the APT traceability framework with the Opinion Dynamics system. This proof of concept was integrated on a testbed

that simulates an industrial environment using realistic hardware and protocols. For this proof of concept, rather than integrating a full-fledged network-based and host-based intrusion detection system as an input for the Opinion Dynamics algorithm, we deployed a set of simple heuristics that searched for anomalies in the communication channel. The reason for this is simple: this experiment aims to provide a baseline that shows how the Opinion Dynamics system can help to provide the trace of a kill chain while using as an input only lightweight anomaly detection rules.

As for the structure of this section, firstly we present and provide the technical specifications of the testbed used for the simulations (the so-called I4Testbed). Then, we explain how the Opinion Dynamics system has been applied in this context. Finally, we describe the execution of the different attacks cases, and analyze the results provided by the Opinion Dynamics system.

### 6.5.1 I4Testbed: An Industry 4.0 Testbed

The advent of the Industry 4.0 paradigm is basically a consequence of a plethora of technologies that are being imported from the IT world (e.g., the Internet of Things, cloud computing, Big Data) to industrial control systems, which have been working in an isolated way for decades. This has also caused the appearance of new attack vectors against these infrastructures, which has fostered the research of advanced cybersecurity solutions. Precisely, the *I4Testbed* testbed has been developed in the University of Malaga to provide a realistic environment where novel detection mechanisms can be assessed without facing the whole investment of deploying a complete industrial infrastructure.

The overall architecture of the *I4Testbed* is depicted in Figure 6.8. It is designed to accommodate different industrial applications in a realistic fashion. For this particular case, we model a solar, hydraulic and wind electricity generation system. Each of the three sources are virtually simulated by using an open API that retrieves the climate conditions in Malaga in real time [336]. These values are then fed to the physical sensors, so that the turbines are ultimately activated from the SCADA system depending on specific conditions of humidity and temperature.

As shown in Figure 6.8, different devices are placed in the lowest level of the topology, which includes light indicators, emergency buttons, industrial sensors (using protocols such as IO-Link, WirelessHART and ISA100.11a) and IoT sensors (TelosB using 6LoWPAN over IEEE 802.15.4). These sensors are connected to their respective gateways which, along with other field devices based on Intel Galileo Gen1, RevPi Core 3 and Raspberry Pi, gather the different measures of the generation process and then relay them to three different PLCs: one SIMATIC S7-1200 (using Profinet) that governs the hydraulic generator, one PLC based on Raspberry Pi 3 (using ModBus TCP) that controls the eolic and solar generator, and another one implemented purely via software, that controls the AC system of the power transformer. These three PLCs are then operated by the SCADA system (which is based on Linux with Python) and two different HMIs: one SIMATIC KTP700 and another one implemented with a Raspberry Pi. This SCADA system, that also works as HMI (as shown in Figure 6.9), and the IBH Link UA Gateway can be accessed

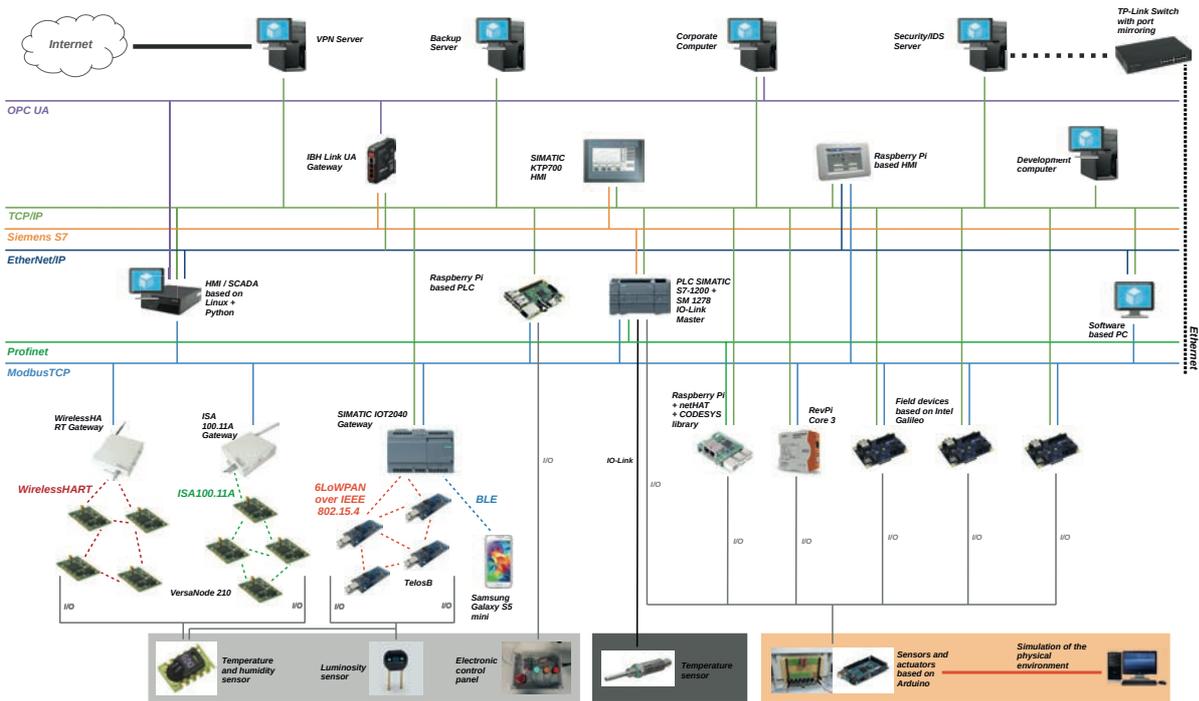


Figure 6.8: Overall architecture of the  $I_4$ Testbed testbed

by local entities through TCP/IP and OPC UA; and by external entities through a virtual private network (VPN) connection. Additionally, the testbed also integrates a backup server, a corporate computer and another one for development purposes.

As for the monitoring capabilities of the  $I_4$ Testbed, the previously presented topology contains a security server with high computational resources that is able to capture all the information from the communication channels via a network switch in port mirroring mode. Despite the logical topology, as all devices are physically connected through one switch, the security server can retrieve all the traffic from the nodes. This way, the security server can also function as a centralized entity (as discussed in Section 4.2.1), where we can deploy a virtual agent for each physical node that must be monitored. Such agents will then perform the different computations of the Opinion Dynamics algorithm.

### 6.5.2 Implementation of the Virtual Agents

Within the Opinion Dynamics system, every agent will process the traffic handled by its associated physical node, and study the security state of its neighbourhood. As a result, it will create a quantitative value (i.e., the opinion of that agent) which will be used as an input to the Opinion Dynamics algorithm. For the purpose of our experiments, in this proof of concept implementation

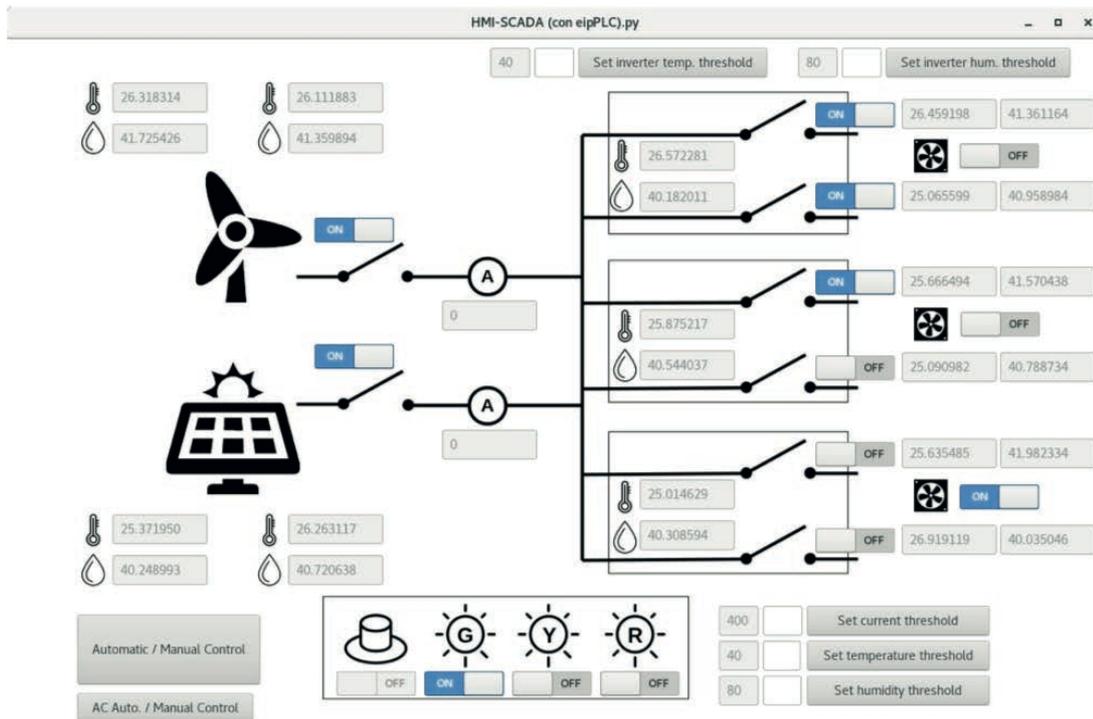


Figure 6.9: Interaction panel GUI on the SCADA system

we will make use of an heuristic to compute the unique anomaly value, which considers the following characteristics:

- Variation of traffic volume: by analyzing the number of packets per protocol and device connected with each channel: Enabling the detection of added/removed devices within the topology, in addition to non-frequent communications.
- Variation of the commands received by the industrial protocols: through the analysis of the number and type of commands, with the aim of detecting anomalous actions performed by potentially compromised devices.
- variation of the delays experienced between received commands by the industrial protocols: To infer the presence of anomalous processes running in each device.

For the computation of an unique anomaly value, the average and standard deviation of the different characteristics monitored (e.g., number of connections, packets exchanged) are calculated in normal conditions. For the sake of simplicity, we have assumed that, for a given characteristic, a value sensed at any time is considered as anomalous when it exceeds the standard deviation of such characteristic in normal conditions. Finally, the opinion of each agent is chosen as the highest anomaly value for all the characteristics monitored. Even though this criteria is an adjustable

parameter for the simulations, we have specifically considered the following equation to compute the anomaly value for a given characteristic:

$$\left(\frac{(NormalValue - CurrentValue) - 2 * StdDev}{StdDev}\right)^2 * 5 \quad (6.1)$$

Then, the process is analyzed periodically to sense multiple anomalies across the entire topology. For this test case, we have considered slots of 5 minutes: during this period, each pair of devices that exchange information are considered as neighbours, and all characteristics of the communications are gathered by each agent to compute its anomaly degree. Lastly, these anomalies (i.e., the agents opinions) are correlated using Opinion Dynamics, to ultimately output the health status of the industrial system.

In order to implement the virtual agents, we have deployed three different components (cf. Figure 6.10) in the security server. These components are as follows:

1. A **collector component** retrieves the raw traffic from all devices of the testbed to generate a list of events that are of interest for the analysis of the variation in each characteristic.
2. Then, the **detector component** creates one agent for each of the components that are deployed over the network. This agent analyzes the different characteristics involved for its monitored node and computes Equation 6.1 to finally obtain an opinion value.
3. Finally, a **correlator** executes the Opinion Dynamics algorithm to accurately identify the most affected areas of the infrastructure, as explained in Section 4.3.2. In addition, the  $\delta$  value is also returned to represent the overall health status of the network.

For this particular experiment, these three components have been developed using Python 2.7.13. In order to capture the network traffic, we have also used the *scapy* library and several dissectors such as *scapy-cip-enip*.

### 6.5.3 APT Test Case with I4Testbed

In this section we show how the Opinion Dynamics-based technique performs against a test case of an APT composed by four different attack stages. The aim is to check how the different agents that are spread over the topology sense the different anomalies caused by these vectors. As a result of this analysis, the system should provide a trace of the whole attack, plus an aggregated indicator of the health of all resources of the I4Testbed. In order to (i) achieve an acceptable degree of realism, and (ii) provide as many sources of anomalies as possible, the entire kill chain has been defined as a sequence of the following stages:

1. **First intrusion:** an initial access to the network is perpetrated. More specifically, the adversary (potentially an insider) steals some access credentials (e.g., with social engineering) and takes over the HMI/SCADA by accessing it from the IT network via SSH.



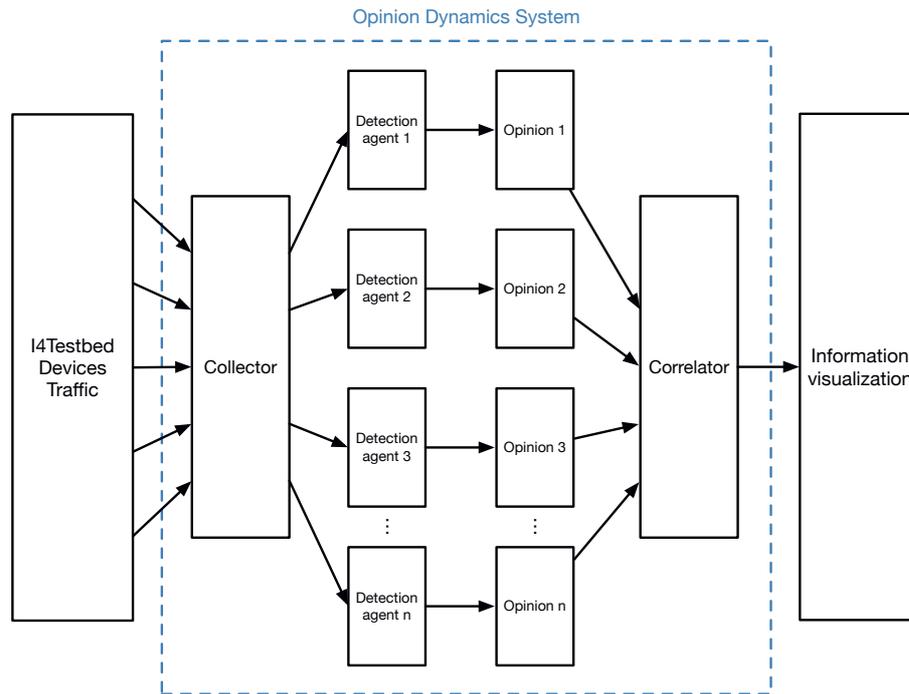


Figure 6.10: Components of the Opinion Dynamics System

2. **Network scanning and lateral movement:** once higher privileges have been obtained and the SCADA system has been compromised, the attacker performs a reconnaissance of the node neighbourhood, seeking for vulnerable services running in each device. This is achieved issuing a *nmap* command on Linux. At this point, we assume that a vulnerability is found on the Raspberry Pi-based PLC and is exploited to take over that node.
3. **Establishment of a covert-channel:** after the PLC has been compromised, the adversary establishes a covert-channel attack against the Modbus communication link. Through this channel, the adversary sends a shutdown command that is expected to be executed in a latter phase. This is perpetrated in a stealthy way, delaying the transmission of a Modbus message, as explained in [301]. There are various publications available in the literature that also explain potential implementations of this attack, such as [337] and [338].
4. **Node disruption:** finally, the PLC executes the shutdown command and closes the communication links with the rest of devices.

In order to visualize how these attacks are detected and reported by the Opinion Dynamics System, the experimentation has been carried out according to the following methodology: firstly, the industrial system is left to work for an hour without taking any special action on the testbed, except for computing the detection algorithm periodically every 5 minutes. This helps the virtual agents (one per device) to compute the average and standard deviation of the different

characteristics introduced before (traffic volume, number of connections and communication commands, etc) in normal conditions. Afterwards, the entire kill chain is executed in sequence, with a waiting time of approximately one hour between the various stages of the attack. During the execution of the kill chain, the Opinion Dynamics system keeps being executed, so that we can keep track of the multiple anomalies measured as attacks take place.

As a result, Figure 6.11 shows an abstract representation of the I4Testbed devices and their connections, along with the respective correlated opinion of all virtual agents, which is computed immediately after each individual attack. Note that, in our experiments, two devices are considered as neighbours by the Opinion Dynamics as long as they exchange information during the last period analyzed (i.e., every 5 minutes, as explained before). This way, the system can detect when a device has been removed from the topology, which affects the anomaly calculation due to a variation on the number of connections. Note also that, in Figure 6.11, dotted lines represent connections that are not used frequently.

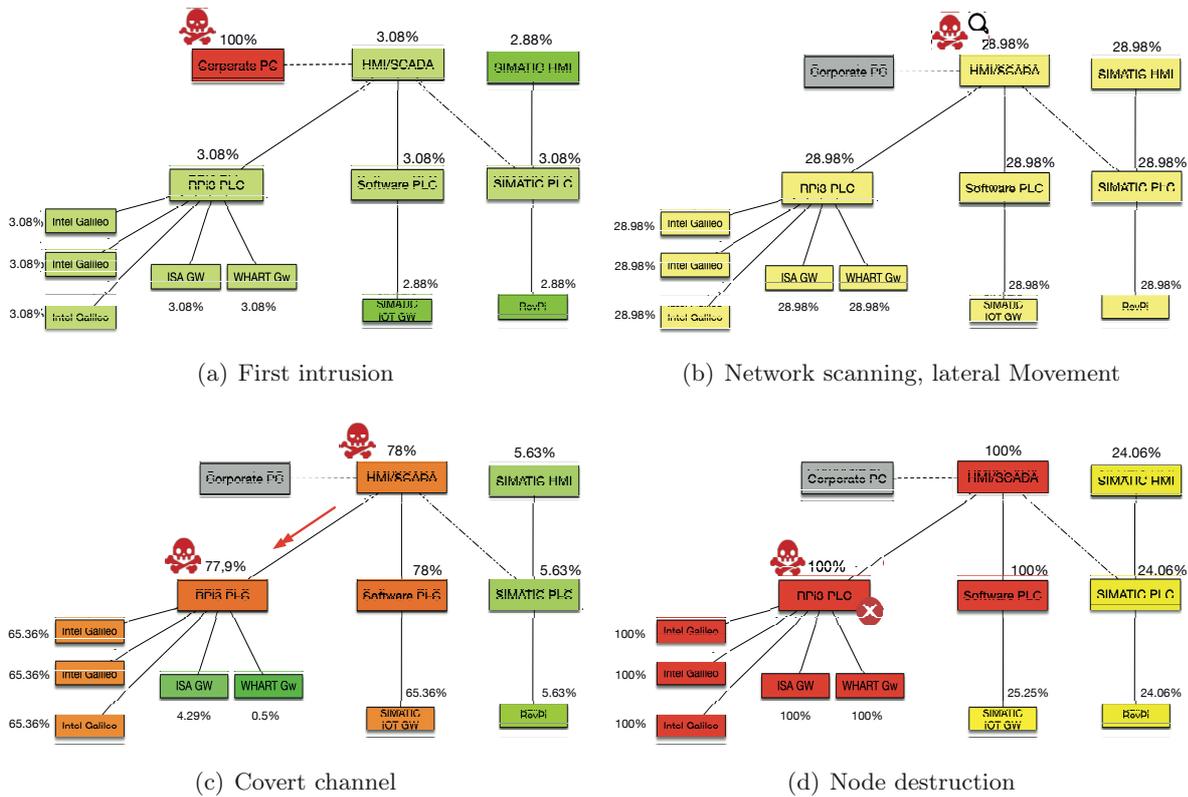


Figure 6.11: Evolution of the Opinion Dynamics values over the test case attack stages

As we can see, the correlation of the different opinions of the virtual agents provide helpful information that is of interest to network security mechanisms and services, as it provides an accurate visualization tool to easily identify the most affected resources at all times. First, Figure 6.11(a) shows that an important anomaly was detected by the virtual agent assigned to the

corporate PC when the SSH connection was opened to communicate with the HMI/SCADA, since it notices an unexpected connection involving that target device. Besides, if host-based IDSs were available, the virtual agent assigned to the HMI/SCADA would also have signalled the existence of an anomaly. Note that this corporate PC is opted out for the Opinion Dynamics computation after this first step because it will not have any more interactions with the rest of the devices during the entire simulation. Then, as seen in Figure 6.11(b), the search for victim devices within the network results in a mild increase in the opinion of most agents, since the network is flooded with TCP connections.

Thirdly, the adversary establishes the covert-channel between the HMI / SCADA and the Modbus PLC, with the aim to issue commands without firing any alert. However, this attack is also detected when the variation of the packet delays is analyzed by the agents involved, which is leveraged to embed the shutdown command for the target PLC; in other words, different clusters of opinions appear as consequence of the correlation of similar opinions due to similar delays experienced in their surroundings links. These are represented in orange in Figure 6.11(c). Lastly, the attacker sends a shutdown command to the RPi3 PLC, paralyzing the production chain. As expected, this generates a critical anomaly (cf. Figure 6.11(d)) that is measured by all devices that work closely to that device. Such anomaly is a consequence of the variation in the traffic volume, caused by delays and requests issued by the industrial devices; namely, the WirelessHART and ISA100.11a gateways, the field devices, the HMI/SCADA system, and the Software PLC.

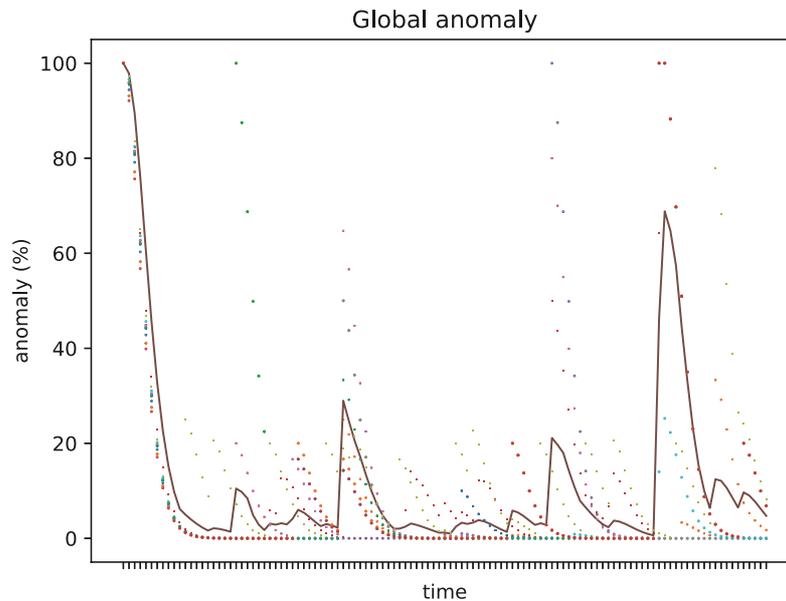


Figure 6.12: Evolution of delta opinions over the test case attack stages

Apart from providing a detailed analysis of the security status of all devices, the Opinion Dynamics System can also provide the health status for the entire network by calculating the  $\delta$  indicator, as introduced in Section 4.6. In particular, Figure 6.12 shows the resulting value of the global anomaly (1 minus the *delta* indicator), calculated as the weighted average of all individual opinions (also represented in the graph) during the entire simulation. In the figure, each mark in the X axis represents a single computation of the Opinion Dynamics algorithm. There are two important aspects that must be highlighted in this figure. First, as all agents run a training phase to determine the normal conditions of the system, every event is considered as an anomaly during that process until they stabilize around zero value. Second, the four different attack stages are actually shown in the figure as peaks in the y-axis. As explained before, the highest peak occurs with the ultimate disruption of the PLC, which results in a global anomaly of 68.83% (so that  $\delta=31.17\%$ ).

# Chapter 7

## Conclusions

This chapter recapitulates our efforts to achieve the goal of improving APT traceability techniques in modern control systems with the advent of Industry 4.0. We begin by summarising the scope of our research and the problems addressed in this thesis. We then present a brief description of our contributions in this area and expand on possible lines of improvement. Finally, we introduce some remaining challenges for the detection and traceability of these sophisticated threats.

### 7.1 Contributions

Today, most critical infrastructures in all industrial sectors (such as transport, the electricity grid or telecommunications) base their management on SCADA control systems. These allow real-time remote access to the devices that govern their production chain. In terms of cybersecurity, these systems have traditionally been deprived of services to deal with external threats, since industrial networks had to operate in isolation from other environments. However, in recent years there has been a gradual interconnection of control systems with other networks (such as the Internet) for the outsourcing of services or data storage, which has been preceded by cheaper equipment and the standardisation of software used in industrial ecosystems. Added to this is the current integration with new information technologies such as cloud computing, Big Data, virtualisation and the Internet of Things. Consequently, the industry is evolving towards a more flexible model where all parties (consumers, suppliers, operators) collaborate to streamline the production chain more interactively, carry out productive maintenance of resources and reduce costs, in what is already known as the fourth industrial revolution or Industry 4.0.

As a result of such evolution in process interoperability, there has also been an evident growth in cybersecurity threats, as industrial systems are now also victims of IT problems, in addition to the risks posed by new communication protocols and Industry 4.0 services. In short, we have greater criticality and complexity in industrial systems, which must be supported by more effective security services. Of particular interest is the implementation of mechanisms against so-called advanced persistent threats. These are sophisticated attacks perpetrated against a specific



organisation, where the perpetrator has considerable expertise and resources to penetrate the victim's network by exploiting a multitude of vulnerabilities and attack vectors, going undetected for a prolonged period of time.

After contextualising the technological background of Industry 4.0 and carrying out a thorough review of the state of the art, it is deduced that there is a lack of mechanisms that allow the detection and effective monitoring of APTs in modern infrastructures, which is the main goal of this thesis. To reach this conclusion, we first carried out a study of the threats to which an industrial control system is exposed and which can form part of an APT. These include attack vectors such as malware, denial of service, code injection, privilege escalation, etc. as well as vulnerabilities in communication protocols and social engineering attacks (e.g., spoofing, phishing). These were classified according to a more detailed taxonomy, grouping them according to the traditional security services concerned: availability, integrity, confidentiality and authentication. At the same time, a differentiation was made between threats that arise as a consequence of weaknesses inherent to industrial systems and those that appear as a result of the integration of IT technologies and Industry 4.0 services in these environments. In light of this taxonomy, we researched solutions that would allow us to put in place a first line of defence in the form of intrusion detection systems, which analyse assets within the organisation in search of anomalies and attack patterns. Specifically, we surveyed more than 100 mechanisms proposed in the commercial, academic and research fields, to classify them according to the type of techniques they use, their level of coverage within the infrastructure, and the type of communication protocol they analyse. From this study, we also deduced that little progress has been made in research into detection techniques that allow us to monitor industrial resources in a holistic manner and simultaneously detect a multitude of attacks, as is the case with APTs. In this sense, all solutions for traceability of attacks in Industry 4.0 focus on specific attacks or have not been proven effective in realistic environments.

For this reason, to close the gap between detection systems and APTs in Industry 4.0, we distilled a set of detection and security requirements that new techniques in this field must meet. These are specified in a traceability framework for the traceability of APTs, which also defines the input interface, a deployment model and the services to be satisfied by potential anomaly correlation algorithms in an industrial infrastructure for the precise traceability of an adversary's movements, which have also been specified. To this end, we conducted a study on the most relevant APTs in the last decade and formalised a realistic attacker model. Then, to illustrate the benefits of this framework, we designed three different techniques based on consensus and clustering, which carry out the distributed correlation of anomalies detected by a set of agents spread across the network. After an initial comparison of these solutions through experiments that evaluate their accuracy in detecting a set of theoretically modelled APTs, we find that the Opinion Dynamics technique is the most flexible and accurate. This algorithm simulates the influence of opinions among a set of agents (which in our case represents the perceived anomaly in their local environment) and their evolution over time. When these opinions are eventually

grouped, we can extract valuable information to determine in which parts of the network the attacker is located and associate aspects such as the persistence and criticality of the attacks, thus fulfilling the requirements initially proposed.

Based on these findings, our objective below was to test the effectiveness of the traceability framework (and therefore that of the solutions that satisfy it) in various Industry 4.0 scenarios from a more practical point of view. For this purpose, we firstly devise response techniques that use the information provided by the detection system to ensure the survivability of the network by guaranteeing the continuity of communications in the presence of an APT. This was implemented utilizing message routing protocols that make use of the information provided by the Opinion Dynamics algorithm, and then they tested with various attack scenarios. On the other hand, its application was also studied in the Industrial Internet of Things, which is a fundamental pillar of Industry 4.0, and in Smart Grid, as use case of sector of the Industry. As for the former, the deployment of the detection system in this industrial paradigm was addressed, studying data extraction at all levels and proposing a theoretical prototype that illustrates the effectiveness of intrusion detection when integrated with this technology. In the case of the Smart Grid, a tool was developed that prevents against potential overloads within the network and monitors anomalies to provide information on the security of resources. This data is then used to establish access control policies based on the real-time status of the infrastructure.

It is worth noting that the effectiveness of each of these detection mechanisms and their derived response techniques has been supported by mathematical proofs of the underlying algorithms. In particular, our Opinion Dynamics-based solution was validated theoretically through game theory, which also helped to draw several conclusions about the optimal defence strategy against certain types of APTs. Similarly, this solution was successfully implemented in an industrial testbed, demonstrating the deployment and traceability of events occurring in the context of an APT composed of several attack phases against different industrial elements.

As a result, this research is of particular interest to raise awareness of this problem in the critical infrastructures that will control our society in the coming decades. In particular, the traceability framework provides useful information for the design of detection systems adapted to the complexity and technological heterogeneity of these environments. This is evidenced by the various experiments carried out, which highlight the accuracy and efficiency of these solutions. Such is the case of Opinion Dynamics, where its contribution translates into better decision making due to real-time monitoring of resources, risk prevention and ultimately the reduction of impact (and therefore costs) thanks to the response services that make use of these innovative solutions.

## 7.2 Challenges and Future Work

The security of critical infrastructures is a hot topic, and is the main obstacle to the adoption of Industry 4.0 in all sectors of society. Proof of this are the numbers of reports that reveal the

millions of dollars in losses caused by targeted attacks against all types of companies. In this line, there are still many challenges and open problems that require aligning the standardization efforts of these technologies with the knowledge of experts and the multidisciplinary collaboration of governments, consortia and private entities. In particular, this thesis has focused on a very specific type of problem (the detection and traceability of APTs) that by its nature encompasses multiple areas of cybersecurity and, therefore, leaves room for a deeper analysis in each of them.

First of all, it is worth mentioning certain possible improvements and extensions of the traceability framework and its enabling solutions. On the one hand, it would be interesting to consider the assimilation of more input data that would result in a more accurate anomaly correlation. At this point, we wish to illustrate in a practical way the automated (and real-time) processing of external information sources (such as cyber-threat intelligence reports) to maintain an updated knowledge base concerning event causation, which we have referred to as the qualitative input. In the particular case of the Opinion Dynamics algorithm, this would translate into an optimal and automatic allocation of weights among agents. So far we have left the door open for these values to be established manually based on different rules, for example, by measuring the quality of service in communications. However, it would be ideal if the platform in question had the autonomous capability to acquire this knowledge and calculate the best assignment, perhaps using machine learning or deep learning techniques. In general and in relation to the latter, it would also be of particular interest to explore the influence of artificial intelligence and machine learning algorithms in the correlation of anomalies, beyond the mechanisms based on distributed consensus and clustering.

Similarly, it would be desirable for the traceability system to offer more output functionalities beyond assessing the current state of the devices and identifying the most affected areas. For example, we would like to study the possibility of making predictions with the data collected, in order to anticipate with certainty the next movements of a stealthy attack within the victim network. Again, this leads us to analyze in detail the machine learning tools applicable in this area, which would operate on the basis of a database containing all the events occurring in the infrastructure, thus fulfilling the traceability functionalities. In this sense, the integration of DLT structures becomes relevant in critical environments to ensure the integrity and replication of data in the long term, as we have incidentally addressed in this thesis for the case of the Smart Grid. However, it is necessary to investigate how these technologies impact grid performance, and how they could be integrated with sensing agents based on their distributed nature.

The deployment of the techniques that satisfy our framework also imposes several challenges that need to be further addressed. In particular, while we have defined different models for implementing the sensing agents physically or virtually (which determines how the information extracted from the field devices is acquired and aggregated), we should also analyze how each of these strategies affects the performance of the control systems, drawing conclusions about the advantages and disadvantages of centralized or distributed correlation. At the same time, it is to be expected that the algorithms that satisfy the traceability framework itself do not present a



high complexity to jeopardize the real-time requirements imposed by these critical environments. All of this is dependent on a set of performance requirements that are specific to the industrial systems where these solutions would be applied, and which is different in each Industry 4.0 sector. For space reasons, this thesis has addressed the case of the Smart Grid and the industrial internet of things as a technological pillar of manufacturing infrastructures. However, we would like to study the behavior of the Opinion Dynamics algorithm (along with other alternative solutions) in additional environments such as the transport network or telecommunications, in order to identify further parameters that could be contemplated by our framework to ultimately characterize our solutions with a higher degree of accuracy.

This thesis has focused on analyzing the generic behavior of APTs based on the most important cases that have been reported in recent years. However, it is expected that the new services offered in the Industry 4.0 by new computing and communications paradigms (such as blockchain, 5G or fog/edge computing) will remain on the rise, thus creating new attack vectors that will force the renewal of existing techniques to cover a broader detection approach. It is therefore crucial to continue researching adaptive detection mechanisms with an increasing degree of autonomy, based on increasingly complex attacker models that are also recognized in academia. This must be supported by new standards that integrate security by default throughout the Industry 4.0 life cycle (such as IIRA and RAMI4.0), to facilitate seamless integration with future traceability solutions.



## Apéndice A

# Resumen en español

Hoy en día, la mayoría de infraestructuras críticas de todos los sectores industriales (como el transporte, la red eléctrica o las telecomunicaciones) están experimentando un proceso de modernización tecnológica. Estos sistemas basan su gestión en los denominados sistemas SCADA (Supervisory Control and Data Acquisition), que permiten el acceso remoto en tiempo real a los dispositivos que gobiernan la cadena de producción. En lo que a ciberseguridad se refiere, estos dispositivos han estado tradicionalmente desprovistos de servicios que hagan frente a amenazas externas, puesto que las redes industriales debían funcionar de manera aislada a otros entornos. Sin embargo, en la actualidad se está llevando a cabo una paulatina interconexión de los sistemas de control con otras redes (como Internet) para la externalización de servicios o el almacenamiento de datos, algo que viene precedido por el abaratamiento del equipamiento y la estandarización del software empleado en los ecosistemas industriales. A ello se le suma la integración de estos dispositivos (que podemos considerar tecnología operacional, en adelante OT por sus siglas en inglés) con tecnologías de la información (IT) tan novedosas como el *cloud computing*, la *blockchain*, el Big Data o el Internet de las Cosas. En consecuencia, estamos asistiendo a una evolución en el modelo de industria donde todas las partes (consumidores, proveedores, operadores) colaboran entre sí de forma distribuida para conseguir un mayor rendimiento con menor coste, en lo que ya se conoce como la cuarta revolución industrial (o Industria 4.0).

Como resultado de tal evolución e interoperabilidad, también se ha producido un evidente crecimiento de amenazas de seguridad, al ser ahora los sistemas industriales también víctimas de los problemas que sufren las tecnologías de la información, además de los riesgos que entrañan los nuevos protocolos de comunicación. En suma, tenemos una mayor criticidad y complejidad en los sistemas industriales, que ha de ser respondida con servicios de seguridad más efectivos. Es de especial interés la puesta en marcha de mecanismos contra las denominadas Amenazas Persistentes Avanzadas (APT, del inglés Advanced Persistent Threats). Se trata de ataques sofisticados perpetrados contra una organización en concreto, donde el responsable posee experiencia y recursos considerables para penetrar en la red de la víctima aprovechando multitud de vulnerabilidades y vectores de ataque, pasando desapercibido durante un prolongado periodo de tiempo.



El foco de esta tesis se enmarca en el contexto de la exploración, diseño e implementación de servicios de detección ante este tipo de amenazas relacionadas con los sistemas de control de los entornos de Industria 4.0. Los esfuerzos de nuestro trabajo se centran en crear una base de conocimiento que nos permita especificar una serie de requisitos de seguridad y de detección, recogidos en un marco de trabajo que sirva como base para el desarrollo de soluciones de trazabilidad de APT en estos entornos.

## A.1 Marco de la tesis, objetivos y contribuciones

Para contextualizar nuestra investigación, es preciso considerar el marco industrial y las tendencias actuales que configuran el paradigma de la Industria 4.0 o cuarta revolución industrial. Se le acuña este nombre en referencia al proceso de modernización tecnológica que actualmente están experimentando los sistemas de control industrial (ICS), tras la incorporación de la mecanización con las máquinas de vapor, la electricidad y la automatización electrónica en las anteriores revoluciones a lo largo de la historia. No obstante, el concepto de Industria 4.0 no está tan maduro debido a la falta de acuerdo sobre el conjunto de tecnologías consideradas y a los diferentes intereses de los actores implicados (incluyendo investigadores, comités de estandarización, empresas y entidades gubernamentales) [5].

En este sentido, cabe destacar principalmente dos iniciativas a nivel internacional que guían gran parte del progreso actual de la Industria 4.0: por un lado, el programa alemán *Industrie 4.0*, que nació como una iniciativa de ámbito europeo, y que ha alcanzado un alcance global debido a su influencia en otros programas y a las colaboraciones entre diversos consorcios. En el caso de España, se ha visto implementado en la llamada Comisión de Industria 4.0 [14], dependiente de AMETIC (Asociación Multisectorial de Empresas de Tecnologías de la Información, Comunicaciones y Electrónica). Por otra parte y con una capacidad de influencia internacional equiparable, en el ámbito estadounidense prevalece el denominado Industrial Internet Consortium (IIC)[21], cuyo objetivo igualmente es el de automatizar la industria en multitud de dominios. Es precisamente de la mano de IIC de donde surge uno de los conceptos que se relacionan frecuentemente con la Industria 4.0: el Industrial Internet of Things (IIoT), promovido inicialmente por empresas americanas (AT&T, Cisco, General Electric, IBM e Intel). Ambas persiguen objetivos similares, pero con ligeras diferencias. Mientras que la Industria 4.0 centra sus esfuerzos principalmente en los procesos de fabricación, el IIoT también busca la integración con diversos ámbitos industriales (por ejemplo, infraestructuras críticas, ciudades inteligentes). Además, la Industria 4.0 se centra más en el hardware y en la coordinación de los procesos de producción, mientras que el IIoT se centra más en el software y en la interacción entre entidades [30]. Aun así, hay puntos en común entre ambas iniciativas, y actualmente están trabajando para alinear sus dos arquitecturas de referencia: la Arquitectura de Referencia de Internet Industrial (IIRA), desarrollada por el IIC [31], y el Modelo Arquitectónico de Referencia Industrie 4.0 (RAMI4.0), desarrollado por el consorcio Platform Industrie 4.0 [32]. Ambas referencias proporcionan dos arquitecturas interoperables

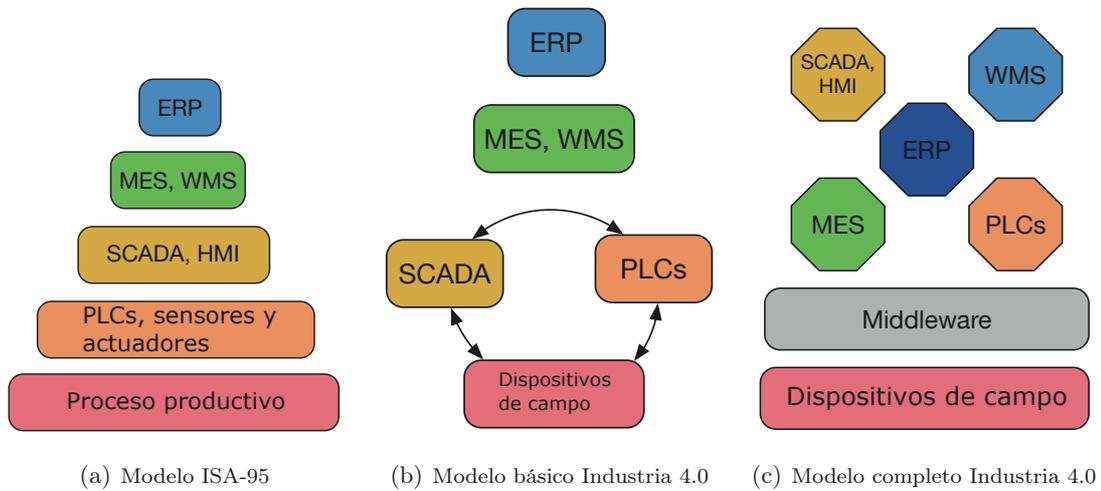


Figura A.1: Evolución de la arquitectura industrial tradicional basada en el estándar ISA-95 y evolución hacia la Industria 4.0

orientadas a servicios, que combinarán componentes de IT y OT accesibles a través de interfaces comunes desde todos los procesos y entidades de la organización, logrando así la digitalización de la red y el modelo descentralizado que persigue la industria del futuro.

Aunque el ecosistema de la Industria 4.0 es especialmente variado y está sujeto a varias de estas iniciativas, se puede definir este paradigma desde una perspectiva técnica como la combinación de procesos productivos con tecnologías punteras de la información y las comunicaciones. Tal como aparece ilustrado en la Figura A.1, el objetivo es hacer evolucionar el modelo rígido de industria tradicional basado el estándar ISA-95 [36]. Este modelo está compuesto por cinco niveles en forma de pirámide, donde en la base se encuentra el proceso productivo y por encima los dispositivos de control y los sistemas que controlan el flujo de trabajo, como Sistemas de Ejecución de Manufactura (MES). Por último, la cúspide contiene la infraestructura de logística, inventario y Planificación de Recursos Empresariales (ERP). Al cambiar esta pirámide hacia un modelo que proporcione una infraestructura descentralizada más dinámica y reconfigurable (como las mostradas en la Figura A.1), se potencia la creación de nuevos servicios optimizando los ya existentes [38], propiciando una mayor productividad con una reducción de costes.

Desde un punto de vista más técnico, todas estas ventajas de la Industria 4.0 se pueden conseguir mediante un conjunto de tecnologías que en esta tesis anticiparemos y resumiremos en torno a cinco áreas distintas:

- **Internet de las Cosas Industrial (IIoT):** con objeto de integrar verticalmente todos los componentes de la arquitectura, desde los sistemas de control hasta las máquinas o los propios productos. A este paradigma se le unen otros modelos de computación en la periferia de la red, como el Mobile Edge Computing (MEC) o el fog computing.

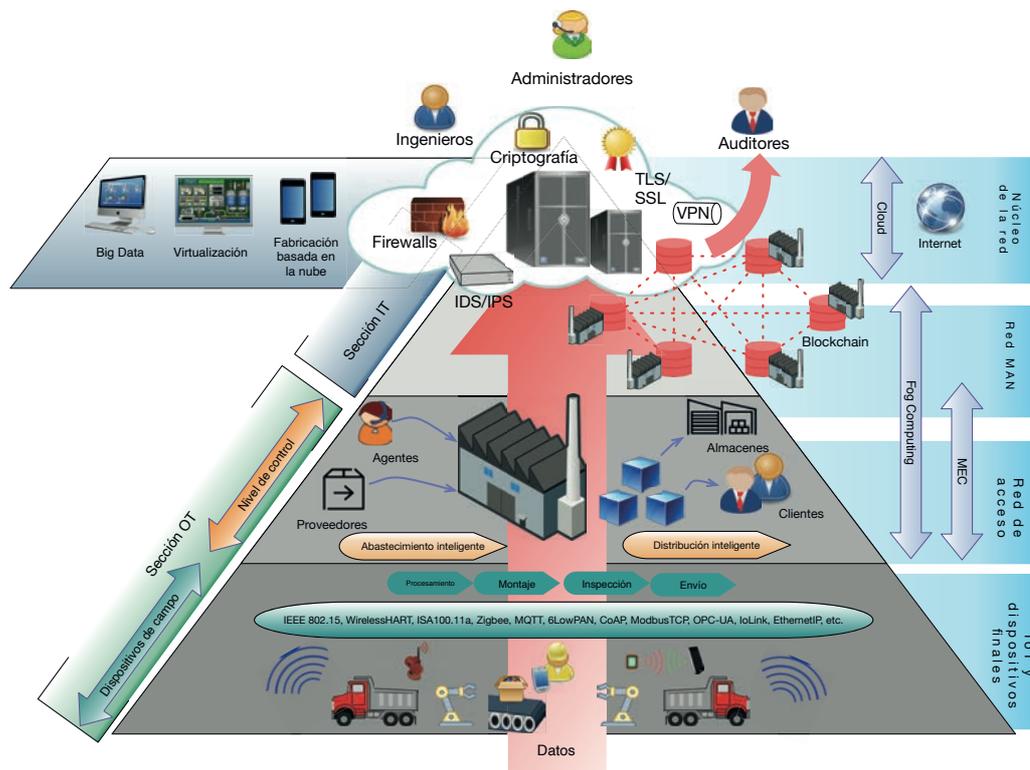


Figura A.2: Visión general del modelo de infraestructura de la Industria 4.0 y sus tecnologías asociadas

- **Cloud computing:** constituye un pilar fundamental para realizar procedimientos con los datos recogidos de el sistema productivo, así como el despliegue de servicios entre clientes o proveedores, en lo que se conoce como fabricación basada en la nube [41].
- **Big Data:** abarca el análisis de toda la información proporcionada por las entidades del ecosistema industrial, buscando servicios de valor añadido como la supervisión del funcionamiento de las entidades del ecosistema, la optimización de los procesos y la identificación de anomalías.
- **Blockchain:** constituye un sistema transparente y seguro para almacenar datos, y que permite una gran cantidad de aplicaciones empresariales, que van desde el comercio Peer-to-Peer (P2P) con energía eléctrica en microrredes [43] hasta históricos de registros a prueba de manipulaciones [44].
- **Virtualización:** supone un conjunto de tecnologías que permiten la representación virtual de todas las máquinas y componentes que intervienen en el proceso de producción (también denominados como "gemelos digitales"), para realizar simulaciones de cara a prevenir fallos y optimizar la línea de producción.

Una visión global de la infraestructura de la Industria 4.0 con la integración de todas estas tecnologías se ilustra en la Figura A.2. En conjunto, estas tecnologías permitirán a la industria modelar de forma flexible las operaciones que se realizan dentro del ciclo de vida de la producción.

Sin embargo, la desventaja principal de estos entornos reside en la ciberseguridad. Y es que una infraestructura tan tecnológicamente heterogénea como esta incrementa la probabilidad de exponerse a nuevas amenazas [67] que operan bajo diferentes modos de ataque [68] que no han sido abordados anteriormente, poniendo en riesgo todos los servicios de seguridad. Desde el punto de vista de la disponibilidad, sería posible lanzar un ataque de denegación de servicio (DoS) desde cualquier elemento de la organización. Desde la perspectiva de la integridad, la manipulación de las tecnologías de la Industria 4.0 puede permitir a un adversario manipular no sólo el comportamiento local, sino también el comportamiento global a través de procesos de decisión distribuidos y cooperativos. A nivel de confidencialidad, la cantidad de información sensible gestionada por las entidades conectadas aumentará, con el consiguiente incremento del riesgo y el impacto de los ataques. De igual modo, la autenticación se resiente a medida que se difuminan las barreras entre los distintos subsistemas y se integran tecnologías como el Big Data y la virtualización. Por último, la privacidad también está en riesgo, tanto a nivel humano como a nivel de las propias empresas del sector industrial.

Como resultado, un sistema industrial de la Industria 4.0 es considerado cada vez más complejo y crítico, y puede ser objetivo de múltiples vectores de ataque que pueden ser finalmente aprovechados para perpetrar una APT [69, 70]. Estamos hablando de ataques sofisticados perpetrados por un adversario experto, y se caracterizan por su capacidad de pasar desapercibidos dentro de la red de la víctima durante un cierto período de tiempo. Estos incidentes degeneran en pérdidas millonarias para organizaciones industriales de todos los sectores, tal como ponen de relieve numerosos informes. Un ejemplo es el realizado por la entidad Accenture [2], que aparece reflejado en la Figura A.3. En ella se muestran los costes derivados de las brechas de seguridad en el año 2018, en un estudio donde consultado a 254 grandes industrias. Como resultado, se estimó una media de 13 millones de dólares en costes, consecuencia de un aumento del 12% en los incidentes de seguridad con respecto al año anterior, afectando especialmente a las infraestructuras críticas del sector financiero, como se muestra en el gráfico.

Debido a la complejidad y el impacto de estos ataques, es crucial entender cuál es el verdadero alcance y las capacidades de detección de la primera línea de defensa ante APT; en otras palabras, los Sistemas de Detección de Intrusiones (IDS) existentes en la actualidad. La motivación de nuestra investigación aparece al explorar el estado del arte y concluir que aún hay cuestiones que necesitan ser abordadas para desarrollar herramientas eficaces capaces de detectar, rastrear y disuadir APT en estos entornos. En primer lugar, muy pocos trabajos hacen uso de la investigación existente sobre el comportamiento de las APT [71, 74] para validar sus mecanismos de detección. Por otro lado, aunque existan sistemas específicos con capacidad para detectar gran parte de los vectores de ataque de una APT por separado, no hay única solución que pueda hacer frente a todas las amenazas potenciales. Por este motivo, es interesante estudiar la integración de

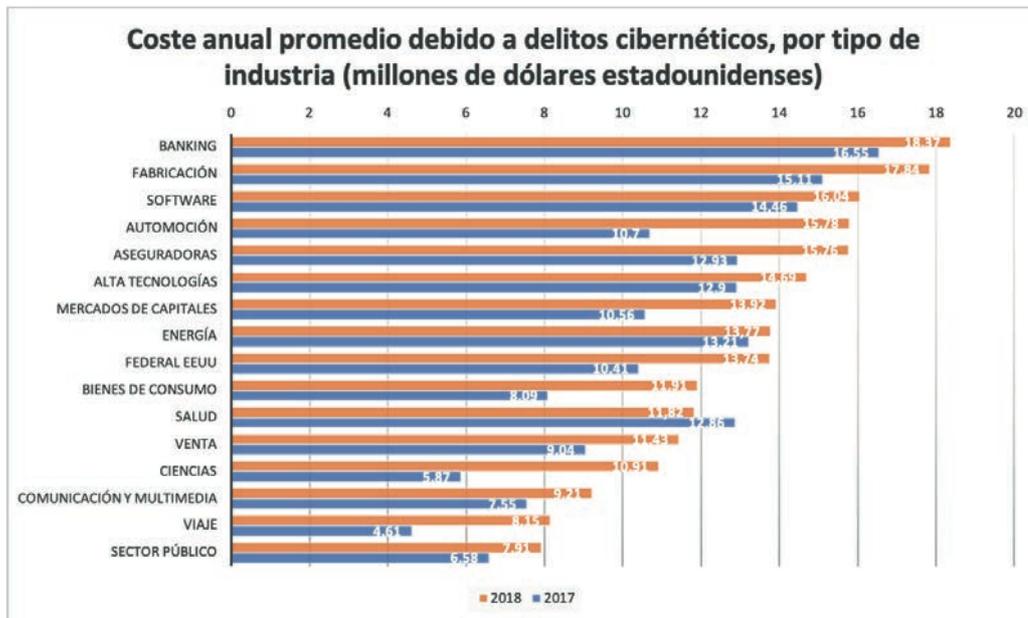


Figura A.3: Promedio del coste generado por el incremento de brechas de seguridad en 2018 en 254 empresas consultadas internacionalmente [2]

soluciones de defensa holísticas en las infraestructuras críticas existentes, no solo en términos de detección, sino también en términos de usabilidad. Por ejemplo, la disponibilidad de herramientas para facilitar la trazabilidad de posibles APT y la formación de los usuarios [81]. Esto último es extremadamente relevante en redes especialmente cambiantes donde se integran tecnologías de vanguardia como las citadas anteriormente, y a medida que también aumenta el volumen de información accesible para el atacante.

En base a esta problemática, en esta tesis se aborda el diseño de un marco de trabajo para la detección y trazabilidad de APT en entornos y aplicaciones de la Industria 4.0. Su objetivo es cubrir el hueco existente entre los sistemas de detección clásicos y los requisitos impuestos por los APT. La premisa es combinar mecanismos capaces de monitorizar todos los dispositivos y procesos que están interconectados dentro de la organización, recuperar datos sobre la cadena de producción a todos los niveles (alarmas, registros de red, tráfico en bruto, etc.) y correlacionar los eventos de forma distribuida para rastrear las etapas de uno de estos ataques a lo largo de todo su ciclo de vida. Estas medidas proporcionarían la capacidad de detectar y anticipar de forma holística los APT, así como los fallos, de manera oportuna y autónoma, a fin de impedir la propagación del ataque y minimizar su impacto.

Para hacer frente a estos objetivos de ciberseguridad, el marco de trabajo extrae los avances más importantes de las soluciones más novedosas del estado del arte en la Industria 4.0, como el algoritmo de Opinion Dynamics [82]. Estas técnicas se basan en algoritmos avanzados de correlación que analizan una red industrial de manera distribuida, aprovechando mecanismos de minería de datos y aprendizaje automático. En conjunto, el marco resultante sirve de guía para

el diseño y desarrollo futuro de sistemas de detección avanzados que cumplan con un conjunto de requisitos de seguridad y detección alineados con los avances tecnológicos experimentados en estos entornos.

Podemos resumir las contribuciones de esta tesis en torno a los siguientes puntos, abordados en capítulos independientes del documento:

- En primer lugar, estudiamos los problemas de seguridad que amenazan a las arquitecturas industriales actuales. El objetivo es caracterizar el contexto y crear una taxonomía de ataques que pueden formar parte de una APT contra los activos industriales actuales y los futuros despliegues de la Industria 4.0.
- A continuación, realizamos un análisis de la evolución y aplicabilidad de los IDSs que se han propuesto tanto en la industria como en el ámbito académico. De este modo, identificamos las áreas que necesitan más investigación, en cuanto a la aplicabilidad e integración de mecanismos de detección proactiva y su integración en la Industria 4.0.
- A partir de los requisitos de seguridad y detección extraídos, definimos un marco de trabajo formal para el diseño de soluciones que permitan la correlación distribuida de eventos provocados por una APT. Este marco considera diversas arquitecturas de red, tipos de ataque y modelos de adquisición de datos, para posteriormente definir las entradas y salidas que deben incluir las soluciones de trazabilidad para cumplir con los requisitos mencionados. De este modo, sentamos las bases para el desarrollo y la comparación de nuevas soluciones en este contexto.
- Como medio para validar el marco propuesto, definimos dos mecanismos de protección basados en *clustering* y consenso distribuido. Tras esto llevamos a cabo diferentes experimentos con objeto de comparar su precisión a la hora de rastrear diferentes APT, basándonos en modelos de ataque realistas creados a partir del análisis de amenazas anterior.
- Posteriormente, evaluamos la efectividad de estos mecanismos para el despliegue de técnicas de respuesta y su aplicabilidad en varios escenarios de la Industria 4.0, siendo los más relevantes la Smart Grid (para desplegar mecanismos que garanticen la seguridad de la red y sus sistemas de autorización) y el Internet de las Cosas Industrial.
- Realizamos la verificación y validación del marco de trabajo definido, los algoritmos de correlación y las técnicas de respuesta desarrolladas. Para ello, empleamos distintas demostraciones teóricas además de un prototipo real en un banco de pruebas industrial.
- Por último, identificamos algunas cuestiones de investigación abiertas en esta tesis doctoral y comentamos algunas líneas de trabajo futuro.

Tabla A.1: Principales amenazas de ciberseguridad de las tecnologías de la Industria 4.0

	<b>IIoT</b>	<b>Cloud/fog</b>	<b>Big Data</b>	<b>Virtualización</b>	<b>Blockchain</b>
Disponibilidad	Agotamiento de recursos	Inundación de red, robo de servicio	Múltiples puntos de fallo	Múltiples puntos de fallo	Inundación de red, manipulación del consenso
Confidencialidad	Exposición de información sensible	Acceso a datos por el proveedor de servicios, ataques side-channel	Falta de medidas criptográficas, problemas de privacidad con el análisis masivo	Fuga de información en simulaciones	Privacidad con transacciones rastreables a los usuarios
Integridad	Manipulación de los datos o del encaminamiento	Máquinas virtuales maliciosas	Servidores no confiables, ausencia de medidas de integridad	Disparidad entre los parámetros físicos y virtuales	Contratos inteligentes vulnerables, inyección de código
Autenticación	Apropiación de identidad	Phishing	Falta de controles de acceso de grano fino a nodos y tablas	Falta de servicios AAA para acceder a los datos de dispositivos heterogéneos	Robo de identidad o de nodos, ataques Sybil

## A.2 Amenazas de ciberseguridad en la Industria 4.0

Para comenzar a adentrarnos en la problemática de la trazabilidad de las APT, en el Capítulo 2 estudiamos las amenazas de ciberseguridad a los que se enfrentan las arquitecturas industriales actuales. Primero, revisamos las amenazas que afectan a los sistemas industriales tradicionales, para luego analizar los problemas de ciberseguridad que presentan las tecnologías que permiten la Industria 4.0, tanto por separado como en los propios servicios que ofrece este paradigma. Tras ello, estudiamos cómo estos vectores de ataque pueden formar parte de una amenaza persistente avanzada, con el objetivo de exponer los retos a los que se enfrentan las soluciones de detección actuales.

Para llevar a cabo nuestro análisis, clasificamos las amenazas identificadas en la bibliografía e informes específicos según la taxonomía recogida por el estándar RFC 7416 del IETF [89], agrupándolas en función de los servicios de seguridad [90] que son objetivo del ataque: la *disponibilidad*, la *integridad*, la *confidencialidad* y la *autenticación*.

Comenzando por los sistemas tradicionales, cabe mencionar los ataques de Denegación de Servicio Distribuidos (DDoS, del acrónimo en inglés) como principal amenaza contra la disponibilidad. En términos de integridad, incluimos desde el sabotaje físico de los equipos industriales hasta la inyección de malware, con objeto de alterar los protocolos de comunicación industrial y/o los

valores reales de tráfico producidos por los dispositivos de campo, los controladores o los equipos de la red corporativa. En cuanto a confidencialidad, es preciso destacar el robo y la divulgación de información sensible del proceso industrial en cuestión, a menudo a través de ataques de cross-site scripting (XSS) o SQL Injection contra páginas web. Por último, la autenticación estaría en riesgo con vulnerabilidades en el software que permiten obtener un acceso no autorizado a los recursos o una escalada de privilegios. Esto es frecuente cuando se combinan técnicas de ingeniería social (por ejemplo, ataques de *phishing* o correos de spam) para recoger información estratégica del sistema. A esto también se le une la fácil movilidad de los operarios en la planta y sus interacciones mediante el uso de interfaces muy diversas (como teléfonos inteligentes, tabletas u ordenadores portátiles) que también provocan problemas de seguridad causados por una mala configuración o un control de acceso inadecuado, tanto a nivel lógico (uso de contraseñas simples) como físico (acceso a los equipos).

En cuanto a las tecnologías que dan pie a la Industria 4.0 actual, un resumen de las principales amenazas halladas queda recogido en la Tabla A.1. En ella aparecen clasificadas en torno a los citados servicios de seguridad para las cinco tecnologías identificadas en la introducción de la tesis doctoral: el IIoT, el Big Data, el *cloud computing* y la *blockchain*.

No obstante, estas tan solo son las amenazas que afectan a las principales tecnologías habilitadoras de la Industria 4.0 por separado. Como se ha mencionado ya, nuestro objetivo a continuación es revisar cuáles pueden afectar a los servicios de los ecosistemas industriales de la Industria 4.0 en su conjunto. La razón es sencilla: si bien estos servicios heredan las amenazas de sus tecnologías habilitadoras, también existen diversas amenazas novedosas que surgen debido a sus características particulares. Para este análisis, cuyos resultados se han obtenido a través de una revisión experta del estado del arte disponible de la Industria 4.0, también hemos seguido el estándar IETF RFC 7416 [89] para catalogar las amenazas. Tales servicios o mejoras quedan resumidos de la siguiente manera, junto con sus amenazas principales:

**Nuevas infraestructuras.** La transición gradual hacia arquitecturas más descentralizadas está trayendo consigo un entorno más heterogéneo y complejo, donde todos los elementos pueden interactuar y cooperar entre sí. Desde el punto de vista de la *disponibilidad*, esta transición significa que no sólo un insider malicioso podría atacar cualquier elemento, sino que también podría lanzarse un ataque DoS desde cualquier elemento de la infraestructura. En términos de *integridad*, debemos considerar que un adversario puede alterar el comportamiento global (por ejemplo, los flujos de trabajo de los procesos) manipulando los sistemas locales. Esto está relacionado con los problemas de *confidencialidad*, donde los ataques maliciosos contra las entidades locales podrían exponer el comportamiento a nivel global. Por último, en lo que respecta a las amenazas de *autenticación*, al difuminarse las barreras entre los distintos subsistemas es indispensable desplegar políticas de seguridad adecuadas que puedan limitar los daños causados por los accesos no autorizados. No obstante, la complejidad de esas políticas probablemente dan lugar a sistemas mal configurados, que igualmente pueden ser explotados por un adversario.

**Modernización de maquinaria o ‘retrofitting’.** Es posible llevar los beneficios de la Industria 4.0 a los sistemas heredados desplegando y conectando las nuevas tecnologías a los subsistemas más antiguos [113]. Aun así, estos despliegues traen consigo problemas de seguridad adicionales a considerar. La existencia de un subsistema paralelo (por ejemplo, un sistema de monitorización) puede traer consigo ciertos problemas de *disponibilidad* e *integridad*: no sólo los componentes que sirven de puente entre lo antiguo y lo nuevo pueden convertirse en un punto de fallo, sino que también las nuevas tecnologías podrían utilizarse para lanzar ataques contra esos elementos heredados (comúnmente conocidos como *legacy*). Además, también existen amenazas de *confidencialidad*, ya que las nuevas tecnologías suelen actuar como una “capa de detección” que puede exponer información sobre el estado y el comportamiento de los procesos industriales monitorizados. En cuanto al impacto de las amenazas de *autenticación*, depende del grado de integración de los nuevos subsistemas. En este sentido, sus interfaces pueden limitar la cantidad de información que puede recuperarse de los subcomponentes internos.

**Espacio de datos compartido.** Uno de los objetivos de la Industria 4.0 es crear espacios comunes para el intercambio seguro de información entre socios industriales [114]. La creación de estos espacios cooperativos podría traer consigo amenazas adicionales desde el punto de vista de la *disponibilidad* y la *integridad*: en particular, la existencia de ataques DoS que interrumpan el flujo de información en momentos críticos, o de componentes comprometidos de proveedores maliciosos que afecten a otros elementos, abriendo la puerta a posibles efectos en cascada dentro de la cadena de suministro. Por otra parte, la *confidencialidad* también es especialmente importante en este contexto de cara a asegurar que la información intercambiada por los socios no facilita la extracción de datos para la competencia. Aun así, configuraciones anómalas y ataques internos podrían abrir la puerta a fugas de información más graves. Por último, las amenazas de autenticación también se agravan en este espacio de cooperación, ya que los accesos no autorizados pueden tener un mayor impacto para la extracción de información valiosa.

**Fabricación en la nube.** Otro de los principios de este paradigma es la creación de aplicaciones industriales basadas en el *cloud* que aprovechan recursos de fabricación distribuidos [115]. Esta distribución de recursos crea ciertas amenazas que ya se han descrito en el contexto de las nuevas infraestructuras digitales: desde ataques DoS que pueden lanzarse desde cualquier lugar (*disponibilidad*), hasta la manipulación de los componentes distribuidos (*integridad*). La principal diferencia aquí es la naturaleza de estas amenazas: máquinas virtuales maliciosas, ataques DoS contra los servidores de la nube o la conexión de red, etc. De igual modo, las amenazas contra la confidencialidad también se vuelven más críticas, ya que la infraestructura de la nube puede contener información sensible sobre procesos empresariales que pueden quedar expuestos al proveedor de servicios *cloud*. Por último, la complejidad en la gestión de este tipo de infraestructuras basadas en la nube también abre más oportunidades para los ataques de *autenticación*.

Tabla A.2: Amenazas de ciberseguridad de los servicios de la Industria 4.0

	Nuevas infr.	Retrofitting	Esp. Datos comp.	Fab. en la nube	Agentes	Inter. avanzadas
Disponibilidad	Amplia superficie de ataques	Único punto de fallo	Efectos en cascada	Amplia superficie de ataques	Agentes malware	Denegación de servicio
Confidencialidad	Acceso a datos compartidos	Exposición de datos sensibles	Fuga de información	Fuga de procesos empresariales	Datos del agente en el contexto local	Fuga de información
Integridad	Manipulación del comportamiento	Ataques transversales	Efectos en cascada	Manipulación de componentes	Datos o agentes manipulados	Manipulación de la toma de decisiones
Autenticación	Complejidad y configuración errónea	Sistemas legacy ilegítimos	Mayor alcance de los ataques	Complicada gestión de credenciales	Agentes atacados o atacantes	Escalada de privilegios

**Agentes.** Ya existen varias pruebas de concepto relacionadas con la integración de los llamados agentes software en los procesos de fabricación, como los planificadores del flujo de trabajo o los sistemas de montaje autoorganizados [116]. No obstante, también hay peligros asociados al despliegue de agentes en un entorno industrial genérico. Y es que un agente malicioso puede comportarse como una pieza de malware, afectando a la *capacidad* de otros elementos industriales. Además de la *integridad* de los propios agentes, también hay que considerar cómo otros elementos manipulados pueden ejercer una influencia (in)directa sobre el comportamiento de los mismos. Por ejemplo, manipulando el entorno que rodea al agente o incluso el propio agente, de manera que sea posible lanzar ataques de *confidencialidad* para extraer el flujo de información que este procesa, lo que se ve agravado en escenarios sin una infraestructura adecuada de *autenticación*.

**Interacciones avanzadas.** Como se ha comentado ya, las tecnologías facilitadoras de la Industria 4.0 relacionadas con la virtualización permiten la creación de servicios novedosos como los "gemelos digitales" (representaciones virtuales de subsistemas) y los "trabajadores digitales", que proporcionan interacción con Interfaces Hombre-Máquina (HMI) avanzadas. Sin embargo, también hay ciertas amenazas relacionadas con el uso real de tales tecnologías. Estos sistemas pueden ser manipulados por operadores humanos, aumentando el daño causado por una persona que posea información privilegiada. Como ejemplo, un trabajador digital malintencionado podría llevar a cabo varios ataques, como el lanzamiento de ataques de denegación de servicio (*disponibilidad*), la interferencia en los procesos de toma de decisiones (*integridad*), la extracción de información confidencial (*confidencialidad*) y la ejecución de ataques de escalada de privilegios (*autenticación*). Por otro lado, estos sistemas mejorados pueden convertirse ellos mismos en atacantes, causando daños de forma sutil. Ejemplo de ello sería el de un atacante manipulando un HMI para que obligue al trabajador a realizar una acción incorrecta, para culpabilizarlo a continuación.

Por su parte, las principales amenazas que atentan a cada uno de estos servicios aparecen descritas en la Tabla A.2. En conjunto, muchos de estos vectores de ataque (tanto de las amenazas tradicionales como de las futuras en los sistemas industriales) pueden ponerse en práctica en las

denominadas APT o amenazas persistentes avanzadas. Tal como se introdujo con anterioridad, estamos hablando ataques sofisticados perpetrados contra una organización concreta, en la que los atacantes tienen una experiencia y unos recursos importantes. Estos atacantes se infiltran en las redes de las víctimas aprovechando una multitud de vulnerabilidades (a menudo desconocidas, es decir, de día cero), y pasan desapercibidos durante un periodo de tiempo prolongado [69, 70]. Si bien en un principio las APT atacaban a organizaciones militares, en la actualidad se dirigen a un amplio abanico de industrias y gobiernos con el objetivo de apoderarse de los sistemas de control y causar daños o extraer información suculenta.

Stuxnet fue el primer ataque de este tipo, denunciado en 2010, que sabotó el programa nuclear iraní causando daños físicos en la infraestructura y ralentizando el proceso global durante cuatro años. Desde entonces, el número de incidentes de este tipo ha aumentado drásticamente, agravando el problema en entornos de la Industria 4.0, que favorecen la convergencia de todo tipo de infraestructuras IT/OT y, por tanto, amplían la superficie de ataque de las infraestructuras críticas. Por este motivo, y para poner en marcha técnicas de defensa precisas en este contexto, es necesario estudiar cómo afectan las APT a la detección de anomalías y la aplicación de soluciones de trazabilidad. Para tal fin, a continuación repasamos algunos de las APT más importantes de los que se ha informado en los últimos años, con objeto de definir un modelo de atacante que se ajuste al comportamiento de este tipo de amenazas.

De ese estudio extraemos una plantilla de APT compuesta por distintas fases de ataque que son comúnmente aplicadas por el atacante, y que al mismo tiempo han sido ampliamente estudiadas y descritas por varios investigadores del ámbito académico e industrial [74, 70, 129]. A continuación, resumimos tales etapas:

- **Reconocimiento (R).** El adversario reúne información sobre la red objetivo para encontrar vulnerabilidades explotables y crear un plan de ataque que penetre sus defensas.
- **Envío.** Después de elegir un conjunto de nodos vulnerables (el llamado "paciente cero"), el atacante establece una comunicación (C) con la red y envía el malware a esos ordenadores, ya sea directamente (por ejemplo, a través de correos electrónicos de spear phishing o servicios vulnerables) o indirectamente (por ejemplo, contaminando los sitios web de un tercero) [130].
- **Ejecución.** El malware se ejecuta (E) en la máquina objetivo y toma el control de la misma, efectuándose la primera intrusión en la red. Esta etapa implica varios pasos, como el *escalada de privilegios*, el mantenimiento de la *persistencia* y la ejecución de *técnicas evasivas*.
- **Command and Control.** Una vez que el malware controla el "paciente cero", abre un canal de comunicación con los dominios del atacante instalando puertas traseras, que serán utilizadas para ejecutar comandos, extraer información, etc. de manera remota. Esta fase

puede incluir el Rastreo (T) de las vulnerabilidades de día cero en base a la información recogida por el adversario.

- **Movimiento lateral.** Engloba los diferentes pasos que da el malware para conseguir la propagación (P) del ataque a otras zonas de la red. Esta etapa incluye el *reconocimiento interno*, el *control* de sistemas adicionales y la *recogida de información sensible*.
- **Ejecución final (F).** El malware finalmente realiza el ataque contra la red industrial objetivo. Esto incluye la *exfiltración* de datos sensibles o la *destrucción* de recursos.

Una vez tenemos visión completa de las amenazas presentes y futuras a las que se enfrenta un sistema industrial, podemos relacionarlas con las etapas de una APT introducidas anteriormente, tal como se ilustra en la Tabla A.3, donde también se especifica el impacto sobre cada uno de los segmentos de la red industrial. Como se puede apreciar, la explotación de estas amenazas puede darse en múltiples etapas de una amenaza persistente avanzada. Más concretamente, podemos observar que la mayoría de las amenazas pueden ser potencialmente aprovechadas para la primera intrusión y la posterior ejecución de exploits. Sin embargo, la recopilación inicial de información sobre los puntos de entrada y las vulnerabilidades se realiza principalmente mediante el análisis de los metadatos que emanan de los servidores a los sensores, y también mediante la ingeniería social. En cuanto a la exfiltración final de información, normalmente se requiere que el atacante se haya apoderado del dispositivo para enviar datos de forma que se asemejen al tráfico de red normal, lo que dificulta cualquier intento de detección. Esta información es especialmente útil para idear nuevas soluciones de defensa y adaptar los actuales mecanismos de detección, tal como abordamos a continuación.

Tabla A.3: Visión general de las amenazas de los sistemas industriales y relación con las etapas de una APT

Amenazas	Tradicional	IIoT	Cloud Comp.	Big Data	Block.	Virtual.	fases APT	Impacto en				
								Proceso productivo	Red IT	Usuarios finales		
<b>Disponibilidad</b>												
Substracción de dispositivos	✓	✓				✓	E	✓			✓	
Ataques DDoS	✓	✓	✓	✓	✓	✓	C, E, P	✓			✓	
Ataques a la ruta de información	✓	✓			✓		C, E, T, F, P	✓			✓	
Agotamiento de recursos	✓	✓			✓		C, E	✓			✓	
Robo de servicio			✓	✓	✓	✓	C, E	✓			✓	
<b>Integrity</b>												
Configuración incorrecta	✓	✓	✓	✓	✓	✓	C, E	✓			✓	
Ingeniería inversa	✓	✓	✓	✓	✓	✓	R, C, P, E, T, F	✓			✓	
y/o inyección de malware												
Inyección de datos falsos	✓	✓		✓	✓	✓	C, E, P	✓			✓	
Spoofing	✓	✓		✓	✓	✓	C, E	✓			✓	
Manipulación de encaminamiento	✓	✓		✓	✓	✓	C, E, P	✓			✓	
<b>Confidencialidad</b>												
Robo de información sensible	✓	✓	✓	✓	✓	✓	C, E, F	✓			✓	
Exposición del estado del nodo (ataques side-channel)		✓	✓			✓	R, C, E, F	✓			✓	
Análisis pasivo del tráfico	✓	✓		✓		✓	R, C, E, T, F, P	✓			✓	
Exposición de la infraestructura (sistemas de memoria compartida)			✓		✓	✓	C, E, T, F, P	✓			✓	
<b>AAA</b>												
Escalada de privilegios	✓	✓	✓	✓		✓	C, E, P	✓			✓	
Ingeniería social	✓		✓	✓	✓	✓	R, C, E	✓			✓	
Control de acceso deficiente	✓	✓	✓	✓	✓	✓	C, E	✓			✓	
Suplantación de nodos (nodos fake/dummy)	✓	✓		✓	✓	✓	C, E	✓			✓	

## A.3 Servicios de detección en los sistemas de control modernos

Después de estudiar con detenimiento las amenazas que puede aprovechar una APT en sus fases de ejecución, concluimos que es necesario combinar múltiples soluciones de seguridad a diferentes niveles debido a la variedad de vectores de ataque que pueden utilizar. Por esta razón, el Capítulo 3 de esta tesis doctoral lo dedicamos a analizar los mecanismos de detección que se pueden aplicar como primera línea de defensa. Para ello, analizamos los IDS disponibles comercialmente y en el ámbito académico, bajo la premisa de identificar las áreas que necesitan de mayor investigación. A partir de los conocimientos extraídos en este estudio, definiremos el marco de trabajo que nos permita desarrollar soluciones de trazabilidad ante APT, lo que representa el núcleo de nuestro trabajo.

Debido al amplio espectro de soluciones disponibles en el terreno de los IDS, comenzamos en primera instancia clasificándolos en función del método empleado para la detección. Una posibilidad es el IDS *basado en firmas*, que trata de encontrar patrones específicos en las tramas transmitidas por la red. Sin embargo, precisamente por eso les resulta imposible detectar nuevos tipos de ataques cuyo patrón es desconocido [132]. Otra posibilidad son los IDS *basados en anomalías*, que compara el estado actual del sistema y sus datos generados con el comportamiento normal del sistema, para identificar las desviaciones presentes cuando se produce una intrusión. En este punto, hay que tener en cuenta restricciones como la heterogeneidad de los datos recogidos en un entorno industrial, el ruido presente en las mediciones y la naturaleza de las anomalías (con objeto de distinguir ataques frente a fallos no intencionados).

Dentro de esta categoría se han desarrollado numerosas técnicas de detección basadas en áreas como la estadística o la inteligencia artificial [133], cada una con un nivel de adaptación diferente en función del escenario de la aplicación a proteger [134]. A continuación, se reseña cada una de ellas:

- **Detección basada en minería de datos:** se analiza una enorme cantidad de información en busca de características que permitan distinguir si los datos son anómalos. En esta clase encontramos técnicas de clasificación, basadas en *clustering* o reglas de asociación.
- **Detección estadística de anomalías:** con este enfoque encontramos pruebas de inferencia para verificar si un dato se ajusta o no a un modelo estadístico determinado, con el fin de confirmar la existencia de intrusiones. Bajo esta categoría cabe mencionar técnicas basadas en series temporales, cadenas de Markov o teoría espectral.
- **Detección basada en el conocimiento:** en este caso, la información sobre ataques o vulnerabilidades específicas se adquiere de forma progresiva y se almacena en una base de conocimiento. Ejemplos de estas técnicas son las redes de *Petri* o los sistemas de expertos.
- **Detección basada en aprendizaje automático (*machine learning*):** este tipo de técnicas basan la detección en la creación de un modelo matemático que aprende y mejora

su precisión a medida que adquiere información sobre el sistema. En esta categoría encontramos técnicas de inteligencia artificial cuyos fundamentos están también muy ligados a la estadística y a la minería de datos, como pueden ser redes neuronales, redes bayesianas, lógica difusa, etc.

Por último, también existen IDS *basados en especificación* [135]. El principio es similar al de los sistemas basados en anomalías, ya que el estado actual del sistema se compara con un modelo existente. Sin embargo, en este caso las especificaciones son definidas manualmente por expertos, lo que reduce el número de falsos positivos cuando se definen con detalle. A menudo se utilizan diagramas de estado, autómatas finitos, métodos formales, etc. y pueden combinarse con IDS basados en firmas y en anomalías.

Además del tipo de método empleado para realizar la detección, también es posible clasificar los IDS según su cobertura de detección para las distintas secciones de una red industrial (ya sea enfocados en los dispositivos de campo, los dispositivos de control o la red corporativa), según la arquitectura de red especificada por el estándar ISA-95 [36] y reflejada en la Figura A.2. De igual manera, también existen sistemas IDS especializados en el análisis de tráfico proveniente de protocolos de comunicación específicos, lo que acota su grado de aplicación en entornos especialmente heterogéneos.

Esta taxonomía tan diversa es palpable tras llevar a cabo el análisis del estado del arte de soluciones IDS existentes en la bibliografía, que queda plasmado en las Tablas A.4, A.5 y A.6. Estas ofrecen una clasificación por categorías (según la cobertura de detección, el protocolo analizado y el mecanismo de detección, respectivamente) del número de artículos más relevantes publicados en revistas y/o congresos internacionales en este campo entre los años 2013 y 2020. Parte de este análisis se lleva a cabo como parte de SADCIP [131], un proyecto de investigación financiado por el Ministerio de Economía, Industria y Competitividad. El proyecto gira en torno al desarrollo de sistemas de detección avanzados capaces de hacer frente a amenazas sofisticadas de los ecosistemas industriales modernos, considerando las características específicas de la Industria 4.0.

Cobertura	2013	2014	2015	2016	2017	2018	2019	2020
Dispositivos de campo	2	-	3	15	9	8	10	6
Dispositivos de control	4	8	9	5	9	9	10	5
Sistemas de control	1	3	3	9	17	12	18	11
Sistema completo	-	1	-	5	2	6	9	7

Tabla A.4: Evolución de los IDS según su cobertura de detección

De este profundo análisis podemos extraer varias conclusiones interesantes. De cara a nuestra investigación, cabe enfatizar el incremento del interés por los sistemas IDS basados en algún tipo de aprendizaje automático, así como el auge de mecanismos que pretenden abordar la detección de anomalías para un sistema industrial al completo, sin restringirse a dispositivos concretos. Esto se consigue desplegando varios componentes de detección, tanto hardware como software,

### A.3. Servicios de detección en los sistemas de control modernos

Protocolo	2013	2014	2015	2016	2017	2018	2019	2020
Protocolos de bus de campo	2	1	2	3	2	2	4	2
Protocolos de comunicación	2	3	10	14	8	8	9	7
Protocolos de control y gestión	1	-	1	1	1	2	3	2

Tabla A.5: Evolución de los IDS según el protocolo analizado

Mecanismo	2013	2014	2015	2016	2017	2018	2019	2020
Detección basada en firmas	-	3	-	4	5	6	4	5
Minería de datos	2	2	4	5	6	7	10	6
Detección estadística	-	-	4	5	3	2	4	3
Detección basada en el conocimiento	1	1	2	1	-	4	5	4
Aprendizaje automático	3	3	2	8	9	9	11	13
Detección basada en especificación	1	3	2	8	10	4	7	4
Otros mecanismos	-	-	3	5	5	4	9	7

Tabla A.6: Evolución de los IDS según su mecanismo de detección

que obtienen información y la procesan a nivel local para todas los procesos de la infraestructura. Esta información se enviará después a un sistema central, que puede detectar de forma más eficiente las amenazas que afectan a varios elementos del sistema de forma encubierta [176]. Por ejemplo, algunas arquitecturas permiten que los dispositivos de campo estén totalmente monitorizados junto a todos los demás elementos del sistema de control [170], mientras que otras arquitecturas mejoran la detección de anomalías cuyo impacto se distribuye a todos los elementos del sistema [177]. También hay arquitecturas, como [178][179], que dividen el sistema global en varias particiones lógicas, con el fin de facilitar el trabajo de los sistemas de detección de anomalías. Por último, algunas arquitecturas despliegan agentes que están específicamente diseñados para buscar infecciones de malware provenientes de APT [180].

En particular, estos sistemas nos sirven de gran inspiración para el desarrollo de técnicas de detección holística y para la trazabilidad de APT, que es el objetivo del marco propuesto en esta tesis doctoral. El problema aparece al encontrarnos que, independientemente de la estrategia de detección utilizada en las instalaciones industriales o su cobertura, estos IDS solo suponen una primera línea de defensa, y no existe una solución única que nos permita detectar todo tipo de amenazas con precisión. Además, es necesario realizar un análisis posterior de las evidencias generadas (alarmas, eventos de red) y del tráfico en bruto en toda la red para anticiparse a los efectos de ataques sofisticados y persistentes como las APT [227]. Esto se lleva a cabo mediante mecanismos de trazabilidad y correlación avanzada, que proporcionan información del estado de salud general de la red y facilitan el despliegue de medidas de respuesta precisas basadas en la evolución de la amenaza. Aunque esto se ha abordado mayoritariamente en los entornos corporativos tradicionales mediante técnicas proactivas (las evidencias se analizan a medida que se producen los incidentes) o reactivas (las evidencias se estudian una vez que se producen los eventos), aún queda extenso margen de mejora en el campo de los sistemas de control industriales.

En los ecosistemas industriales tradicionales, estas soluciones de trazabilidad se han proporcionado mediante soluciones de conciencia del contexto (o *context awareness*) [232]. Este proceso implica la monitorización constante de los dispositivos para recuperar datos sobre la cadena de producción a todos los niveles (por ejemplo, alarmas, eventos de red, tráfico bruto). Sin embargo, la introducción de topologías cada vez más dinámicas y la creciente gama de ataques extremadamente localizados en la IIoT y la Industria 4.0 complican este proceso de adquisición de información [233]. Por lo tanto, es cada vez más necesario integrar más de una solución de detección para garantizar la máxima cobertura [218]. Además, todas las soluciones deben coexistir bajo una plataforma de detección avanzada que tome la infraestructura desde una perspectiva holística, correlacionando todos los eventos y rastreando todas las amenazas a lo largo del ciclo de vida de una APT [234].

Debido a que el progreso en la Industria 4.0 no ha sido significativo con respecto a estas soluciones de trazabilidad de APT, es nuestra motivación la de proporcionar un primer paso en este área. En este sentido, soluciones como el enfoque de Opinion Dynamics [82] abre el camino a una nueva generación de soluciones basadas en el despliegue de agentes de detección distribuidos por la red. Las anomalías reportadas por estos agentes se correlacionan para extraer conclusiones sobre la secuencia de acciones realizadas por el adversario, así como para identificar las áreas más afectadas de la infraestructura. Esta evaluación puede realizarse en una entidad centralizada o utilizando una arquitectura distribuida de pares [236]. Al mismo tiempo, está abierta la posibilidad de integrar IDSs externos para examinar anomalías en las proximidades de los nodos, así como la abstracción de diversos parámetros como la criticidad de los recursos o la persistencia de los ataques.

A pesar de las numerosas capacidades de esta solución (explicadas en secciones posteriores), es necesario definir un modelo de detección más genérico que asiente las bases para la aplicación de más soluciones de trazabilidad de APT en el paradigma de la Industria 4.0. Y es que las capacidades de Opinion Dynamics pueden implementarse de forma modular, pueden integrarse en otros algoritmos de correlación y cada una de ellos tiene un efecto diferente en diversos aspectos de seguridad, detección, despliegue y eficiencia. En última instancia, esto influye considerablemente en cómo deben desarrollarse y gestionarse los IDS en estos entornos. En base a esto y teniendo en cuenta las características de los IDS actuales, podemos establecer una serie de requisitos de detección y de seguridad sobre los que definir nuestra propuesta.

Por una parte, en cuanto a requisitos de detección contemplamos los siguientes:

- (D<sub>1</sub>) **Cobertura.** Las APT hacen uso de un amplio conjunto de vectores de ataque que ponen en peligro a las organizaciones a todos los niveles. Por ello, el sistema debe ser capaz de asimilar el tráfico y los datos procedentes de dispositivos y secciones heterogéneas de la red, al tiempo que incorpora el input de sistemas de detección externos.
- (D<sub>2</sub>) **Holismo.** Para identificar comportamientos anómalos, el sistema debe ser capaz de procesar todas las interacciones entre usuarios, procesos y salidas, así como registros producidos.



Esto permite generar informes de anomalías y trazabilidad a múltiples niveles (por ejemplo, por aplicación, dispositivo o porción de la red, así como indicadores de salud globales).

- (D<sub>3</sub>) **Inteligencia.** Más allá de meramente detectar eventos anómalos en la red, el sistema debe inferir conocimientos correlacionando los eventos actuales con las etapas pasadas y anticipar los movimientos futuros del atacante. Del mismo modo, debe proporcionar mecanismos para integrar la información procedente de fuentes externas, también denominadas *cyber threat intelligence* [237].
- (D<sub>4</sub>) **Simbiosis.** El sistema debe tener la capacidad de ofrecer su información de detección a otros servicios de la Industria 4.0, mediante interfaces bien definidas. Esto incluye mecanismos de control de acceso (para adaptar las políticas de autorización en función del estado de seguridad de los recursos) o servicios de virtualización (que permitan simular técnicas de respuesta bajo diferentes escenarios sin interferir la configuración real), entre otros.

Por otro lado, también podemos establecer los siguientes requisitos de seguridad con respecto al despliegue de la solución de detección en la red:

- (S<sub>1</sub>) **Recolección de datos distribuida.** Es necesario encontrar mecanismos distribuidos (como agentes P2P) que permitan recoger y analizar la información lo más cerca posible de los dispositivos. El objetivo final es que el sistema de detección sea completamente autónomo y resistente a los ataques dirigidos.
- (S<sub>2</sub>) **Inmutabilidad.** La solución ideada debe evitar las modificaciones de los datos de detección a todos los niveles. Esto incluye preservar la fiabilidad y la veracidad de los datos intercambiados entre los agentes (por ejemplo, mediante niveles de confianza que ponderen la información de seguridad recibida), y el almacenamiento de dichos datos (por ejemplo, mediante medios de almacenamiento inalterables y mecanismos de replicación de datos como bases de datos inmutables o libros de contabilidad distribuidos).
- (S<sub>3</sub>) **Confidencialidad de los datos** Además de la protección contra la modificación de los datos, es obligatorio que el sistema proporcione mecanismos criptográficos y de autorización para controlar el acceso a la información generada por la plataforma de detección, así como a todas las interacciones externas del sistema.
- (S<sub>4</sub>) **Supervivencia.** El sistema debe funcionar correctamente incluso con la presencia de fallos accidentales o deliberados en la infraestructura industrial, y al mismo tiempo no puede ser utilizado como punto de ataque. Para lograrlo, los mecanismos de detección deben desplegarse en una red aislada que sólo pueda recuperar información de la infraestructura industrial.
- (S<sub>5</sub>) **Rendimiento en tiempo real.** El sistema no debe introducir retrasos operativos en la infraestructura industrial, y sus algoritmos no deben imponer una alta complejidad para

garantizar la detección en tiempo real. Para ello, pueden utilizarse procedimientos de segmentación de la red y nodos de computación separados (por ejemplo, usando *fog* o *edge computing*).

Partiendo de estas premisas, pretendemos definir un marco de detección y trazabilidad que facilite el desarrollo de soluciones adecuadas para el contexto de la Industria 4.0, como se resume a continuación.

## A.4 Diseño de un marco de trabajo para la detección y trazabilidad de ataques persistentes avanzados

A partir de los requisitos de seguridad y detección extraídos anteriormente, el Capítulo 4 de la presente tesis doctoral se dedica a definir el marco de trabajo para el desarrollo de soluciones que permitan la correlación distribuida de eventos provocados por una APT y, por tanto, llevar a cabo su trazabilidad. En primer lugar, se introducen algunos conceptos preliminares sobre controlabilidad estructural y teoría de grafos que son necesarios para definir formalmente dicho *framework*. A partir de ellos, se presenta el marco de trazabilidad de APT, especificando su modelo de infraestructura junto con sus entradas y salidas. Posteriormente, con objeto de ilustrar la viabilidad y eficacia de nuestra propuesta, identificamos algunos algoritmos de correlación que satisfacen esa especificación. Por último, realizamos una comparación cualitativa y cuantitativa de estas técnicas para valorar su precisión detectando ataques, antes de aplicarlos experimentalmente en escenarios de la Industria 4.0 en el siguiente capítulo.

### A.4.1 Modelado de redes y ataques de la Industria 4.0 con teoría de grafos

En esta sección, establecemos la base teórica para la representación formal de las infraestructuras de la Industria 4.0, el modelo de atacante de una APT sobre una red definida y las técnicas de detección presentadas en este capítulo.

Comenzando por las infraestructuras de la Industria 4.0, asumiremos que toda red vendrá formalmente definida por un grafo dirigido  $G = (V, E)$ , donde  $V$  es el conjunto de vértices y  $E$  alude al conjunto de aristas o conexiones entre los nodos de la red. A través de este grafo es posible caracterizar las redes de control incluyendo la interconexión de los dispositivos de control con los dispositivos de campo (por ejemplo, sensores o actuadores) para transmitir comandos de control en un sentido y recuperar datos en el contrario.

Para representar el volumen de tráfico experimentado por cada dispositivo nodo utilizamos el concepto de *betweenness centrality* (BC) [240], que da una idea de la conectividad experimentada por cada nodo dentro del grafo. Este indicador adquiere un mayor valor en nodos con un mayor número de caminos más cortos que pasan por ese vértice, de manera que aquellos con más conectividad son precisamente los que participan en un gran número de caminos mínimos. Este



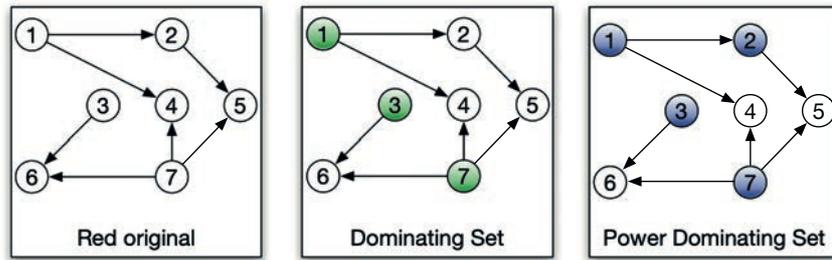


Figura A.4: Elección de nodos dominantes en un grafo

concepto guarda relación con otra noción particularmente útil en nuestra investigación, la cual está asociada con el *conjunto dominante* del grafo (DS, de *Dominating Set*). Este se define como el subconjunto mínimo de nodos  $DS \subseteq V$  que son adyacentes a todo el resto de vértices dentro del grafo. A su vez, el *Power Dominating Set* (PDS) es una extensión de este concepto que se define como el subconjunto mínimo de nodos que son adyacentes a todo el resto de vértices y aristas de  $G(V, E)$ . Ambos conceptos fueron formulados originalmente por Haynes *et al.* en [241], que fueron simplificados por dos reglas fundamentales de observación por Kneis *et al.* en [242]. Un ejemplo de la elección del DS y PDS en un grafo aparece ilustrado en la Figura A.4.

Por otra parte, y con la meta de dotar de realismo a nuestras simulaciones, también caracterizamos la topología del grafo  $G(V, E)$  según las infraestructuras industriales actuales, dividiendo la red en torno a dos secciones: IT y OT, interconectadas ambas secciones por firewalls intermedios. De esta manera, consideramos que la red está compuesta por los subgrafos  $G(V_{IT}, E_{IT})$  y  $G(V_{OT}, E_{OT})$  unidos por un conjunto de cortafuegos ( $V_{FW}$  en adelante), de forma que  $V = V_{IT} \cup V_{OT} \cup V_{FW}$ . Además, cada uno de estos subgrafos es construido aleatoriamente por un generador de topologías con el fin de reproducir las características de una infraestructura real. En concreto,  $G(V_{OT}, E_{OT})$  se construye en base a una distribución power-law del tipo  $y \propto x^{-\alpha}$ , que se utiliza ampliamente para modelar la topología jerárquica de los sistemas de control industriales [243]. Por su parte, la sección IT (dada por  $G(V_{IT}, E_{IT})$ ) se modela según una distribución de red del tipo *small-world*, que representa la topología convencional de las redes TCP/IP [245]. Una vez generados los dos grafos, ambas secciones se unen de manera que todos los nodos de la red IT se conectan a los elementos  $V_{FW}$ , mientras que solo los PDS de la red OT se interconectan con los firewalls. La razón es que mientras que los dispositivos IT poseen amplias capacidades de cómputo y una conectividad más abierta, solo los sistemas SCADA y con mayor jerarquía dentro de la red OT se comunican con la red corporativa. La Figura A.5 muestra un ejemplo sencillo de una red con cinco nodos IT y cinco nodos OT, que se fusionan a través de dos firewalls.

Una vez formalizada la arquitectura de red utilizada en nuestro análisis, también procedemos con la formalización de un modelo de atacante basado en el comportamiento de una APT. Para ello, nos valemos del estudio de amenazas realizado en el Capítulo 2 con respecto a las fases de ataque de una amenaza persistente avanzada. Asumiremos que todos los elementos de la red están

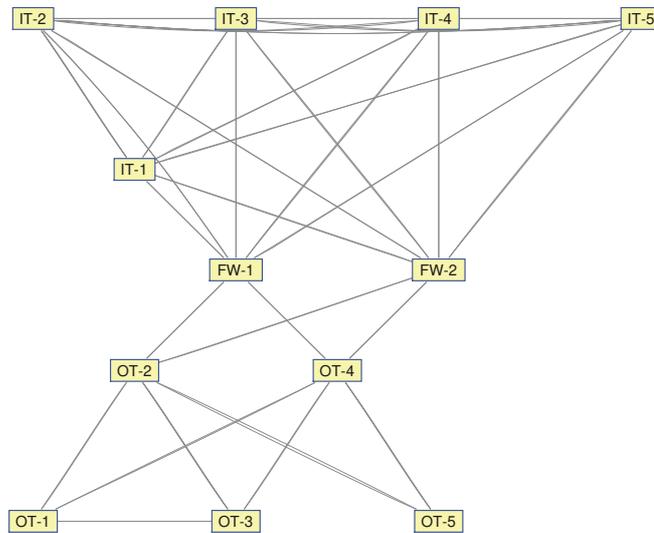


Figura A.5: Ejemplo de red con cinco nodos IT y cinco nodos OT fusionados a través de dos firewalls

cubiertos por mecanismos distribuidos de detección de anomalías, anticipando los principios del marco de trabajo propuesto en este capítulo. En comparación con los mecanismos de detección tradicionales, estos mecanismos se caracterizan por la capacidad de correlacionar anomalías por toda la red y, por tanto, de rastrear la ubicación de los ataques, teniendo en cuenta su gravedad y persistencia. Para este objetivo, recuperarán de forma segura la información proveniente de cualquier mecanismo de detección basado en el host y en la red desplegado en la infraestructura anteriormente definida por el gráfico  $G(V, E)$ . Llegados a este punto asumimos también que, como resultado de la correlación de estos mecanismos de detección que monitorizan el comportamiento de cada nodo y sus vecinos, a cada dispositivo se le asignará matemáticamente una determinada probabilidad de detección (es decir, probabilidad de que se produzca un ataque) en un intervalo de tiempo determinado.

De esta manera, el modelo de atacante vendrá representado por una cadena definida de fases que la APT puede perpetrar contra la red  $G(V, E)$ . Formalmente hablando, estas fases podrán ser del siguiente tipo en nuestro análisis y simulaciones, en alusión a las etapas extraídas del análisis de APT reales efectuado en el Capítulo 2:

- ***intrusionInicial*** $_{(IT,OT,FW)}$ . El acceso inicial infecta un nodo  $n_0$  (conocido como "paciente cero") de la red IT, la red OT o el firewall, respectivamente.
- ***compromiso***. El adversario toma el control de un determinado nodo  $n_i$ , obteniendo mayores privilegios, manteniendo la persistencia y ejecutando técnicas evasivas para eludir la defensa de la red. Esta etapa también incluye el reconocimiento interno de la vecindad directa de  $n_i$ , denotada como  $neighbours(n_i)$ .

- **movimientoLateralDirigido<sub>(IT,OT,FW)</sub>**. Desde un determinado nodo  $n_i$ , el atacante elige un nodo  $n_j$  de la red IT, OT o el firewall del conjunto  $neighbours(n_i)$ , y ejecuta un movimiento lateral hacia ese nodo. Nótese que, en este modelo, el concepto de movimiento lateral sólo abarca el envío de malware hacia el nodo objetivo.
- **movimientoLateralControl**. A partir de un determinado nodo  $n_i$ , el adversario elige el nodo  $n_j$  del conjunto  $neighbours(n_i)$  con mayor betweenness centrality (es decir, el nodo con más conectividad), y ejecuta un movimiento lateral hacia ese nodo.
- **movimientoLateralAleatorio**. Desde un determinado nodo  $n_i$ , el adversario elige un nodo aleatorio  $n_j$  del conjunto  $neighbours(n_i)$ , y ejecuta un movimiento lateral hacia ese nodo.
- **exfiltración**. Desde un determinado nodo  $n_i$ , el adversario establece una conexión con un command&control remoto y extrae información sensible.
- **destrucción**. El adversario inutiliza o destruye el equipamiento físico controlado por el nodo  $n_i$ .
- **idle**. En esta fase no se realiza ninguna operación. Incluimos esta etapa para representar el transcurso de tiempo sin que el atacante efectúe ninguna acción.

Una vez formalizadas las fases, es posible representar una APT en forma de un conjunto ordenado de estas fases,  $attackSet_{APT}$ . Por ejemplo, el conjunto de ataque de Stuxnet [123] puede representarse de la siguiente manera:

$$attackSet_{Stuxnet} = \{intrusionInicialIT, compromiso, exfiltración, movimientoLateralDirigido_{FW}, compromiso, movimientoLateralDirigido_{OT}, \dots, movimientoLateralDirigido_{OT}, idle, \dots, destrucción\}$$

En cuanto a cómo influyen las diferentes etapas de ataque en el cálculo de las probabilidades de detección, hay que tener en cuenta que ciertas etapas de ataque generarán más alertas de seguridad. Esto, a su vez, aumentará la probabilidad de detectar esa etapa de ataque en particular. Por consiguiente, tenemos que considerar la existencia de diferentes clases de probabilidades de detección. Para ello definimos  $\Theta$  como un *conjunto ordenado de probabilidades de detección* de tamaño  $d$ , donde  $\Theta = \{\theta_1, \dots, \theta_d\}$  and  $\theta_i = [0, 1]$ , tal que  $\forall \theta_i, \theta_i > \theta_{i+1}$ .

En base a  $\Theta$  podemos crear un modelo que asigne cada fase del conjunto  $attackStages$  a los elementos de  $\Theta$ . Dicho modelo, en el que  $d = 5$  y  $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5\}$ , se describe en la Tabla A.7. El razonamiento tras esta asignación es la siguiente:

- Asignamos  $\theta_1$  sólo a la fase de *destrucción*, ya que cualquier interrupción en la funcionalidad de un dispositivo probablemente active múltiples alertas de alta prioridad.

$initialIntrusion(n_0)$	$\theta_3$
$compromise(n_i \rightarrow neighbours(n_i))$	$\theta_2 \rightarrow \theta_5$
$*LateralMovement_{IT,FW}(n_i \rightarrow n_j)$	$\theta_5 \rightarrow \theta_4$
$*LateralMovement_{OT}(n_i \rightarrow n_j)$	$\theta_5 \rightarrow \theta_3$
$spreadLateralMovement(n_i \rightarrow neighbours(n_i))$	$\theta_5 \rightarrow \theta_4$
$exfiltration(n_i)$	$\theta_4$
$destruction(n_i)$	$\theta_1$

Tabla A.7: Asignación del conjunto  $attackStages$  a  $\Theta$

- $\theta_2$  sólo se asigna al nodo protagonista de la fase de *compromiso* ( $n_i \rightarrow neighbours(n_i)$ ). La razón de esto es simple. El acto de comprometer y tomar el control de  $n_i$  no sólo desencadenará varias alertas de host, sino también múltiples alertas de red debido a las diversas consultas de descubrimiento dirigidas a todos los nodos vecinos en  $neighbours(n_i)$ . La correlación de todos estos eventos llamará la atención sobre el estado de  $n_i$ .
- Para  $\theta_4$ , consideramos las alertas de seguridad causadas por la combinación de una única conexión anómala a un nodo más la entrega de malware a ese nodo. Como tal, esta  $\theta$  cubre todos los elementos del lado derecho de las fases del tipo *movimientoLateral*. No obstante, en algunos casos particulares (como la etapa *intrusionInicial* y la fase del tipo *movimientoLateral<sub>OT</sub>*), se detectarán anomalías adicionales. Por un lado, una conexión externa potencialmente anómala. Por otro, una cierta inestabilidad en el entorno OT (normalmente estable). En esas fases les asignaremos  $\theta_3$ .
- Por último,  $\theta_5$  se asigna a aquellas etapas en las que los nodos producen o reciben tráfico anómalo (por ejemplo, una conexión que se desvía de lo que se considera tráfico normal). Nuevamente, en las situaciones en las que se produce una conexión con el exterior (por ejemplo, la etapa de *exfiltración*), la posibilidad de tráfico anómalo y, por tanto, el valor de  $\theta$  aumentará.

En nuestras simulaciones, estos valores de  $\Theta$  son asignados considerando un escenario de ataque realista, y añadiendo cierto nivel de aleatoriedad para representar potenciales desviaciones en la detección de anomalías. Con esto, hemos introducido formalmente los elementos clave que intervienen en el modelo de atacante y la detección de anomalías distribuida, necesarios para entender el marco de trazabilidad de APT ideado a continuación.

#### A.4.2 Especificación del marco de trabajo para la trazabilidad de APT

Tras estudiar los sistemas más representativos para la detección de intrusiones en entornos de la Industria 4.0 y formalizar la infraestructura de estos entornos junto con el comportamiento de una APT, presentamos en esta sección el núcleo de nuestro trabajo. El marco de trazabilidad nace para agregar la cobertura de múltiples sistemas de detección que son desplegados de manera distribuida por la red, bajo una especificación común que correlaciona anomalías y aprende

permanentemente de todos los patrones de malware detectados, adaptándose flexiblemente a las tecnologías integradas y monitorizando el ciclo de vida de una APT.

En primer lugar, presentamos el modelo de adquisición de información proveniente de la infraestructura, para la detección de anomalías. Suponemos que la red viene representada por un grafo  $G(V, E)$  donde  $V = V_{IT} \cup V_{OT} \cup V_{FW}$ , tal como se dispuso anteriormente. Con objeto de cumplir con los requisitos de cobertura (D1, c.f. Sección 3.5) y recolección de datos distribuida (S1), asumiremos que para cada uno de los dispositivos de la red existe un agente de detección asociado, encargado de monitorizar en todo momento el estado del mismo (incluyendo información del host, uso de recursos computacionales, parámetros de red o de las comunicaciones, así como valores medidos o comandos ejecutados). Estos agentes pueden ser virtuales o físicos, según las restricciones del despliegue por la red y la capacidad para integrar dispositivos de medición de tráfico por la infraestructura, tal como se ilustra en la Figura A.6. Según se aprecia, podemos considerar hasta cuatro modelos de despliegue distintos:

- a) **Implementación centralizada:** suponemos que todo el tráfico de los dispositivos es procesado por una entidad centralizada encargada de ejecutar un proceso por cada agente virtual asociado a esos nodos.
- b) **Implementación distribuida:** en este caso, los agentes son dispositivos físicos asociados al equipamiento industrial o bien ejecutados sobre el propio hardware de control, siendo la comunicación plenamente distribuida entre ellos.
- c) **Implementación centralizada con brokers de datos:** se trata de una solución intermedia donde estos dispositivos son colocados en puntos estratégicos de la red para captar el tráfico y asociarlo a agentes de detección virtuales. Estos brokers son, además, desplegados en una red aislada, para cumplir con los requisitos de inmutabilidad (S2), confidencialidad (S3) y supervivencia (S4).
- d) **Implementación mixta:** se trata de una solución híbrida, donde parte de los agentes son dispositivos físicos que se comunican con otros implementados virtualmente en nodos con una mayor capacidad de cómputo, como los ya citados brokers.

Independientemente del modelo de adquisición y despliegue, cada agente de detección es capaz de derivar un valor de anomalía (un número real en el intervalo  $[0,1]$  que puede expresarse porcentualmente) del dispositivo monitorizado en función de la información recabada tras aprender su comportamiento (haciendo uso de técnicas de *machine learning*) o integrando técnicas de IDS externas, cumpliendo así con el requisito D3 (inteligencia). Este valor de anomalía será correlacionado con la de los agentes vecinos, siguiendo la topología de red descrita por la infraestructura industrial. Por este motivo y dado que podemos encontrarnos con varios modelos de despliegue de agentes, esa técnica o algoritmo de correlación podrá adoptar dos modelos de datos distintos: (i) ya sea un modelo global donde consideramos que todos los agentes disponen de

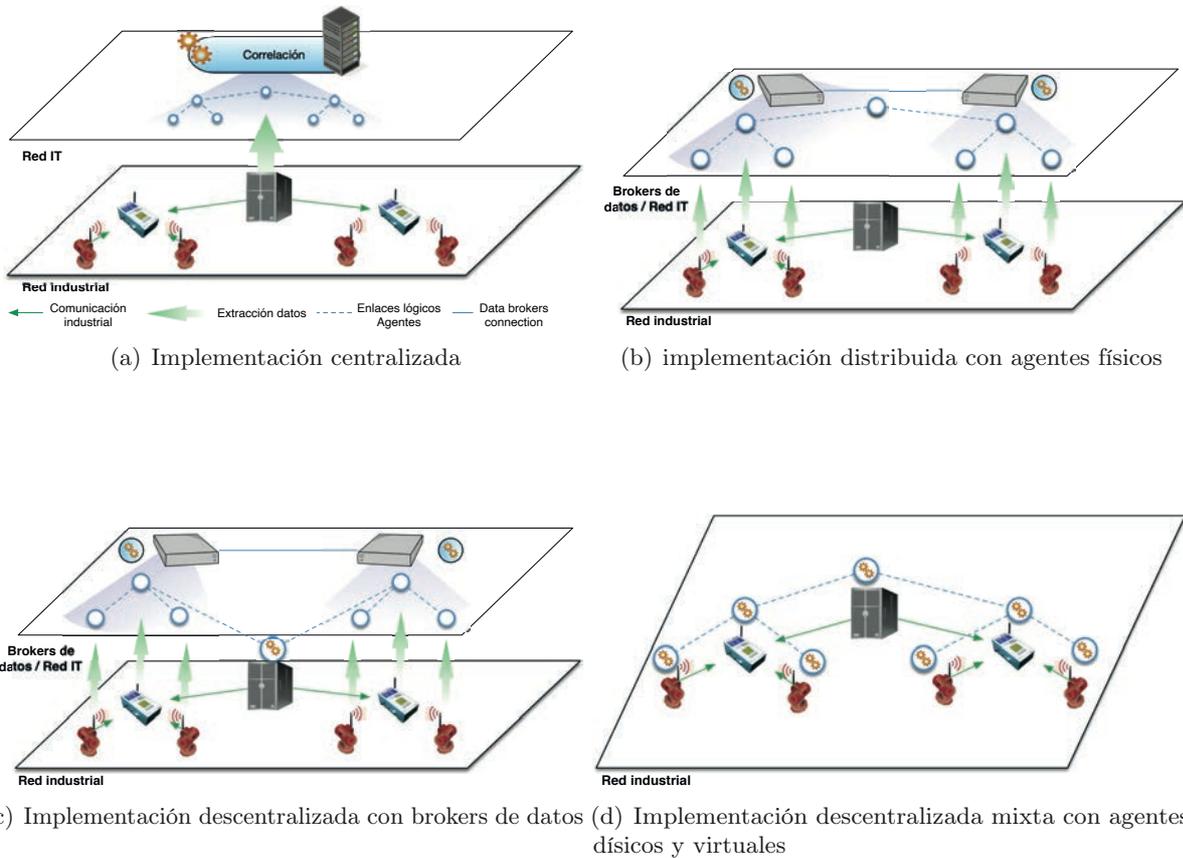


Figura A.6: Implementación de los agentes de detección para la adquisición de información y correlación de anomalías

información completa acerca de la red, o bien (ii) asumiendo que solo disponen de información de su contexto local. Esto afecta a cómo los agentes han de comunicarse para propagar la información a lo largo y ancho de la red (en especial si disponemos de un modelo de datos global) o a la sincronización de datos entre nodos distintos.

Al fin y al cabo, la elección final del algoritmo de correlación, el modelo de datos y el diseño arquitectónico de los agentes responde a restricciones de rendimiento y de despliegue de la infraestructura. En cualquier caso y para dar paso a la especificación del propio marco de trabajo, ese algoritmo de correlación debe satisfacer una interfaz específica de inputs o entradas de datos:

- (I<sub>1</sub>) **Input cuantitativo:** lo expresaremos con el vector  $x$  para asignar a cada activo en la red un valor de anomalía antes de realizar la correlación. Como se ha mencionado anteriormente, puede ser calculado por cada agente asociado o utilizando mecanismos de detección externos, tomando un amplio conjunto de entradas de datos para cumplir con los requisitos D1 (cobertura) y D2 (holismo). En nuestras simulaciones, este valor viene dado por las fases

de ataque ejecutadas en la red de forma probabilística (es decir, el valor de  $\Theta$ ), sin que el mecanismo de detección del agente tenga conocimiento de las fases reales.

- ( $I_2$ ) **Input cualitativo:** los valores anteriores necesitan ser enriquecidos con información del contexto que permita correlacionar eventos en dispositivos cercanos e inferir la ejecución de etapas de ataque específicas que reportan un mayor valor de anomalía. Podemos suponer que este conocimiento puede reflejarse en forma de un peso  $w_{ij}$ , que es asignado por cada agente  $i$  a cada uno de sus vecinos para representar el nivel de confianza adjudicado a sus indicaciones de anomalía al realizar la correlación.

En cuanto a los resultados de las soluciones de trazabilidad que se ajusten a este marco, deben incluir, entre otros, los siguientes elementos u *outputs*:

- ( $O_1$ ) **Información local** para determinar si el agente ha encontrado una anomalía provocada por la infección real del nodo asociado, en base a un nodo vecino o si es debida a un falso positivo.
- ( $O_2$ ) **Información a nivel global**, para determinar el grado de afección de la red y los nodos que han sido comprometidos, filtrados por zonas. Esto permite distinguir qué dispositivos están experimentando el mismo grado de anomalía producido por una misma fase ataque, lo cual es esencial para aplicar técnicas de respuesta efectivas y aislar el ataque mientras el resto de las zonas pueden seguir funcionando como en condiciones normales.
- ( $O_3$ ) **Información contextual** que permite correlacionar eventos pasados y visualizar la evolución de la amenaza, además de anticipar los recursos que pueden ser objetivo de la APT, de acuerdo a los requisitos de inteligencia (D3) y simbiosis (D4). Esto incluye el registro completo de eventos ocurridos en la red desde el momento en que la intrusión irrumpió en ella. Para ello, hay que tener en cuenta la persistencia de los ataques en todo momento, ya que una amenaza avanzada puede pasar desapercibida durante meses antes de ejecutar una nueva acción. En lo que respecta al algoritmo de correlación, esto implica que también es necesario hacer un seguimiento de las antiguas y sutiles anomalías detectadas en la red para evaluar su relevancia con respecto a las anomalías actuales.

En conjunto, esta especificación define un marco para el desarrollo de soluciones de detección distribuida de APT en escenarios industriales, tal y como se representa en la Figura A.7. Este diagrama ilustra el flujo de datos desde su adquisición desde los dispositivos finales hasta la correlación de anomalías, utilizando los brokers de datos introducidos con antelación. Tras esto, presentamos ahora las soluciones candidatas que implementan este *framework* para lograr los objetivos de trazabilidad de APT propuestos hasta ahora.

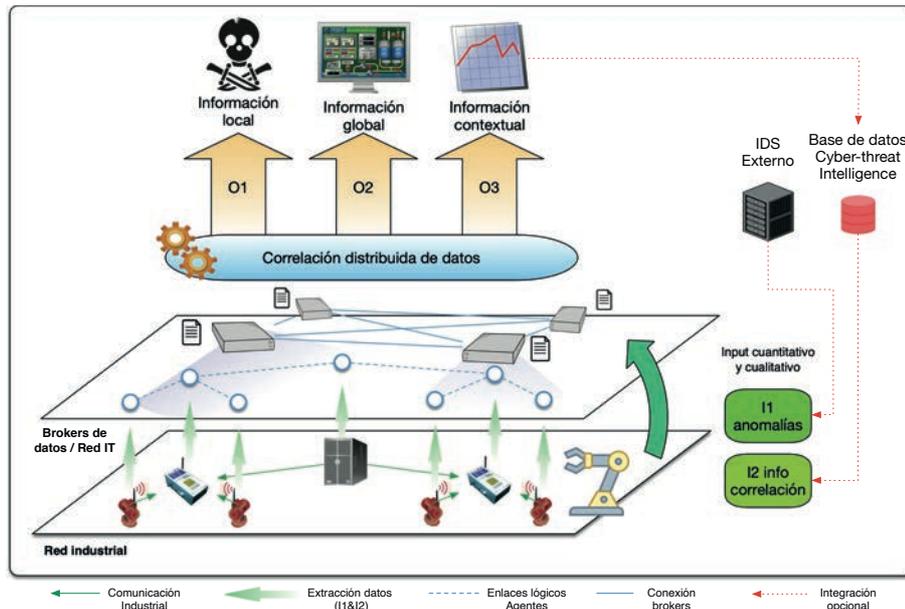


Figura A.7: Especificación de entradas y salidas del marco de detección y trazabilidad de APT

### A.4.3 Técnicas de correlación distribuida

El marco propuesto anteriormente define claramente un modelo de adquisición de información y una interfaz de datos de entrada y salida que debe ser implementada por un algoritmo de correlación. En esta sección, presentamos las soluciones candidatas para la trazabilidad de las APT, antes de comparar su precisión en la siguiente sección.

Para la elección de estos algoritmos candidatos, procedimos primero a realizar una primera exploración de algoritmos de consenso. Se trata de un área de especial interés en computación distribuida y sistemas multiagente, que estudia cómo un conjunto de agentes son capaces de colaborar y obtener la misma información para alcanzar un objetivo común [248]. Además, se ha aplicado ampliamente en aplicaciones del mundo real como la sincronización de relojes, la agregación de datos entre nodos de una blockchain o la coordinación de robots autónomos, entre otras.

Aunque el despliegue de estos sistemas ofrece una mayor eficiencia y capacidad operativa que sistemas autónomos y centralizados, el problema reside en que su principal función es precisamente la consecución de una solución común de manera colaborativa. Traducido al contexto de detección de APT, estos nos proporcionarían una herramienta para evaluar la salud de la red en su conjunto, pero no nos arrojaría información útil a nivel local, con objeto de cumplir con el output  $O_2$  deseable para la solución de trazabilidad.

Ante esta problemática, aparece una alternativa basada en la presencia de más de un consenso distribuido por la red, adaptándose, por tanto, a nuestra especificación. Se trata del algoritmo de Opinion Dynamics [252], que permite modelar la influencia entre individuos de un grupo o

sociedad, donde existe un amplio espectro de opiniones. Primero, cada agente elabora su propia opinión teniendo en cuenta las del resto de agentes con cierto grado de influencia. Este proceso continúa hasta alcanzar un estado estacionario en el que los agentes ya no cambian su opinión. En ese momento, las opiniones se distribuyen en varios espectros, y es posible estudiar su propagación. Para nuestros objetivos, esto significa poder fragmentar la red en función de las múltiples amenazas que puedan tener lugar en zonas independientes, cuyo valor de consenso individual represente el grado de severidad de los ataques sobre esa región concreta de la infraestructura.

A continuación, presentamos de manera resumida el algoritmo en cuestión, que constituye una ligera modificación del enfoque propuesto en [252]. Para empezar, suponemos la presencia de agentes desplegados por una red  $G(V, E)$ , de modo que cada nodo  $v$  tiene un agente asociado. En este contexto,  $x_i(t)$  representa la opinión (que va de cero a uno) de un agente  $i$  en el intervalo tiempo  $t$ , donde  $t$  se refiere a la iteración del algoritmo. Así, el vector  $x(t) = (x_1(t), \dots, x_n(t))$  representa las opiniones en el momento  $t$  para todos los agentes de la red, cumpliendo con la especificación del input cuantitativo  $I_1$  de nuestro marco de trazabilidad. Por otra parte, dado un agente  $i$ , el peso dado a la opinión de cualquier otro agente  $j$  se denota por  $w_{ij}$ , donde  $\sum_{k=1}^n w_{ik} = 1$  (por consiguiente, el agente  $i$  también tiene en cuenta su propia opinión). Estas ponderaciones pueden cambiar con el tiempo o por opinión, de modo que un agente  $i$  ajusta su opinión en el periodo  $t + 1$  teniendo en cuenta la opinión de cada agente  $j$  en el momento  $t$ . A su vez esto representa el conocimiento para discernir qué anomalías están relacionadas, de acuerdo al segundo input cualitativo ( $I_2$ ) del marco de trabajo definido.

Finalmente, la opinión para el agente  $i$  en la siguiente iteración  $t + 1$  se calcula así:

$$\sum_{j=1}^n x_j(t) w_{ij} = x_i(t+1)$$

En una notación matricial, esta expresión se puede escribir como:

$$x(t+1) = W(t, x(t))x(t)$$

donde la matriz  $W(t) = [w_{ij}]$  es la matriz cuadrada que recoge los pesos entre agentes. Según el algoritmo original y por simplicidad, asumimos que, para un agente dado, el valor del peso asignado a sus vecinos se divide uniformemente en aquellos agentes cuya opinión está muy cerca de su propio valor, estableciendo un umbral  $\varepsilon$  entre ambas opiniones. Esto representa el hecho de que los agentes cercanos con el mismo grado de anomalía probablemente acaben detectando la misma amenaza en su entorno.

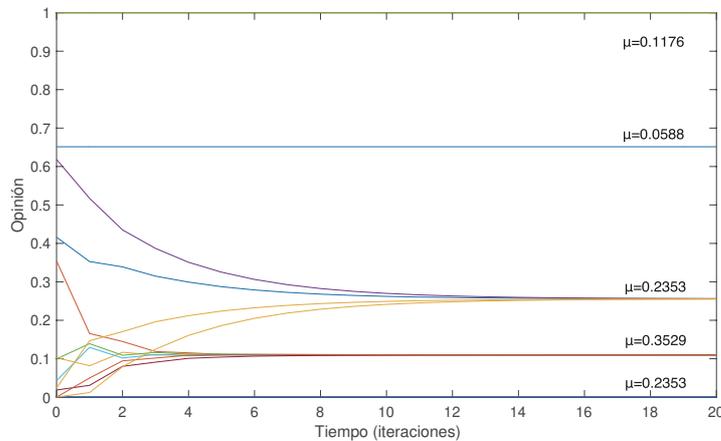


Figura A.8: Cálculo de las opiniones para un conjunto de 30 nodos sometidos a 10 fases de ataque

Como conclusión, cada agente ajusta su opinión en el periodo  $t + 1$  tomando una media ponderada de las opiniones del resto de agentes, adaptándose así a una arquitectura distribuida. Por un lado, esta opinión resultante arroja información acerca de si el dispositivo en cuestión ha sido comprometido, en alusión al primer output del marco de trazabilidad ( $O_1$ ), relativo al contexto local del agente. Por otra parte, cuando  $t$  tiende a infinito y la red entra en equilibrio, se forman consensos de opiniones compartidas por grupos disjuntos de agentes que pueden representarse visualmente. Véase a modo de ejemplo la Figura A.8, donde se ilustra la ejecución de este algoritmo para una red de 30 nodos y 17 agentes tras sufrir una APT compuesto por 10 ataques. Las líneas representan la evolución en las opiniones para cada agente, teniendo finalmente múltiples consensos entre ellos. Al gráfico se ha añadido además un valor  $\mu$ , que contiene la proporción de agentes que entran finalmente en consenso. De igual manera, esta información se corresponde con el segundo output de nuestro *framework*, relacionado con la salud global de la red en un momento determinado.

Con respecto al tercer output del marco de detección ( $O_3$ ), es preciso estudiar cómo este algoritmo de correlación puede proporcionar información útil sobre la evolución de la amenaza. Esta funcionalidad gira en torno a la fluctuación de los valores de  $O_1$  y  $O_2$  a lo largo del tiempo. Y es que si llevamos un registro de las opiniones generadas por los agentes a lo largo del tiempo, es posible identificar y representar visualmente la secuencia de fases de ataque efectuadas por una APT y extraer indicadores de salud globales para la red, tal como se muestra en la Figura A.9. En ella, se ilustra la evolución de las opiniones resultantes tras ejecutar el algoritmo de Opinion Dynamics después de cada una de las fases de la APT Stuxnet que, como se describió anteriormente, comienza comprometiendo recursos de la red IT para luego propagarse a la red OT y destruir un PLC (Programmable Logic Controller).

En nuestras simulaciones, cabe señalar que el valor de anomalía medido por cada agente (y que corresponde con su opinión en  $t = 0$ , es decir, antes de ejecutar el algoritmo para correlacionarla

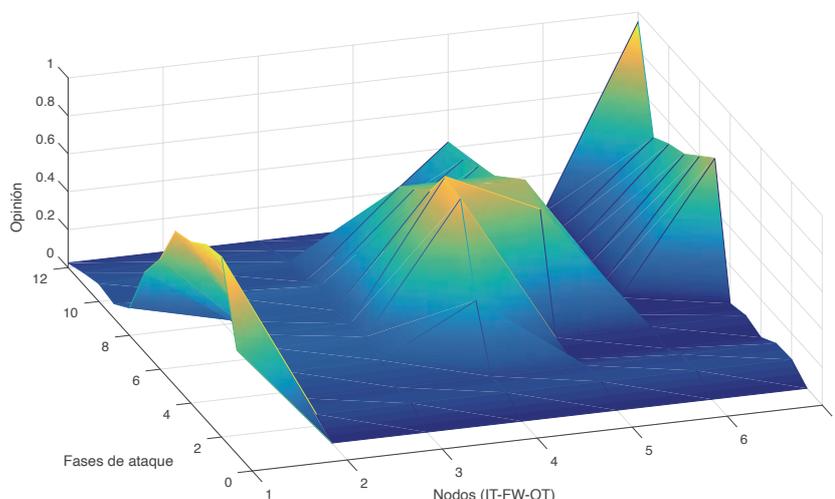


Figura A.9: Evolución de las opiniones a lo largo de las fases de ataque de Stuxnet

con la anomalía de los vecinos) es calculado según el valor de  $\Theta$  previamente introducido, que es más alto cuanto mayor es la gravedad de la fase APT en cuestión (o mejor dicho, su probabilidad de detección). De igual modo, para simular el paso del tiempo a lo largo de las distintas fases de ataque, esas anomalías se irán devaluando según un indicador de atenuación para reducir la influencia de ataques antiguos a la hora de computar las nuevas opiniones de los agentes. Tal como se describe en la Sección 4.6 de esta tesis doctoral, este decremento en el valor de las anomalías (el valor del input  $I_1$ ) dependerá de la severidad del ataque y de la criticidad del dispositivo afectado: cuanto más devastadora sea la alerta generada (durante la fase de detección), más tardará en desaparecer su efecto.

Gracias al valor del output  $O_3$  del marco de trabajo es posible trazar los movimientos del atacante y visualizar los nodos afectados en cada una de las fases de la APT, llevando a cabo la ejecución sucesiva del algoritmo de correlación a partir de la información de detección que extraen los agentes. Esto se puede apreciar en la Figura A.10, donde varias fases de la misma APT explicada anteriormente son estudiadas de manera independiente monitorizando las opiniones de todos los nodos en una red.

En esta sección presentamos, además, otra solución alternativa al algoritmo de Opinion Dynamics, utilizando *clustering* (o de agrupamiento). Estas técnicas se han utilizado tradicionalmente como método no supervisado para el análisis de datos, en el que un conjunto de datos se agrupa según algún criterio de similitud. En nuestro caso, disponemos de dispositivos que se ven afectados por ataques relacionados y que, por tanto, generan anomalías similares, pudiendo agruparlos en base a este criterio. Al fin y al cabo, el algoritmo de Opinion Dynamics simplemente divide una red en subgrupos de dispositivos que presentan una anomalía similar, y relaciona las zonas que

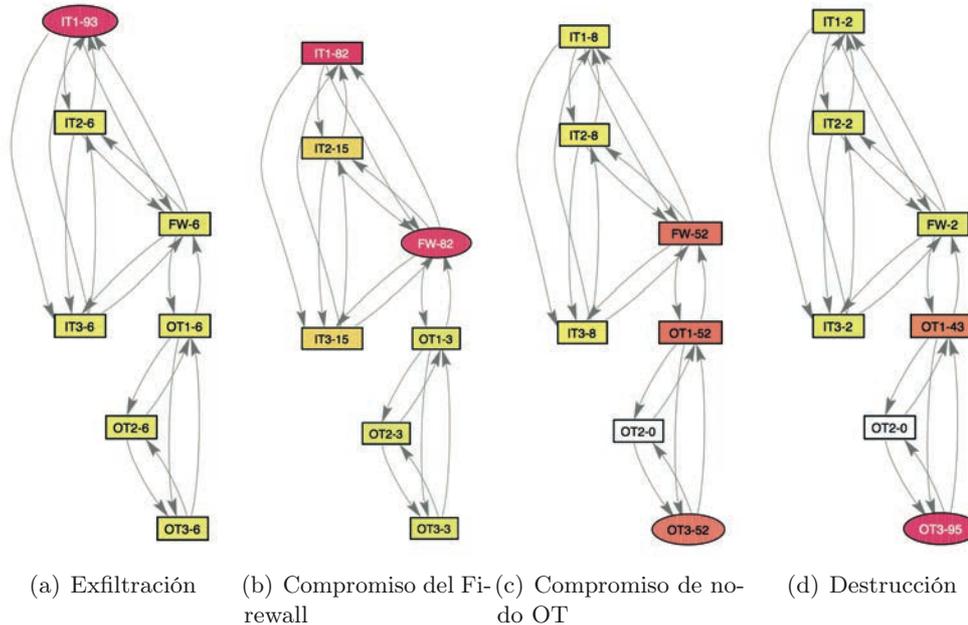


Figura A.10: Ejecución de Opinion Dynamics tras varias fases de la APT Stuxnet sobre una red sencilla

pueden haber experimentado el mismo ataque. Este razonamiento también puede satisfacerse con mecanismos de *clustering*, cumpliendo el marco de trazabilidad establecido.

En más detalle, son los enfoques de *clustering* basados en centroides los que se adecuan a esta especificación. Métodos clásicos como K-means dividen un conjunto de datos seleccionando inicialmente  $k$  centroides de cluster y asignando cada elemento a su centroide más cercano. Los centroides se actualizan repetidamente hasta que el algoritmo converge en una solución estable. En el escenario de la trazabilidad de APT, son las anomalías detectadas por los agentes las que desempeñan el papel de las instancias de datos que deben agruparse en conjuntos disjuntos (es decir, el input cuantitativo  $I_1$ ). Por otra parte, la relación entre anomalías es la que determina cuáles de ellas se agrupan en un mismo clúster, que puede determinarse en base a múltiples criterios, representando el input cualitativo ( $I_2$ ) de la solución a desarrollar. Mientras que en el algoritmo de Opinion Dynamics esto quedaba representado por el peso  $w_{ij}$  asignado entre agentes, en el enfoque basado en *clustering* se puede modelar en forma de dimensiones adicionales de las instancias de datos (el valor de las anomalías, en un principio unidimensionales).

No obstante, el problema principal al que se enfrenta este enfoque se puede resumir en estos dos puntos:

- **La elección del valor de  $k$ .** Es un inconveniente clásico del algoritmo K-means, y es que el número de conjuntos en que dividir las anomalías ha de ser especificado desde el principio y no suele ser conocido de antemano. Algunas publicaciones sugieren determinar el valor

más adecuado usando métodos estadísticos [269] o visualmente [270]. También es habitual estudiar los resultados de un conjunto de valores en lugar de un único  $k$ , para elegir el más óptimo en base a distintos indicadores.

- **La representación de la topología de red en el algoritmo de correlación.** Al aplicar K-means, suponemos que el conjunto de datos está formado por un conjunto de puntos multidimensionales. En nuestro caso, tenemos un vector unidimensional de anomalías en el rango  $[0,1]$ . Sin embargo, la agrupación de estos valores está sujeta a la topología de red, dado que pretendemos correlacionar anomalías similares entre agentes cuyos dispositivos monitorizados están efectivamente conectados, tal como realiza implícitamente el algoritmo de Opinion Dynamics. Por tanto, es necesario proporcionar este conocimiento al algoritmo y reflejar estas condiciones del entorno como entradas (al margen de  $I_1$  e  $I_2$ ) a la correlación. En este sentido, algunas propuestas sugieren un clustering con K-means sujeto a restricciones [272], o esquemas específicos para dividir un grafo en clusters utilizando *spanning trees* o componentes fuertemente conectados [273].

En el contexto de esta tesis doctoral, hemos optado por probar varias de las soluciones propuestas. Entre ellas, barajamos un *clustering basado en la localización* de los nodos como dimensión adicional que, sin embargo, seguía arrastrando el problema de la selección del valor de  $k$ . Finalmente, nos quedamos con una técnica propia a la que hemos apodado como *accumulative anomaly clustering* o *clustering acumulativo*. Este algoritmo comienza seleccionando el nodo más afectado dentro de la red y, posteriormente, aplica su influencia a los nodos circundantes. Esto se representa añadiendo un valor entero a las anomalías de dichos agentes (inicialmente de 0 a 1), que es proporcional a la anomalía del nodo influyente. Esta influencia se ejerce siempre que la diferencia entre ambas anomalías (es decir, la del nodo influyente y la del influido) no supere un umbral definido  $\varepsilon$ , similar al enfoque de Opinion Dynamics para cumplir con  $I_2$ . A continuación, el algoritmo continúa seleccionando el siguiente en la lista de nodos ordenados inversamente por el valor de la anomalía, hasta que todos los nodos hayan sido influenciados o hayan influido en otros. En ese momento,  $k$  es asignado automáticamente con el número de nodos influyentes en la red, y podemos ejecutar K-means con las instancias de datos modificadas. Los valores resultantes de cada agente corresponden a la parte decimal de su centroide asociado, siendo comparable a las opiniones del enfoque de Opinion Dynamics.

La idea tras esta técnica (que puede enriquecerse para incluir factores adicionales a  $I_2$ ) asume que los ataques sucesivos generan un valor de anomalía similar en los agentes más cercanos, como postula el algoritmo Opinion Dynamics. Al mismo tiempo, aborda la cuestión de la selección de  $k$  y la inclusión de información topológica a la clusterización, tal como hemos explicado anteriormente.

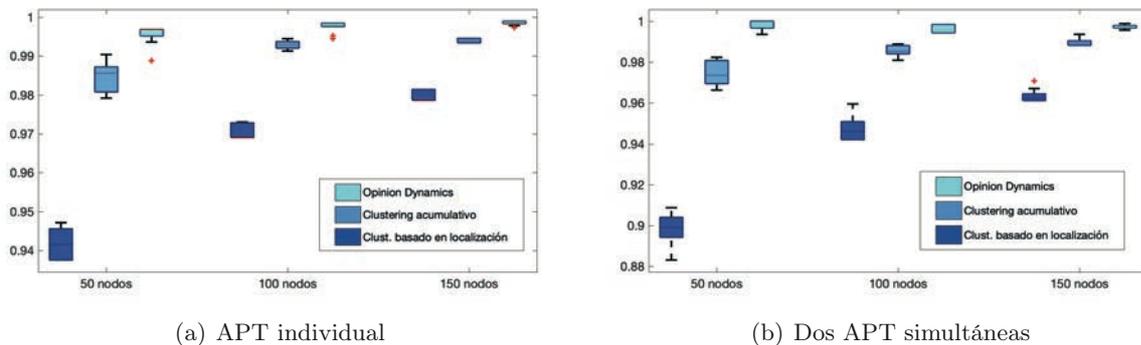


Figura A.11: Promedio de la pureza para las técnicas de correlación

#### A.4.4 Comparación de soluciones

Después de presentar las dos soluciones candidatas a cumplir con el marco de trabajo presentado en el núcleo tesis (Opinion Dynamics y la técnica de clustering acumulativo), procedemos a poner a prueba ambas técnicas para comparar su precisión ante diversas APT modeladas teóricamente, siguiendo la formalización presentada en la Sección A.4.1.

Con objeto de evaluar sus capacidades de detección y trazabilidad, ejecutamos diez APT generadas aleatoriamente como conjuntos de fases de ataque perpetrados contra una red representada en forma de grafo, con distintas instancias de 50, 100 y 150 nodos generadas aleatoriamente. Las anomalías producidas por estos ataques (calculadas de manera probabilística tal como ya se ha explicado) son monitorizadas constantemente por ambas técnicas de correlación. Tras cada fase individual, se evalúa la precisión con la que ambas soluciones identifican los nodos que están siendo atacados, haciendo uso de un indicador de "pureza", que es un criterio de evaluación bastante extendido para medir la calidad de los algoritmos de *clustering* [274].

La Figura A.11 representa el promedio de pureza en esas simulaciones. Concretamente, se ha estudiado su valor para la técnica de Opinion Dynamics y de *clustering* acumulativo, incluyendo además la técnica de *clustering* basado en la localización mencionada anteriormente. Adicionalmente, se ha llevado a cabo las mismas simulaciones pero incluyendo dos APT ejecutadas simultáneamente sobre la red. En todos los casos, tal como podemos visualizar, el algoritmo de Opinion Dynamics arroja mejores resultados, con una precisión ligeramente superior a la reportada por la técnica de *accumulative anomaly clustering*. Esta superioridad también queda en evidencia con el estudio de otros indicadores de precisión, como el *Rand Index* [275].

Como resultado de estas pruebas, podemos concluir que pese a que ambas técnicas se adoptan fielmente a la especificación de nuestro *framework*, es el algoritmo de Opinion Dynamics el que muestra una mayor precisión en el seguimiento de ataques complejos. Como consecuencia, será esta técnica la que utilizaremos en el siguiente capítulo para probar la efectividad de nuestro marco de trabajo en distintos escenarios de seguridad de la Industria 4.0.

## A.5 Casos de uso para la protección de la Industria 4.0

El marco de trazabilidad de APT tiene en cuenta diversas arquitecturas de red, tipos de ataque y modelos de adquisición de datos, para posteriormente definir las entradas y salidas que deben incluir las soluciones para cumplir con los requisitos de detección y seguridad. Esto sienta las bases para el desarrollo y la comparación de nuevas soluciones en este contexto. Como medio para validar este marco, hemos definido dos mecanismos de detección basados en *clustering* y Opinion Dynamics. Según los experimentos teóricos descritos anteriormente, este último presenta una mayor precisión para el seguimiento de amenazas.

En el Capítulo 5 ponemos en práctica este mecanismos de detección para profundizar en su aplicación práctica en varios escenarios de la Industria 4.0. El objetivo es evaluar su efectividad para la puesta en marcha de técnicas de respuesta que disminuyan el impacto de las amenazas persistentes avanzadas. Un resumen de cada uno de estos análisis es ofrecido a continuación.

### A.5.1 Protocolo de encaminamiento de mensajes seguro

Como medio para probar la utilidad del *framework* de trazabilidad de APT (por medio del algoritmo de Opinion Dynamics), en primer lugar exploramos la implementación de protocolos de encaminamiento seguros. El objetivo que nos proponemos es el de aprovechar la información acerca del estado de seguridad de la red (y, más concretamente, los outputs  $O_1$  y  $O_2$ ) para garantizar la continuidad de la infraestructura en presencia de ataques.

Esta funcionalidad es diseñada e implementada a través de dos técnicas de respuesta con objetivos distintos. La primera de ellas asume la presencia de un atacante que toma el control de algunos nodos de la red, con la capacidad de interceptar el tráfico (poniendo en jaque la confidencialidad de la información) o directamente denegar el servicio en algunos enlaces de comunicación. Ante este modelo de atacante, proponemos el despliegue de una arquitectura de red redundante, que permita el envío efectivo de mensajes entre cualquier nodo emisor y destinatario. Para ello seguimos tres estrategias (STG, del inglés *strategy*):

- **STG1:** añadir ejes redundantes a todos los nodos de la red.
- **STG2:** añadir ejes solo a aquellos nodos con mayor conectividad en la topología (los del conjunto dominante o DS).
- **STG3:** añadir ejes redundantes a solo a los nodos que no son parte del DS.

Además de ello, para evitar la interceptación de mensajes en la red, implementamos un protocolo de compartición de secretos. Esto permite dividir el mensaje original en  $n$  trozos de manera que el atacante ha de reunir un número mínimo  $1 \leq k \leq n$  para poder acceder a la información (y al mismo tiempo evitamos que el receptor legítimo obtenga el mensaje aun cuando algunas de esas partes se pierden debido a un ataque de denegación de servicio). Este valor será

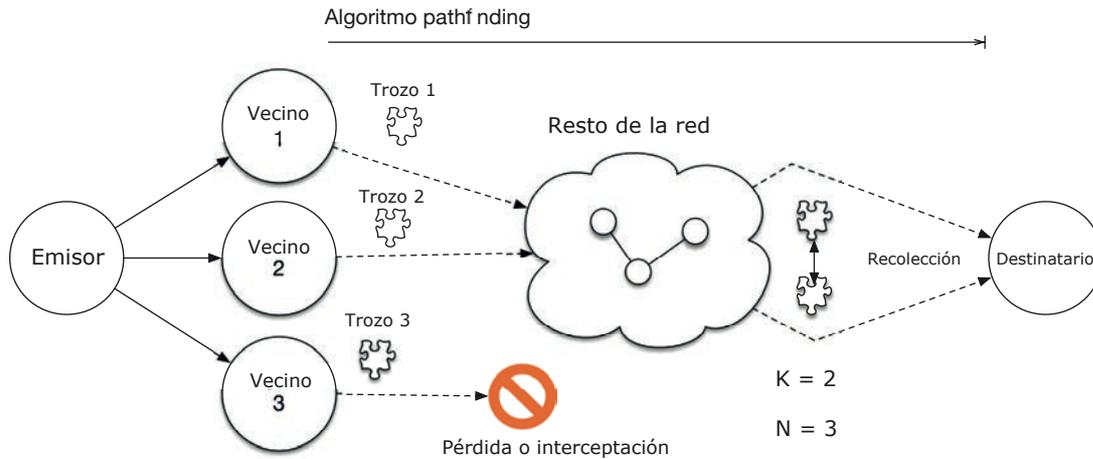


Figura A.12: Protocolo de compartición de secretos para el envío de mensajes

mayor para aquellos nodos que experimenten un valor de opinión más elevado según el algoritmo de Opinion Dynamics (ya que están sujetos a un ataque). Un esquema del funcionamiento de este protocolo queda ilustrado en la Figura A.12.

Para probar este sistema en la práctica, generamos en MATLAB diferentes topologías de red de 100, 200 y 300 nodos, sobre las que ejecutamos un conjunto de 50 ataques contra los nodos y aristas de la infraestructura generada. Para cada una de las tres estrategias de redundancia, generamos un paquete de 100 mensajes a distribuir por la red con un par emisor-destinatario aleatorio y, por último, contamos el porcentaje de paquetes perdidos. Como se aprecia en la Figura A.13 (que condensa los resultados de tales experimentos), la combinación de la información provista por Opinion Dynamics con una estrategia de redundancia (siendo suficiente con añadir enlaces adicionales a los nodos del DS), proporciona resultados significantes para proteger el envío de información de manera satisfactoria por la red.

Por otra parte, implementamos una segunda técnica en torno al encaminamiento de mensajes aprovechando la salida del algoritmo de Opinion Dynamics. Se trata de una técnica de encaminamiento para el envío fiable de información en redes cuyos canales poseen un Quality of Service (QoS) variable. Dicho de otro modo, lo que queremos es enviar la información a través de aquellos canales que ofrecen un mejor QoS (incluyendo ancho de banda, pocos retrasos, etc.) al mismo tiempo que consideramos también la seguridad de los nodos, tal como hemos procedido antes.

De esta manera, dotamos al *framework* la capacidad de evaluar no solo las características de seguridad de los nodos, sino también la calidad de servicio de los canales de comunicación. Para este cometido, en primer lugar, ideamos un indicador para representar la fiabilidad de un canal, en función del ancho de banda que posee, la cantidad de retrasos experimentados y la de paquetes perdidos. En base a este indicador, caracterizamos cada uno de los enlaces de una red representada por un grafo del tipo  $G(V, E)$  y enriquecemos el modelo original del algoritmo de Opinion Dynamics para modificar el procedimiento de asignación de pesos entre agentes. Lo

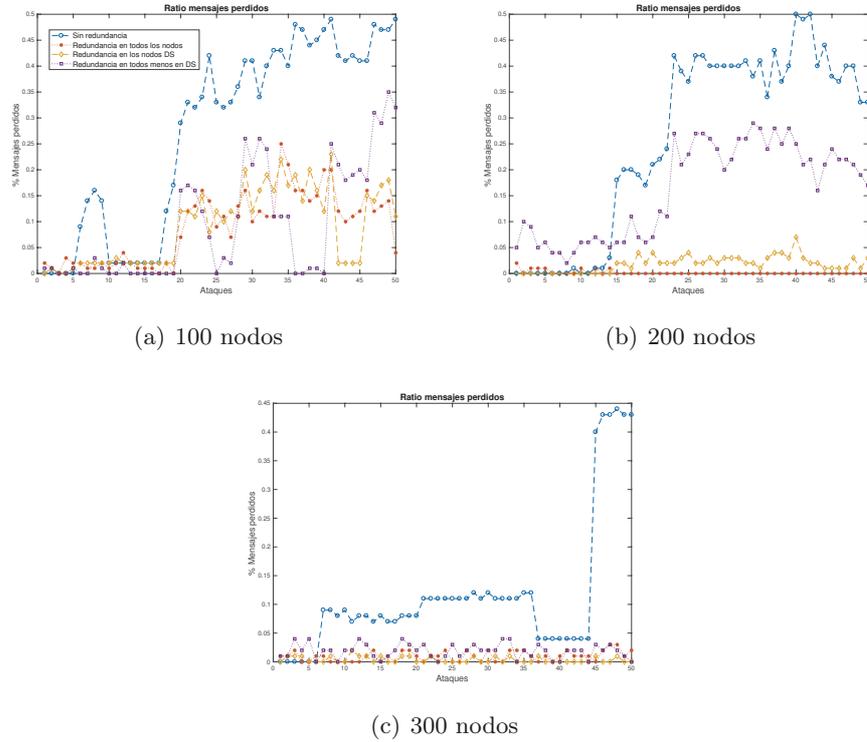


Figura A.13: Ratio de mensajes perdidos para las tres estrategias de redundancia, con 100 mensajes y 50 ataques sobre una red de 100, 200 y 300 nodos

que pretendemos es priorizar la correlación de anomalías entre aquellos agentes cuyos enlaces de comunicación poseen una mayor fiabilidad, al margen de la similitud entre opiniones debido a anomalías provocadas por problemas de seguridad. Con esto, la información resultante en torno a la seguridad de los nodos (con el algoritmo modificado de Opinion Dynamics) y la calidad de los enlaces (con el indicador mencionado anteriormente) es proporcionada a un protocolo de encaminamiento inspirado en el algoritmo de Bellman-Ford [293]. Este asegura que todos los agentes disponen de información sobre su vecindad, para luego determinar el camino descrito por los datos entre un emisor y destinatario, priorizando la seguridad y la QoS de los nodos y los enlaces intermedios.

Por último, llevamos a cabo distintas simulaciones con objeto de demostrar su efectividad en el envío de información por una red sujeta a distintas APT como las descritas para el anterior algoritmo de encaminamiento. El resultado es que el algoritmo efectivamente consigue maximizar la seguridad y QoS en comparación con aquellos algoritmos que calculan el camino más corto (como el de Dijkstra [278]) carentes de seguridad alguna, mientras que al mismo tiempo se acerca a los valores óptimos que se obtendrían si tales algoritmos priorizaran la seguridad de los nodos o la QoS de los enlaces por separado.

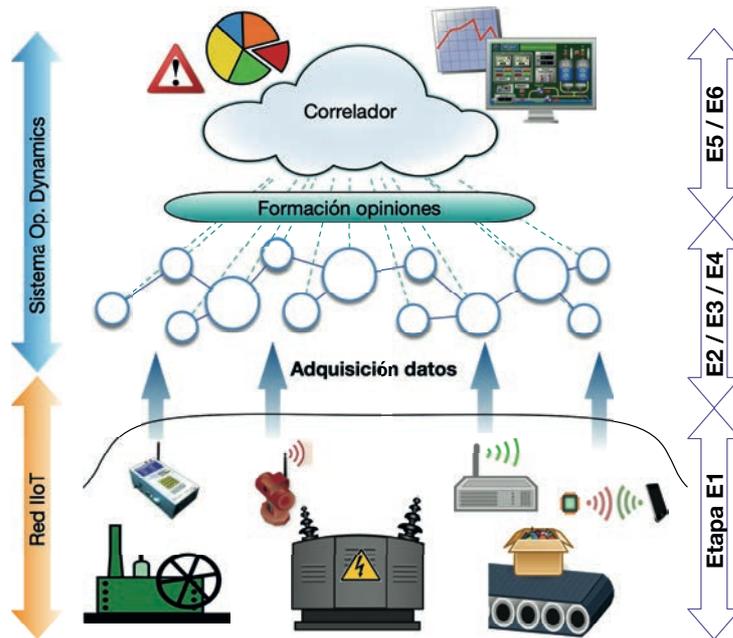


Figura A.14: Etapas para aplicar el *framework* de detección de APT

### A.5.2 Despliegue a un entorno de Internet de las Cosas Industrial

Debido a la profunda relación que guarda este paradigma con el concepto Industria 4.0 desde un punto de vista tecnológico y en cuanto a la problemática de seguridad, en la Sección 5.3 de la presente tesis doctoral abordamos un estudio de la aplicabilidad del *framework* de trazabilidad de APT a un entorno exclusivamente basado en el Internet de las Cosas Industrial. Este estudio es especialmente interesante para analizar ciertas limitaciones y cuestiones que deben abordarse. Por ejemplo, si el uso de una entidad centralizada como modelo de despliegue es una solución viable en todos los escenarios, o cómo instanciar con precisión los agentes de detección en una infraestructura física cuya criticidad puede restringir las modificaciones de hardware y software. Además, la posible sobrecarga generada en las comunicaciones o el aprovisionamiento de interfaces de red paralelas para recabar el tráfico de red son aspectos que resolvemos en este estudio.

Más concretamente, analizamos la instanciación del *framework* en una infraestructura IIoT, haciendo hincapié en la integración del algoritmo con la red a bajo nivel. Para ello, dividimos el proceso de aplicación de la técnica basada en Opinion Dynamics en seis etapas, tal como se muestra en la Figura A.14.

En la etapa 1, *configuración de la recuperación de datos*, el sistema extrae el tráfico y las salidas de posibles mecanismos de detección de anomalías. En este punto estudiamos cómo recoger el tráfico procedente de redes inalámbricas de ámbito local (WPAN, del inglés Wireless Personal Area Network) basadas en tecnologías como Bluetooth, de redes WLAN (Wireless Local Area Network) como las basadas en el estándar IEEE 802.11 o de redes celulares. Sobre ellas nos encontramos protocolos de comunicación IoT basados en MQTT (MQ Telemetry Transport)

o basados en interfaces tipo REST (Representational State Transfer). El objetivo consiste en procesar parámetros de red que nos permitan crear un modelo topológico de la infraestructura y obtener información suficiente para alimentar el algoritmo de correlación.

En la etapa 2, *creación de agentes*, todos los datos asociados a un dispositivo concreto se asignan a su correspondiente agente de detección (independientemente del modelo de despliegue adoptado). A partir de los datos brutos no extraídos de los IDS existentes, como el tráfico de red, se extraen diversas características (por ejemplo, el tipo de conexiones establecidas, número de paquetes intercambiados o los comandos ejecutados) en la etapa 3, *extracción de características*.

En la etapa 4, *selección de características y formación de opinión*, cada agente  $i$  combina en un momento dado  $t$  todos los datos disponibles en la opinión  $x_i(t)$ , (el estado de seguridad de su nodo monitorizado). Para esta tarea consideramos diferentes modelos para ponderar cada característica en función del escenario de seguridad actual y de las anomalías detectadas. Tras esto, la *correlación de opiniones* se efectúa tras esto en la fase 5, siguiendo el funcionamiento del algoritmo de Opinion Dynamics ya introducido.

Como resultado de esta correlación, obtenemos información del estado de seguridad en la etapa 6, *cálculo de indicadores*. Tal como se dispuso con la formalización del *framework*, esto incluye una representación de los segmentos de red afectados por ataques además de la visualización de la potencial APT a lo largo de su ciclo de vida.

En última instancia, para ilustrar la aplicabilidad este estudio y demostrar los beneficios de un despliegue conceptual del *framework* en un entorno IIoT, implementamos un caso de uso teórico mediante simulaciones. Para ello, formalizamos una red compuesta por distintas secciones siguiendo la topología común de una red IIoT, empleando generadores de red específicos. El tráfico generado por esta red es recogido por una entidad centralizada encargada de ejecutar el algoritmo de correlación, que finalmente es capaz de hacer el seguimiento de una APT compuesta por diversas fases de ataque sobre dicha infraestructura.

#### A.5.3 Aplicabilidad en la Smart Grid

Como parte del estudio de la aplicabilidad de nuestro marco de trabajo en distintos sectores de la Industria 4.0, también hemos efectuado un estudio sobre las ventajas específicas que puede aportar a la red eléctrica inteligente (o la bien conocida Smart Grid). Con la integración de tecnologías de comunicación en estos entornos, se ha producido un cambio hacia un modelo de red más interactivo, interconectado y dinámico. Su principal ventaja es el flujo de información bidireccional entre los consumidores (a través de contadores inteligentes) y la compañía eléctrica, que permite que los usuarios puedan reducir el consumo eléctrico con más flexibilidad, y al mismo tiempo la empresa pueda mejorar su respuesta a la demanda de electricidad en tiempo real.

No obstante, un entorno tan heterogéneo como este (donde conviven muchos actores como compañías eléctricas, plantas de generación, distribuidoras, o los propios clientes) existen varios

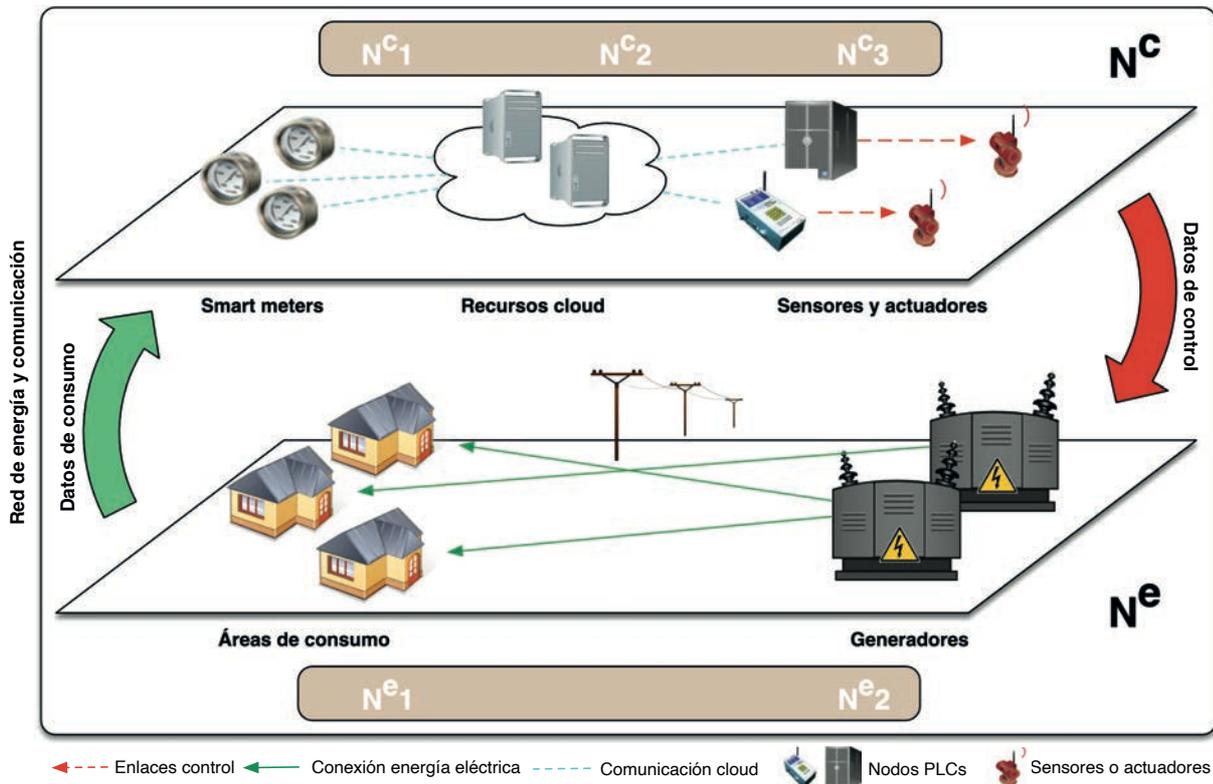


Figura A.15: Arquitectura resiliente para la red eléctrica inteligente

problemas de seguridad asociados, tal como sucede en cualquier entorno de la Industria 4.0 como los estudiados a lo largo de la tesis.

Nuestra primera contribución en este área consiste en la propuesta de una arquitectura resiliente que permita hacer frente a APT en la Smart Grid. Esta arquitectura, ilustrada en la Figura A.15, está basada en una infraestructura *cloud* capaz de realizar dos tareas: (i) la detección de amenazas y (ii) el balanceo de carga en la red, para de esta forma garantizar la seguridad y protección de la infraestructura ante fallos y ataques.

Esta arquitectura establece una división entre una red de comunicación y la propia red de energía, de cara a proporcionar esos dos servicios. Por un lado, un mecanismo de defensa capaz de detectar cambios a nivel hardware y software en los elementos de la red (véase *smart meters*, sensores, agregadores, etc.) y correlacionar anomalías, haciendo uso del algoritmo de Opinion Dynamics. Por otra parte, también se ofrece un servicio para el balanceo de carga en los generadores de electricidad disponibles en la red en cada momento. Este mecanismo permite hacer frente a potenciales picos de demanda que puedan provocar sobrecargas de la red (o incluso apagones), a través de una predicción de consumo en tiempo real. Para este cometido, se aplica un algoritmo de predicción basado en series temporales, que es entrenado previamente con un conjunto de datos de consumo a nivel estatal en España, a lo largo de un año completo (2015). De acuerdo al conocimiento obtenido, la predicción de consumo en un momento

determinado es proporcionada a otro algoritmo de satisfacción de restricciones que finalmente adjudica la carga eléctrica demandada entre un conjunto de generadores disponibles con su capacidad asociada. Ambos servicios son probados satisfactoriamente en sendas simulaciones que finalmente demuestran la efectividad del algoritmo de balanceo y la precisión a la hora de detectar las anomalías provocadas por una amenaza externa, de acuerdo a un modelo de atacante previamente definido y basado nuevamente en las fases de una APT.

Nuestra segunda contribución para la protección de la Smart Grid la proporcionamos con el soporte a los sistemas de autorización y control de acceso a los recursos de la red. Parte de este trabajo se realizó en el contexto de SealedGRID[317], un proyecto H2020 de la Unión Europea que aborda la protección de la Smart Grid ante ataques sofisticados, proporcionando una plataforma de seguridad escalable, de alta confianza e interoperable. En particular, aquí nos centramos en los servicios para gestionar los permisos de los distintos usuarios, dispositivos o procesos cuando solicitan acceder a los múltiples recursos dentro de la infraestructura. Y es que la integración de las tecnologías de la información y el *cloud* dificulta la aplicación de los modelos convencionales de control de acceso en los sistemas industriales (incluyendo la Smart Grid), debido al carácter descentralizado entre entidades con diferentes requisitos de acceso, rendimiento y de normativas. En este complejo escenario, los mecanismos de control de acceso desplegados han de restringir el acceso a cada entidad y las conexiones a aceptar, teniendo en cuenta el estado de seguridad de la red en todo momento.

Para este último propósito, ideamos un mecanismo de *context awareness* basado en el *framework* de trazabilidad de APT presentado en esta tesis. Siguiendo la idea original de recabar tráfico proveniente de toda la infraestructura para correlacionar anomalías e identificar amenazas, este mecanismo se integra en todos los puntos de decisión de políticas (PDP, en referencia a *Policy Decision Points*). Estos nodos son los encargados de aplicar las políticas de seguridad (expresadas en forma de reglas) de manera distribuida allí donde son desplegados, una vez los *Policy Enforcement Points* (PEPs) solicitan acceso a los recursos en nombre de los diversos dispositivos presentes.

En nuestro caso, lo que buscamos es proporcionar a los PDP la capacidad de llevar a cabo decisiones basadas no solo en las políticas de control de acceso, sino también del estado de seguridad de los nodos involucrados en tal acceso. Para ello consideramos el despliegue de un conjunto de agentes de detección dispersos por la red de manera jerárquica que reportan la información a sus respectivos PDP asignados, y que pueden abarcar regiones locales (para un conjunto de viviendas con sus contadores inteligentes, por ejemplo) o tener un ámbito de aplicación más global (para distintas ciudades o provincias a nivel estatal). Esta idea de agentes en este entorno se alinea con los denominados PIP (*Policy Information Points*), que asumimos que coexisten con los dispositivos de campo y suministran información acerca de los activos de la red a los distintos PDP con objeto de aplicar las políticas. Como resultado, obtenemos información útil acerca del estado de la red en función del algoritmo de Opinion Dynamics (mostrado en la Figura A.16), que se puede aprovechar para implementar políticas de autorización flexibles.

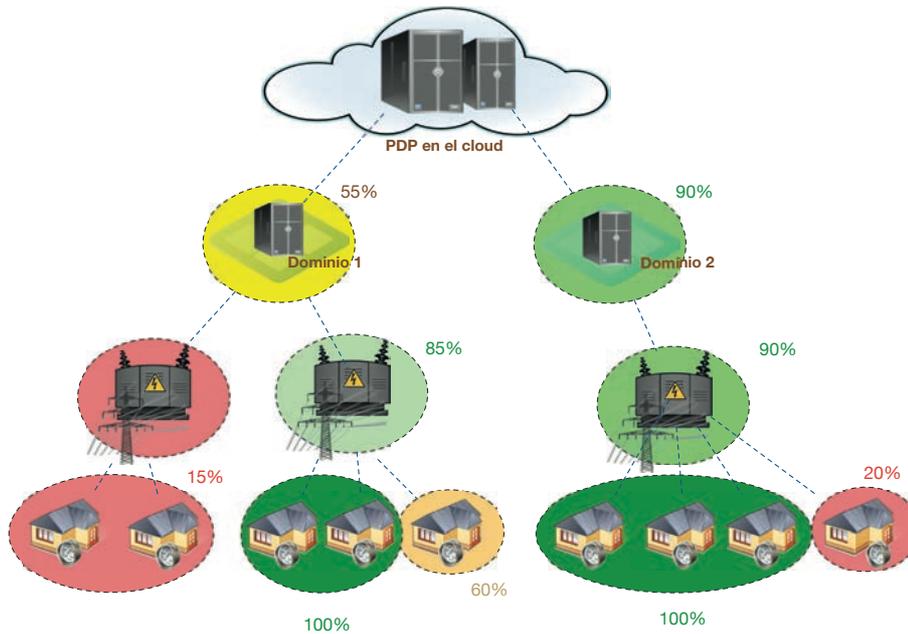


Figura A.16: Fragmentación de la Smart Grid en función de la información sobre amenazas provista por el algoritmo de Opinion Dynamics

Yendo un paso más allá, en este estudio también investigamos el uso de técnicas de aprendizaje para el refinamiento de las reglas de autorización dependiendo del comportamiento de la red en términos de seguridad. Al mismo tiempo, exploramos tangencialmente el concepto de gemelo digital, que nos permite generar un modelo virtual de todos los activos de la red al completo y probar diversos comportamientos sin comprometer el funcionamiento de la infraestructura real.

## A.6 Experimentación y validación de las soluciones propuestas

Después de evidenciar las ventajas del marco de trazabilidad de APT aplicando la solución basada en Opinion Dynamics a algunos escenarios de la Industria 4.0, en el Capítulo 6 de esta tesis doctoral llevamos a cabo la validación y verificación de todos estos resultados, incluyendo los algoritmos de correlación y las técnicas de respuesta desarrolladas. Ambos procesos son diferentes en su definición y objetivo: mientras que la *validación* se refiere a garantizar que el sistema satisface las necesidades del cliente o del usuario, la *verificación* consiste en evaluar si el sistema cumple con los requisitos impuestos originalmente (en nuestro caso, la especificación del *framework*).

En primer lugar, la verificación se realiza mediante demostraciones teóricas de cada enfoque presentado en esta tesis relacionado con la detección y trazabilidad de APT. Esto abarca la detección basada en *clustering* y la técnica inspirada en Opinion Dynamics, así como las técnicas de respuesta en forma de protocolos de encaminamiento seguros. Además de ello, evaluamos

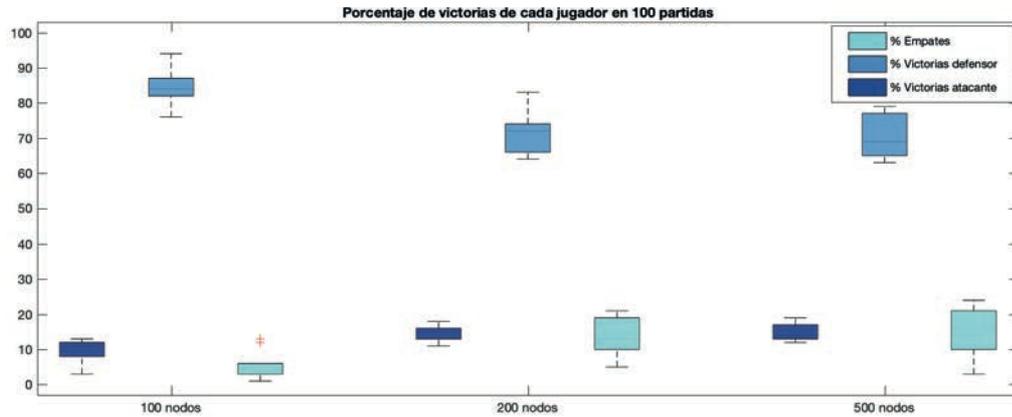


Figura A.17: Porcentaje de victorias para cada jugador en 100 partidas simuladas

distintas estrategias de respuesta contra las APT utilizando la información proporcionada por las soluciones de Opinion Dynamics, a través de teoría de juegos (del inglés, *game theory*).

Para este último cometido, formalizamos la especificación de un juego de dos jugadores (atacante y defensor) sobre un tablero que representa la red industrial, proporcionando un modelo para los movimientos de ambos jugadores. Mientras que el atacante imita el comportamiento definido por una APT (con distintos movimientos según la vulnerabilidad de los recursos y los objetivos fijados), el defensor hace uso del algoritmo de Opinion Dynamics para identificar los nodos comprometidos en la red y erradicar el ataque con diversas medidas de respuesta. Para ello, puede valerse de mecanismos como redundancia en los enlaces de comunicación (en caso de que el atacante realice una denegación de servicio contra ellos), la sustitución de nodos afectados o el uso de *honeypots* dentro de la red que sirvan como señuelo para aprender las capacidades del atacante.

Todas estas características quedan recogidas en un conjunto de reglas que formalizamos de manera igualitaria para ambos jugadores, definiendo una puntuación asociada a cada uno de los movimientos elegibles en el juego (basado en turnos) y varias condiciones de terminación para decidir si el atacante gana la partida (en caso de ejecutar todas las fases de la APT), si el ganador por contra es el defensor (si consigue exterminar la infección de la infraestructura antes de llevar a cabo su fase de ataque final), o si en cambio hay un empate. Este escenario se produce cuando la APT consigue efectuar satisfactoriamente todas sus fases, pero el defensor consigue llevar a cabo el seguimiento de la amenaza en todo momento, sin perder su rastro.

Con esta premisa, ejecutamos diversos experimentos con numerosas partidas para evaluar la efectividad del sistema de detección. La conclusión es que la solución de trazabilidad se postula como una solución efectiva cuando se combina con técnicas de respuesta adicionales como las presentadas anteriormente. Esto queda patente en las diversas estadísticas recabadas tras los experimentos, como la media de victorias conseguidas por el defensor, que aparecen ilustradas

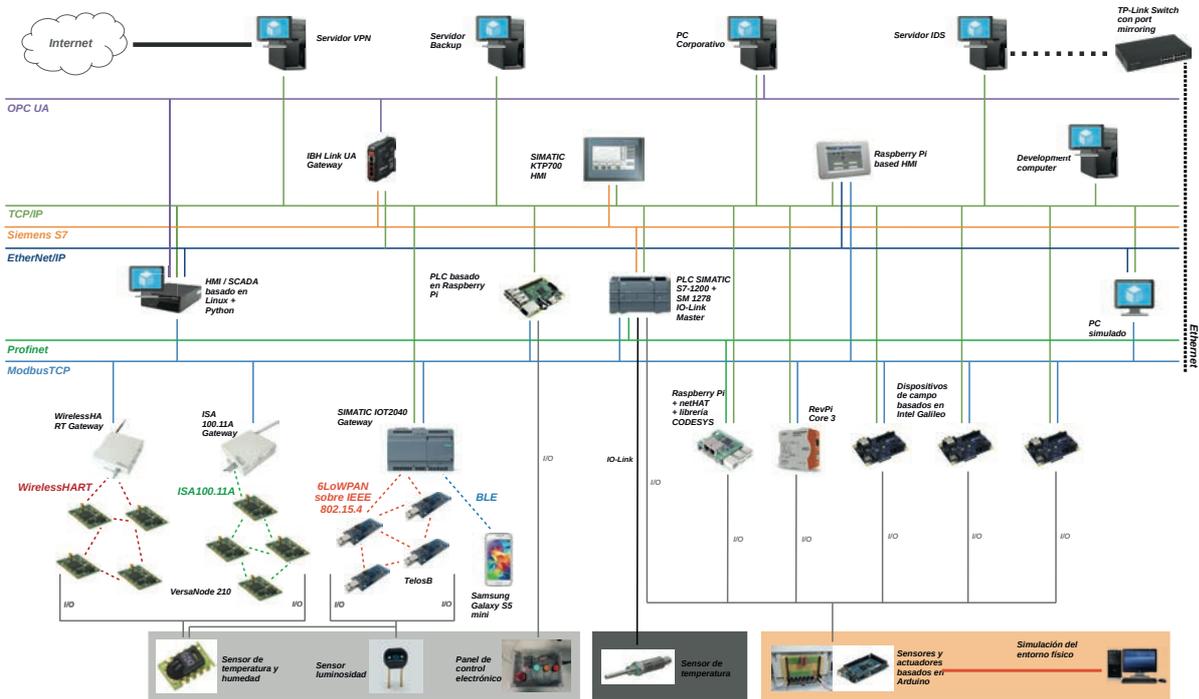


Figura A.18: Diagrama de red del entorno de pruebas industrial *I4Testbed*

en la Figura A.17. Así, extraemos información valiosa para determinar la estrategia de defensa óptima ante amenazas persistentes avanzadas.

Por otro lado, la validación del *framework* propuesto en esta tesis la realizamos desde un punto de vista práctico, implementando una prueba de concepto en un entorno de pruebas real, que integra varios tipos de dispositivos y protocolos industriales. Nos referimos a *I4Testbed*, una *testbed* industrial implementada en la Universidad de Málaga que simula el proceso de generación de electricidad en una central solar e hidroeléctrica, y que está provista de distintos dispositivos como los que habitualmente se encuentran en una infraestructura actual de estas características, tal como se muestra en la Figura A.18. Entre ellos, hay un servidor con capacidad computacional suficiente como para ejecutar el sistema de detección basado en el algoritmo de Opinion Dynamics haciendo uso de un modelo de despliegue centralizado.

En base a esta infraestructura, procedemos con la implementación y despliegue de los agentes virtuales que conforman el marco de trabajo para la detección y trazabilidad de amenazas. En primera instancia, redirigimos todo el tráfico de la red (por medio de un *switch* configurado en modo *port mirroring*) a cada uno de los agentes de detección (ejecutados de forma concurrente en el servidor mencionado anteriormente) para monitorizar la seguridad del dispositivo físico. Con esto se computa el valor actual de distintos indicadores de host y de red (como el uso de CPU, ancho de banda, conexiones establecidas con otros dispositivos, etc.) y se compara con el valor

A.6. Experimentación y validación de las soluciones propuestas

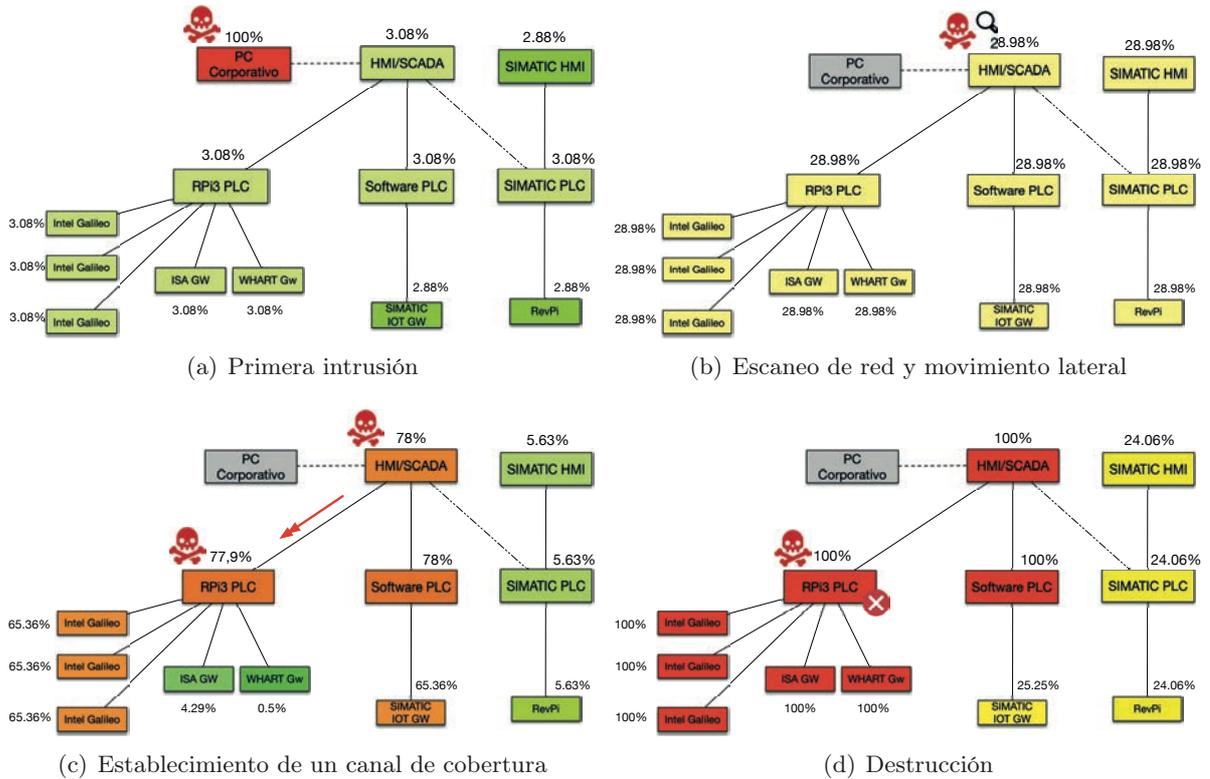


Figura A.19: Evolución de los valores de Opinion Dynamics en las fases de ataque

obtenido en condiciones normales, que ha sido previamente computado y aprendido. Por último, las opiniones de los agentes son correlacionadas haciendo uso del algoritmo de Opinion Dynamics, lo que nos da información útil sobre las anomalías sufridas a lo largo y ancho de la red.

Para la visualización de estas anomalías con un caso de uso real, a continuación procedemos a simular una APT de forma realista a través de distintas fases donde se perpetran ataques contra varios dispositivos de la red. En concreto, desarrollamos cuatro fases de ataque:

1. **Primera intrusión:** se perpetra un acceso inicial a la red. El adversario (potencialmente un *insider*) roba algunas credenciales de acceso (por ejemplo, con un ataque de ingeniería social) y se apodera del HMI o el SCADA accediendo a él desde la red de IT a través de SSH.
2. **Escaneo de red y movimiento lateral:** una vez obtenidos los privilegios y comprometido el sistema SCADA, el atacante realiza un reconocimiento del entorno buscando los servicios vulnerables que se ejecutan en cada dispositivo. Esto se consigue emitiendo un comando *nmap* en Linux. En este punto, suponemos que se encuentra una vulnerabilidad en el PLC basado en Raspberry Pi y se toma el control de ese nodo.

3. **Establecimiento de un canal de cobertura:** después de comprometer el PLC, el adversario establece un canal encubierto a través de la conexión Modbus. A través de este canal, el adversario envía un comando de apagado transcurrido un tiempo.
4. **Destrucción:** finalmente, el PLC ejecuta la orden de apagado y se desconecta del resto de dispositivos.

La Figura A.19 muestra una representación abstracta de los dispositivos involucrados en esta APT y sus conexiones, junto con la respectiva opinión de todos los agentes virtuales tras ejecutar cada una de estas fases de ataque. Nuevamente y esta vez comprobado en un entorno práctico, esta correlación proporcionan información útil para los servicios de seguridad de la red, ya que proporciona una herramienta de visualización precisa para identificar fácilmente los recursos más afectados en cada momento.

## A.7 Conclusiones y trabajo futuro

En esta tesis doctoral hemos abordado la detección y la trazabilidad de amenazas persistentes avanzadas en la Industria 4.0. Esta problemática surge debido al incremento de tecnologías de la información y de las comunicaciones en entornos industriales tradicionalmente desprovistos de medidas de seguridad. Este nuevo modelo sugiere interconectar todas las entidades involucradas en un proceso industrial para ganar flexibilidad en todo el proceso productivo, con el detrimento de las numerosas amenazas de seguridad que aparecen en estos nuevos servicios y que demandan soluciones cada vez más avanzadas. En particular, es de especial interés la implementación de mecanismos contra las llamadas amenazas persistentes avanzadas, compuestas por ataques muy sofisticados que desafían los sistemas de protección actuales.

En primer lugar, comenzamos explorando el contexto tecnológico de la Industria 4.0 para estudiar el espectro completo de ataques a los que está expuesto un sistema de control industrial y que pueden formar parte de una APT. A la luz de este estudio, se hizo un análisis del estado del arte para identificar sistemas de detección de intrusiones que permitieran poner en marcha una primera línea de defensa, tras lo que dedujimos que existe una carencia de mecanismos que permitan la monitorización precisa de las APT en infraestructuras modernas. En especial, pusimos el foco en la investigación de técnicas de detección que permitan monitorizar los recursos industriales de forma holística y detectar simultáneamente multitud de ataques, como en el caso de las APT.

Para atajar este problema, extraímos un conjunto de requisitos de detección y seguridad que deben cumplir las soluciones de detección en este campo. Estos conforman un marco de trazabilidad de APT que define la interfaz de entrada, los potenciales modelos de despliegue y los servicios que deben satisfacer distintos algoritmos de correlación de anomalías en una infraestructura industrial para conseguir detectar los movimientos de una APT, cuyo modelo de atacante modelamos. Para esta tarea realizamos un estudio sobre las APT más relevantes de

la última década y extraímos un modelo de comportamiento común basado en fases de ataque genéricas. Después, para ilustrar las ventajas de este marco, desarrollamos dos técnicas diferentes basadas en consenso distribuido y el *clustering*, y que llevan a cabo la correlación distribuida de las anomalías detectadas por un conjunto de agentes desplegados por la red. Tras una primera comparación de estas soluciones a través de experimentos que evalúan su precisión en la detección de un conjunto de APT modeladas teóricamente, encontramos que la técnica Opinion Dynamics es la más flexible y precisa. Este algoritmo simula la influencia de las opiniones entre un conjunto de agentes (que en nuestro caso representa la anomalía percibida por cada uno en su entorno local) y su evolución en el tiempo. Cuando estas opiniones son agrupadas y cada agente es influenciado por sus vecinos, podemos extraer información valiosa para determinar en qué partes de la red se encuentra el atacante y evaluar el estado de la infraestructura, cumpliendo así los requisitos inicialmente propuestos.

Nuestro objetivo a continuación fue comprobar la eficacia del marco de trazabilidad (y, por tanto, la de las soluciones que lo satisfacen) en diversos escenarios de la Industria 4.0 desde un punto de vista más práctico. En primer lugar, ideamos técnicas de respuesta que utilizan la información proporcionada por el sistema de detección para asegurar la supervivencia de la red, garantizando la continuidad de las comunicaciones en presencia de una APT. Esto se implementó utilizando protocolos de encaminamiento de mensajes que hacen uso de la información proporcionada por el algoritmo Opinion Dynamics, y luego se probaron con varios escenarios de ataque. Por otro lado, también se estudió su aplicación en un entorno de Internet de las Cosas Industrial y en la Smart Grid, como caso de uso del sector de la Industria 4.0. Para esto último se desarrolló una herramienta que previene contra posibles sobrecargas en la red y monitoriza las anomalías para proporcionar información sobre la seguridad de los recursos. Estos datos se utilizaron, además, para reforzar políticas de control de acceso basadas en el estado de la infraestructura en tiempo real.

En consecuencia, esta investigación es de especial interés para concienciar sobre la problemática de seguridad que rodea a las infraestructuras críticas que controlan nuestra sociedad. En particular, este marco de trazabilidad proporciona una guía para el diseño de sistemas de detección adaptados a la complejidad y heterogeneidad tecnológica de estos entornos. Así lo demuestran los diversos experimentos realizados, que ponen de manifiesto la precisión y eficacia de estas soluciones para la toma de decisiones, la prevención de riesgos y, en definitiva, la reducción del impacto (y, por tanto, de los costes) provocado por las APT.

A pesar del trabajo desarrollado en esta tesis doctoral, todavía hay varios retos y problemas abiertos que merece la pena explorar. Para empezar, sería factible ampliar el marco de trazabilidad considerando el procesamiento de más datos de entrada, de forma que lleváramos a cabo una correlación de anomalías más metódica y precisa. En este sentido, sería ideal abordar de forma práctica el procesamiento automatizado (y en tiempo real) de fuentes de información externas (como los informes de inteligencia sobre amenazas) para mantener una base de conocimientos actualizada acerca de la causalidad de los eventos (referido como entrada cualitativa en el contexto

del *framework* de trazabilidad). En el caso particular del algoritmo Opinion Dynamics, esto se traduciría en una asignación óptima y automática de pesos entre los agentes. Hasta ahora hemos dejado la puerta abierta a que estos valores se establezcan manualmente en base a diferentes reglas; por ejemplo, midiendo la calidad del servicio en las comunicaciones. Sin embargo, sería ideal que la plataforma en cuestión tuviera la capacidad autónoma de adquirir este conocimiento y calcular la mejor asignación, quizás utilizando técnicas de *machine learning*. En general, y en relación con esto último, sería interesante investigar la influencia de la inteligencia artificial en la correlación de anomalías, más allá de los mecanismos basados en el consenso distribuido y el *clustering*.

De igual manera, el sistema de trazabilidad podría ofrecer funcionalidades adicionales más allá de evaluar el estado actual de los dispositivos e identificar las zonas más afectadas. Por ejemplo, nos gustaría estudiar la posibilidad de realizar predicciones con los datos recogidos, para anticipar con certeza los próximos movimientos de una APT dentro de la red víctima. También es indispensable examinar el impacto sobre el rendimiento de los diferentes modelos de despliegue, para extraer conclusiones sobre las bondades de la correlación centralizada o distribuida. Al mismo tiempo, es de esperar que los algoritmos que satisfacen el propio marco de trazabilidad no presenten una alta complejidad, para no poner en peligro las restricciones de tiempo real impuestos por estos entornos tan críticos. Todo ello depende de un conjunto de requisitos de rendimiento que son específicos de los sistemas industriales donde se aplicarían estas soluciones, y que es diferente en cada sector de la Industria 4.0. En esta investigación se ha abordado el caso particular de la Smart Grid y entornos IIoT. Sin embargo, nos gustaría estudiar el comportamiento del algoritmo Opinion Dynamics (junto con otras soluciones alternativas) en entornos adicionales como la red de transporte o las telecomunicaciones, con el fin de identificar más parámetros que podrían ser contemplados por nuestro marco para finalmente caracterizar nuestras soluciones con un mayor grado de precisión.

Por último, es de esperar que los nuevos servicios de la Industria 4.0 habilitados por los nuevos paradigmas de computación y de las comunicaciones (como la blockchain, el 5G o fog/edge computing) sigan en aumento, propiciando la aparición de ataques que obliguen a renovar las técnicas existentes para adquirir un enfoque de detección aún más amplio. En consecuencia, es crucial seguir investigando mecanismos de detección adaptativos con un grado de autonomía cada vez más elevado, basados en modelos de atacante cada vez más complejos reconocidos en la industria y en el ámbito académico. Este proceso debe estar alineado con los estándares que integren la seguridad por defecto en todo el ciclo de vida de la Industria 4.0 (como IIRA y RAMI4.0), para facilitar la integración de las futuras soluciones de trazabilidad.

# Bibliography

- [1] Kaspersky. Kaspersky: Threat landscape for industrial automation systems: H1 2020. <https://ics-cert.kaspersky.com/reports/2020/09/24/threat-landscape-for-industrial-automation-systems-h1-2020/>, [Online; Accessed January 2021], 2020.
- [2] Accenture. Cost of cyber crime study. <https://www.accenture.com/us-en/insights/security/cost-cybercrime-study>, [Online; Accessed January 2021], 2019.
- [3] Consumption datasets. <https://www.entsoe.eu/data/data-portal/consumption/Pages/default.aspx>, last retrieved in August 2017.
- [4] R Davies. Industry 4.0. digitalisation for productivity and growth. *European Parliamentary Research Service, Briefing*, 2015.
- [5] Vidosav D Majstorovic and Radivoje Mitrovic. Industry 4.0 programs worldwide. In *International Conference on the Industry 4.0 model for Advanced Manufacturing*, pages 78–99. Springer, 2019.
- [6] Gianfranco Pedone and István Mezgár. Model similarity evidence and interoperability affinity in cloud-ready industry 4.0 technologies. *Computers in industry*, 100:278–286, 2018.
- [7] ICS-CERT. Overview of cyber vulnerabilities. <https://ics-cert.us-cert.gov/content/overview-cyber-vulnerabilities>, June 2017.
- [8] Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.
- [9] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1):1–22, 2019.
- [10] Hongyu Liu and Bo Lang. Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences*, 9(20):4396, 2019.

- 
- [11] Frédéric Rosin, Pascal Forget, Samir Lamouri, and Robert Pellerin. Impacts of industry 4.0 technologies on lean principles. *International Journal of Production Research*, 58(6):1644–1661, 2020.
- [12] Germany government. Platform industrie 4.0. <http://www.plattform-i40.de>, [Online; Accessed January 2021], 2021.
- [13] European Commission. European commission: examples of national initiatives for digitizing industry within the eu. <https://ec.europa.eu/growth/tools-databases/dem/monitor/category/national-initiatives>, [Online; Accessed January 2021], 2021.
- [14] AMETIC (Asociación Multisectorial de Empresas de Tecnologías de la Información. Comisión de industria 4.0. <https://ametic.es/es/areas-de-actuacion/innovacion-emprendimiento-e-internacionalizacion/comision-de-industria-40>, [Online; Accessed January 2021], 2021.
- [15] Comercio y Turismo Comisión de Industria 4.0 Ministerio de Industria. Industria conectada 4.0. <https://www.industriaconectada40.gob.es/Paginas/index.aspx>, [Online; Accessed January 2021], 2021.
- [16] European Commission. Eu public-private partnerships. “factories of the future”. [https://ec.europa.eu/research/industrial\\_technologies/factories-of-the-future\\_en.html](https://ec.europa.eu/research/industrial_technologies/factories-of-the-future_en.html), [Online; Accessed January 2021], 2021.
- [17] European Commission. The energy-efficient buildings ppp: research for low energy consumption buildings in the eu. [http://ec.europa.eu/research/press/2013/pdf/ppp/eeb\\_factsheet.pdf](http://ec.europa.eu/research/press/2013/pdf/ppp/eeb_factsheet.pdf), [Online; Accessed January 2021], 2018.
- [18] ECSEL. The ecseel joint undertaking. <https://www.ecsel.eu>, [Online; Accessed January 2021], 2021.
- [19] U.S. National Science Board. U.s. national science board: Recent trends in federal support for u.s. r&d. <https://nsf.gov/statistics/2018/nsb20181/report/sections/research-and-development-u-s-trends-and-international-comparisons/recent-trends-in-federal-support-for-u-s-r-d>, [Online; Accessed December 2018], 2018.
- [20] PR Newsire. The u.s. and china invest heavily in industry 4.0 technologies to be the world’s largest manufacturer. <https://www.prnewswire.com/news-releases/the-us-and-china-invest-heavily-in-industry-40-technologies-to-be-the-worlds-largest-manufacturer.html>, [Online; Accessed December 2018], 2018.
- [21] Industrial Internet Consortium. Industrial internet consortium. <http://www.iiconsortium.org>, [Online; Accessed January 2021], 2021.

- [22] Control Engineering. Made in china 2025. <https://www.controleng.com/articles/made-in-china-2025-chinese-government-aims-at-industry-4-0-implementation/>, [Online; Accessed December 2018], 2018.
- [23] Jane's 360. Chinese corporations announce industry 4.0 initiatives. <https://www.janes.com/article/80103/chinese-corporations-announce-industry-4-0-initiatives>, [Online; Accessed December 2018], 2018.
- [24] ChinaDaily. China to build 10 enterprise-level industrial internet platforms by 2020. <https://www.controleng.com/articles/made-in-china-2025-chinese-government-aims-at-industry-4-0-implementation>, [Online; Accessed December 2018], 2018.
- [25] Trade Ministry of Economy and Industry of Japan. Japan and germany to advance cooperation in field of industrial cybersecurity. [http://www.meti.go.jp/english/press/2018/0516\\_002.html](http://www.meti.go.jp/english/press/2018/0516_002.html), [Online; Accessed December 2018], 2018.
- [26] Industrial Value Chain Initiative (IVI). Industrial value chain initiative (ivi). <https://www.iv-i.org/en/>, [Online; Accessed December 2018], 2018.
- [27] Australian Government. Australia-germany advisory group: Collaboration, innovation & opportunity report. <https://www.dfat.gov.au/sites/default/files/progress-report-australia-germany-advisory-group.pdf>, [Online; Accessed January 2021], 2016.
- [28] Australian Government. Industry 4.0 funding and incentives. <https://www.industry.gov.au/funding-and-incentives/industry-40>, [Online; Accessed January 2021], 2021.
- [29] Industrial Internet Consortium. Plattform industrie 4.0 and industrial internet consortium agree on cooperation. <http://www.iiconsortium.org/press-room/03-02-16.htm>, [Online; Accessed January 2021], 2016.
- [30] Kris Bledowski. The internet of things: Industrie 4.0 vs. the industrial internet. <https://www.mapi.net/forecasts-data/internet-things-industrie-40-vs-industrial-internet>, [Online; Accessed November 2016], 2016.
- [31] Industrial Internet Consortium. IIRA Reference Architecture. Last accessed 31 August 2019.
- [32] Platform Industrie 4.0. RAMI4.0 Reference Architecture. Last accessed 31 August 2019.
- [33] Mohsen Moghaddam, Marissa N Cadavid, C Robert Kenley, and Abhijit V Deshmukh. Reference architectures for smart manufacturing: A critical review. *Journal of manufacturing systems*, 49:215–225, 2018.

- [34] Stefan-Helmut Leitner and Wolfgang Mahnke. Opc ua–service-oriented architecture for industrial applications. *ABB Corporate Research Center*, 48:61–66, 2006.
- [35] Sebastian R Bader and Ljiljana Stojanovic. Reference architecture models for smart service networks. In *Smart Service Management*, pages 131–136. Springer, 2020.
- [36] International Society of Automation. ISA-95 standard, 2017. <https://www.isa.org/isa95/>, last retrieved in December 2017.
- [37] Paulo Leitão, José Barbosa, Maria-Eleftheria Ch Papadopoulou, and Iakovos S Venieris. Standardization in cyber-physical systems: The arum case. In *IEEE International Conference on Industrial Technology (ICIT'15)*, pages 2988–2993, 2015.
- [38] Armando W Colombo, Stamatis Karnouskos, and Thomas Bangemann. Towards the next generation of industrial cyber-physical systems. In *Industrial cloud-based cyber-physical systems*, pages 1–22. Springer, 2014.
- [39] Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva. Security for the internet of things: a survey of existing protocols and open research issues. *IEEE Communications Surveys & Tutorials*, 17(3):1294–1312, 2015.
- [40] Radhakisan Baheti and Helen Gill. Cyber-physical systems. *The impact of control technology*, 12(1):161–166, 2011.
- [41] Xun Xu. From cloud computing to cloud manufacturing. *Robotics and computer-integrated manufacturing*, 28(1):75–86, 2012.
- [42] M. Chiang and T. Zhang. Fog and iot: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6):854–864, December 2016.
- [43] BMG. Put the power of local energy choice in your hands. <https://www.brooklyn.energy>, last access in April 2020, 2017.
- [44] C Burger, Andreas Kuhlmann, P Richard, and J Weinmann. Blockchain in the energy transition. a survey among decision-makers in the german energy industry. *DENA German Energy Agency*, 60, 2016.
- [45] Aitor Moreno, Gorka Velez, Aitor Ardanza, Iñigo Barandiaran, Álvaro Ruíz de Infante, and Raúl Chopitea. Virtualisation process of a sheet metal punching machine within the industry 4.0 vision. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, pages 1–9, 2016.
- [46] Y. Liao, E. de Freitas Rocha Loures, and F. Deschamps. Industrial Internet of Things: A Systematic Literature Review and Insights. *IEEE Internet of Things Journal*, 5(6):4515–4525, 2018.

- [47] Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. Mqtt-s—a publish/subscribe protocol for wireless sensor networks. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, pages 791–798. IEEE, 2008.
- [48] Jianping Song, Song Han, Al Mok, Deji Chen, Mike Lucas, Mark Nixon, and Wally Pratt. Wirelesshart: Applying wireless technology in real-time industrial process control. In *2008 IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 377–386. IEEE, 2008.
- [49] Siemens. Industrial IoT Solutions: SIMATIC IoT. Last accessed 31 August 2019.
- [50] Jose A. Gutierrez, Edgar H. Callaway, and Raymond Barrett. *IEEE 802.15.4 Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensor Networks*. IEEE Standards Office, USA, 2003.
- [51] Zach Shelby and Carsten Bormann. *6LoWPAN: The wireless embedded Internet*, volume 43. John Wiley & Sons, 2011.
- [52] Diego Dujovne, Thomas Watteyne, Xavier Vilajosana, and Pascal Thubert. 6tisch: deterministic ip-enabled industrial internet (of things). *IEEE Communications Magazine*, 52(12):36–41, 2014.
- [53] Stig Petersen and Simon Carlsen. Wirelesshart versus isa100. 11a: The format war hits the factory floor. *IEEE Industrial Electronics Magazine*, 5(4):23–34, 2011.
- [54] Hussein Mroue, Abbass Nasser, Sofiane Hamrioui, Benoît Parrein, Eduardo Motta-Cruz, and Gilles Rouyer. Mac layer-based evaluation of iot technologies: Lora, sigfox and nb-iot. In *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, pages 1–5. IEEE, 2018.
- [55] Nitin Naik. Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http. In *2017 IEEE international systems engineering symposium (ISSE)*, pages 1–7. IEEE, 2017.
- [56] Steve Vinoski. Advanced message queuing protocol. *IEEE Internet Computing*, 10(6):87–89, 2006.
- [57] Peter Saint-Andre et al. Extensible messaging and presence protocol (xmpp): Core. 2004.
- [58] CC Sobin. A survey on architecture, protocols and challenges in iot. *Wireless Personal Communications*, 112(3):1383–1429, 2020.

- [59] Jorg Swetina, Guang Lu, Philip Jacobs, Francois Ennesser, and JaeSeung Song. Toward a standardized common m2m service layer platform: Introduction to onem2m. *IEEE Wireless Communications*, 21(3):20–26, 2014.
- [60] International Data Corporation. Iot and digital transformation: A tale of four industries. [http://www.digitalistmag.com/files/2016/03/IDC\\_IoT\\_white\\_paper\\_Mar2016.pdf](http://www.digitalistmag.com/files/2016/03/IDC_IoT_white_paper_Mar2016.pdf), [Online; Accessed December 2018], 2016.
- [61] Tim Stock and Günther Seliger. Opportunities of sustainable manufacturing in industry 4.0. *Procedia Cirp*, 40:536–541, 2016.
- [62] Gartner. Gartner top 6 trends impacting infrastructure & operations in 2021. <https://www.gartner.com/smarterwithgartner/gartner-top-6-trends-impacting-infrastructure-operations-in-2021/>, [Online; Accessed February 2021], 2020.
- [63] Dazhong Wu, Matthew John Greer, David W Rosen, and Dirk Schaefer. Cloud manufacturing: Strategic vision and state-of-the-art. *Journal of Manufacturing Systems*, 32(4):564–579, 2013.
- [64] Michael Grieves and John Vickers. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary perspectives on complex systems*, pages 85–113. Springer, 2017.
- [65] Shiyong Wang, Jiafu Wan, Daqiang Zhang, Di Li, and Chunhua Zhang. Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, 101:158–168, 2016.
- [66] Kaspersky Lab ICS CERT. Threat landscape for industrial automation systems. H2 2018. <https://ics-cert.kaspersky.com/reports/2019/03/27/threat-landscape-for-industrial-automation-systems-h2-2018/>, 2019. Last accessed on September 2019.
- [67] Juan E. Rubio, Rodrigo Roman, and Javier Lopez. Analysis of cybersecurity threats in industry 4.0: the case of intrusion detection. In *The 12th International Conference on Critical Information Infrastructures Security*, volume Lecture Notes in Computer Science, vol 10707, pages 119–130. Springer, Springer, 08/2018 2018.
- [68] Lorena Cazorla, Cristina Alcaraz, and Javier Lopez. Cyber stealth attacks in critical information infrastructures. *IEEE Systems Journal*, pages 1–15, March 2016.
- [69] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, Daesung Moon, and Jong Hyuk Park. A comprehensive study on apt attacks and countermeasures for future networks and communications: challenges and solutions. *The Journal of Supercomputing*, pages 1–32, 2016.

- [70] Ping Chen, Lieven Desmet, and Christophe Huygens. A study on advanced persistent threats. In *IFIP International Conference on Communications and Multimedia Security*, pages 63–72. Springer, 2014.
- [71] Antoine Lemay, Joan Calvet, François Menet, and José M. Fernandez. Survey of publicly available reports on advanced persistent threat actors. *Computers & Security*, 72:26–59, 2018.
- [72] European Cyber Security Organisation (ECSO). Ecsocppp. <http://ecs-org.eu/documents/contract.pdf>, [Online; Accessed January 2021], 2016.
- [73] IBM X-Force. Threat intelligence index. <https://www.ibm.com/security/data-breach/threat-intelligence>, [Online; Accessed February 2021], 2020.
- [74] MITRE Corporation. MITRE ATT&CK. <https://attack.mitre.org>, 2018. [Online; Accessed May 2018].
- [75] European Union Agency for Cybersecurity (ENISA). Good practices for security of internet of things in the context of smart manufacturing. <https://www.enisa.europa.eu/publications/good-practices-for-security-of-iot>, [Online; Accessed January 2021], 2018.
- [76] Homeland Security. Strategic principles for securing the internet of things. [https://www.dhs.gov/sites/default/files/publications/Strategic\\_Principles\\_for\\_Securing\\_the\\_Internet\\_of\\_Things-2016-1115-FINAL....pdf](https://www.dhs.gov/sites/default/files/publications/Strategic_Principles_for_Securing_the_Internet_of_Things-2016-1115-FINAL....pdf), [Online; Accessed January 2021], 2016.
- [77] Keith Stouffer, Keith Stouffer, Timothy Zimmerman, CheeYee Tang, Joshua Lubell, Jeffrey Cichonski, and John McCarthy. *Cybersecurity framework manufacturing profile*. US Department of Commerce, National Institute of Standards and Technology, 2017.
- [78] AEGIS Consortium. Policy brief on research and innovation in cybersecurity. <https://aegis-project.org/wp-content/uploads/2019/06/Policy-Brief-on-Research-and-Innovation-in-Cybersecurity-Updated-May-2019.pdf>, [Online; Accessed January 2021], 2019.
- [79] WG15 of IEC TC57. Information security for power system control operations, international electrotechnical commission, IEC 62351. <http://www.iec.ch/smartgrid/standards/>, [Online; Accessed January 2021], 2017.
- [80] Björn Leander, Aida Čaušević, and Hans Hansson. Applicability of the iec 62443 standard in industry 4.0/iiot. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–8, 2019.

- [81] D. E. Whitehead, K. Owens, D. Gammel, and J. Smith. Ukraine cyber-induced power outage: Analysis and practical mitigation strategies. In *70th Annual Conference for Protective Relay Engineers (CPRE'17)*, pages 1–8, April 2017.
- [82] Juan E. Rubio, Rodrigo Roman, Cristina Alcaraz, and Yan Zhang. Tracking apts in industrial ecosystems: A proof of concept. *Journal of Computer Security*, 27:521–546, 09/2019 2019.
- [83] Federal Office for information Security. Industrial Control System Security: Top 10 Threats and Countermeasures 2016. <https://www.allianz-fuer-cybersicherheit.de>, 2016. [Online; Accessed May 2018].
- [84] ICS-CERT. Overview of Cyber Vulnerabilities. <http://ics-cert.us-cert.gov/content/overview-cyber-vulnerabilities>, 2016. [Online; Accessed May 2018].
- [85] Bonnie Zhu, Anthony Joseph, and Shankar Sastry. A taxonomy of cyber attacks on scada systems. In *Internet of things (iThings/CPSCOM), 2011 international conference on and 4th international conference on cyber, physical and social computing*, pages 380–388. IEEE, 2011.
- [86] Symantec. Protecting critical systems while promoting operational efficiency. Technical report, 2012. [Online; Accessed May 2018].
- [87] IBM® X-Force® Research. 2016 Cyber Security Intelligence Index: A survey of the cyber security landscape for financial services. [http://www.ciosummits.com/2016\\_Cyber\\_Security\\_Intelligence\\_Index\\_for\\_Fnl\\_Svcs.pdf](http://www.ciosummits.com/2016_Cyber_Security_Intelligence_Index_for_Fnl_Svcs.pdf), 2016. [Online; Accessed May 2018].
- [88] Sikich. 2016 Manufacturing Report, Taking your business to the next level and ensuring a successful future. [https://www.leadingedgealliance.com/thought\\_leadership/sikich\\_manufacturing\\_report\\_2016r.pdf](https://www.leadingedgealliance.com/thought_leadership/sikich_manufacturing_report_2016r.pdf), 2016. [Online; Accessed May 2018].
- [89] T Tsao, R Alexander, M Dohler, V Daza, A Lozano, and M Richardson. A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs). Technical report, 2015.
- [90] Cristina Alcaraz and Javier Lopez. A security analysis for wireless sensor mesh networks in highly critical systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(4):419–428, July 2010.
- [91] Andreas Moser, Christopher Kruegel, and Engin Kirda. Exploring multiple execution paths for malware analysis. In *IEEE Symposium on Security and Privacy (SP'07)*, pages 231–245. IEEE, 2007.

- [92] A. Khan and K. Turowski. A survey of current challenges in manufacturing industry and preparation for industry 4.0. In *First International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'16)*, pages 15–26. Springer International Publishing, 2016.
- [93] Federal Office for information Security. Industrial control system security: Top 10 threats and countermeasures 2016. [https://www.allianz-fuer-cybersicherheit.de/ACS/DE/\\_/downloads/BSI-CS\\_005E.pdf?\\_\\_blob=publicationFile&v=3](https://www.allianz-fuer-cybersicherheit.de/ACS/DE/_/downloads/BSI-CS_005E.pdf?__blob=publicationFile&v=3), June 2017.
- [94] Linus Wallgren, Shahid Raza, and Thiemo Voigt. Routing Attacks and Countermeasures in the RPL-based Internet of Things. *International Journal of Distributed Sensor Networks*, 2013.
- [95] Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. Security and privacy challenges in industrial internet of things. In *Proceedings of the 52Nd Annual Design Automation Conference, DAC '15*, pages 54:1–54:6, New York, NY, USA, 2015. ACM.
- [96] Kuan Zhang, Xiaohui Liang, Rongxing Lu, and Xuemin Shen. Sybil attacks and their defenses in the internet of things. *IEEE Internet of Things Journal*, 1(5):372–383, 2014.
- [97] Cédric Lévy-Bencheton, Louis Marinos, Rossella Mattioli, Thomas King, Christoph Dietzel, Stumpf Jan, et al. Threat landscape and good practice guide for internet infrastructure. *Report, European Union Agency for Network and Information Security (ENISA)*, 2015.
- [98] Kai Zhao and Lina Ge. A survey on the internet of things security. In *Computational Intelligence and Security (CIS), 2013 9th International Conference on*, pages 663–667. IEEE, 2013.
- [99] Dazhong Wu, David W Rosen, Lihui Wang, and Dirk Schaefer. Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation. *Computer-Aided Design*, 59:1–14, 2015.
- [100] Jaydip Sen. Security and privacy issues in cloud computing. *Architectures and Protocols for Secure Information Technology Infrastructures*, pages 1–45, 2013.
- [101] Yunchuan Sun, Junsheng Zhang, Yongping Xiong, and Guangyu Zhu. Data security and privacy in cloud computing. *International Journal of Distributed Sensor Networks*, 2014.
- [102] Govind Murari Upadhyay and Harsh Arora. Vulnerabilities of data storage security in big data. *IITM Journal of Management and IT*, 7(1):37–41, 2016.
- [103] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

- [104] Kristiina Valtanen, Jere Backman, and Seppo Yrjölä. Blockchain-powered value creation in the 5g and smart grid use cases. *IEEE Access*, 7:25690–25707, 2019.
- [105] C Mohan. State of public and private blockchains: Myths and reality. In *Proceedings of the 2019 International Conference on Management of Data*, pages 404–411, 2019.
- [106] Yang Xiao, Ning Zhang, Wenjing Lou, and Y Thomas Hou. A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, page 1–34, 2020.
- [107] Forbes. Bitcoin hit by ‘massive’ ddos attack as tensions rise. <https://www.forbes.com/sites/leoking/2014/02/12/bitcoin-hit-by-massive-ddos-attack-as-tensions-rise/?sh=15339cef246a>, last retrieved in February 2021.
- [108] Richard Greene and Michael N Johnstone. An investigation into a denial of service attack on an ethereum network. 2018.
- [109] Coindesk. The dao attacked: Code issue leads to \$60 million ether theft. <https://www.coindesk.com/dao-attacked-code-issue-leads-60-million-ether-theft>, last retrieved in February 2021.
- [110] Deloitte. Blockchain & cyber security. let’s discuss. <https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/technology-media-telecommunications/Blockchain-and-Cyber.pdf>, last retrieved in February 2021.
- [111] Alexander Egberts. The oracle problem-an analysis of how blockchain oracles undermine the advantages of decentralized ledger systems. *Available at SSRN 3382343*, 2017.
- [112] Huru Hasanova, Ui-jun Baek, Mu-gon Shin, Kyunghee Cho, and Myung-Sup Kim. A survey on blockchain cybersecurity vulnerabilities and possible countermeasures. *International Journal of Network Management*, 29(2):e2060, 2019.
- [113] T. Stock and G. Seliger. Opportunities of sustainable manufacturing in industry 4.0. *Procedia CIRP*, 40:536–541, 2016.
- [114] Industrial Data Space Association. Industrial data space: Reference architecture. <http://www.industrialdataspace.org/en/>, June 2017.
- [115] Dazhong Wu, Matthew John Greer, David W. Rosen, and Dirk Schaefer. Cloud manufacturing: Strategic vision and state-of-the-art. *Journal of Manufacturing Systems*, 32(4):564–579, 2013.

- [116] Shiyong Wang, Jiafu Wan, Daqiang Zhang, Di Li, and Chunhua Zhang. Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, 101:158–168, 2016.
- [117] Bernhards Blumbergs. Technical analysis of advanced threat tactics targeting critical information infrastructure. Technical report, 2014.
- [118] Adel Alshamrani, Sowmya Myneni, Ankur Chowdhary, and Dijiang Huang. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials*, 21(2):1851–1877, 2019.
- [119] ICS-CERT. Year in review 2015. <https://ics-cert.us-cert.gov>, last retrieved in February 2017.
- [120] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Shining light in dark places: Understanding the tor network. In *International symposium on privacy enhancing technologies symposium*, pages 63–76. Springer, 2008.
- [121] ISACA. State of cybersecurity 2020. <https://www.isaca.org/go/state-of-cybersecurity-2020>, last retrieved in February 2021.
- [122] J. Smit, S. Kreutzer, C. Moeller, and M. Carlberg. “industry 4.0”. *European Parliament, Directorate General for Internal Policies*, feb 2016.
- [123] Falliere, N., Murchu, L.O., Chien, E. W32.stuxnet dossier, version 1.4 (february 2011), 2011. <https://www.symantec.com>, last retrieved in April 2018.
- [124] Symantec Security Response Attack Investigation Team. Dragonfly: Western energy sector targeted by sophisticated attack group, 2017. <https://www.symantec.com>, last retrieved in April 2018.
- [125] SANS Industrial Control Systems. Analysis of the cyber attack on the ukrainian power grid, 2016. <https://ics.sans.org>, last retrieved in April 2018.
- [126] Cherepanov, A. Telebots are back – supply-chain attacks against ukraine, 2017. <https://www.welivesecurity.com>, last retrieved in April 2018.
- [127] Anton Cherepanov. Greyenergy white paper: A successor to blackenergy, Oct 2018. [https://www.welivesecurity.com/wp-content/uploads/2018/10/ESET\\_GreyEnergy.pdf](https://www.welivesecurity.com/wp-content/uploads/2018/10/ESET_GreyEnergy.pdf), last retrieved in February 2019.
- [128] Dragos. Global oil and gas cyber threat perspective. <https://www.dragos.com/wp-content/uploads/Dragos-Oil-and-Gas-Threat-Perspective-2019.pdf>, [Online; Accessed February 2021], 2019.

- [129] Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, 1(1), 2011.
- [130] Beazley. 2018 breach briefing whitepaper. 2018. <https://www.beazley.com/documents/Whitepapers/201802-beazley-breach-briefing.pdf>, last retrieved in February 2019.
- [131] Spain University of Malaga. Sadcip project. <https://www.nics.uma.es/projects/sadcip>, last retrieved in February 2017.
- [132] Animesh Patcha and Jung-Min Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks*, 51(12):3448–3470, 2007.
- [133] Monowar H Bhuyan, Dhruba Kumar Bhattacharyya, and Jugal K Kalita. Network anomaly detection: methods, systems and tools. *IEEE Communications Surveys & Tutorials*, 16(1):303–336, 2014.
- [134] Manasi Gyanchandani, JL Rana, and RN Yadav. Taxonomy of anomaly based intrusion detection system: a review. *International Journal of Scientific and Research Publications*, 2(12):1–13, 2012.
- [135] R Sekar, Ajay Gupta, James Frullo, Tushar Shanbhag, Abhishek Tiwari, Henglin Yang, and Sheng Zhou. Specification-based anomaly detection: a new approach for detecting network intrusions. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 265–274. ACM, 2002.
- [136] Cubix. Tippingpoint intrusion prevention system (ips). [http://cubixindia.com/index.php?option=com\\_content&view=article&id=12&Itemid=476](http://cubixindia.com/index.php?option=com_content&view=article&id=12&Itemid=476), [Online; Accessed May 2018], 2018.
- [137] NetSolution Store. Extreme networks intrusion prevention. <http://www.netsolutionstore.com/IPS.asp>, [Online; Accessed May 2018], 2018.
- [138] Corero. Corero network security. <https://www.corero.com>, [Online; Accessed May 2018], 2018.
- [139] Cristina Alcaraz and Sherali Zeadally. Critical infrastructure protection: Requirements and challenges for the 21st century. *International Journal of Critical Infrastructure Protection (IJCIP)*, 8:53–66, 01/2015 2015.
- [140] Cristina Alcaraz and Javier Lopez. Analysis of requirements for critical control systems. *International Journal of Critical Infrastructure Protection (IJCIP)*, 5:137–145, 2012 2012.

- [141] Bilal Maqbool Beigh, Uzair Bashir, and Manzoor Chahcoo. Article: Intrusion detection and prevention system: Issues and challenges. *International Journal of Computer Applications*, 76(17):26–30, August 2013.
- [142] Deris Stiawan, Abdul Hanan Abdullah, and Mohd. Yazid Idris. Article: Characterizing network intrusion prevention system. *International Journal of Computer Applications*, 14(1):11–18, January 2011. Full text available.
- [143] Cristina Alcaraz, Lorena Cazorla, and Javier Lopez. Cyber-physical systems for wide-area situational awareness. In *Cyber-Physical Systems: Foundations, Principles and Applications*, number Intelligent Data-Centric Systems, chapter 20, pages 305 – 317. Academic Press, Boston, 2017 2017.
- [144] Lorena Cazorla, Cristina Alcaraz, and Javier Lopez. Awareness and reaction strategies for critical infrastructure protection. *Computers and Electrical Engineering*, 47:299–317, 2015.
- [145] NIST. Guidelines for smart grid cybersecurity - volume 1 - smart grid cybersecurity strategy, architecture, and high-level requirements. NISTIR 7628 Rev 1., 2014.
- [146] Evan R Sparks. A security assessment of trusted platform modules. 2007.
- [147] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. Trusted execution environment: what it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 57–64. IEEE, 2015.
- [148] M. Krotofil and D. Gollmann. Industrial control systems security: What is happening? In *11th IEEE International Conference on Industrial Informatics (INDIN'13)*, pages 670–675, July 2013.
- [149] Chih-Yuan Lin, Simin Nadjm-Tehrani, and Mikael Asplund. Timing-based Anomaly Detection in SCADA Networks. In *12th International Conference on Critical Information Infrastructures Security (CRITIS'17)*, Oct 2017.
- [150] S. Ponomarev and T. Atkison. Industrial control system network intrusion detection by telemetry analysis. *IEEE Transactions on Dependable and Secure Computing*, 13(2):252–260, March 2016.
- [151] G. Lontorfos, K. D. Fairbanks, L. Watkins, and W. H. Robinson. Remotely inferring device manipulation of industrial control systems via network behavior. In *IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops'15)*, pages 603–610, Oct 2015.
- [152] Rahul Nair, Chinmohan Nayak, Lanier Watkins, Kevin D. Fairbanks, Kashif Memon, Pengyuan Wang, and William H. Robinson. *The Resource Usage Viewpoint of Industrial Control System Security: An Inference-Based Intrusion Detection System*, pages 195–223. Springer International Publishing, 2017.

- [153] Samuel J. Stone, Michael A. Temple, and Rusty O. Baldwin. Detecting anomalous programmable logic controller behavior using rf-based hilbert transform features and a correlation-based verification process. *International Journal of Critical Infrastructure Protection*, 9:41 – 51, 2015.
- [154] Yu-jun Xiao, Wen-yuan Xu, Zhen-hua Jia, Zhuo-ran Ma, and Dong-lian Qi. NIPAD: a non-invasive power-based anomaly detection scheme for programmable logic controllers. *Frontiers of Information Technology & Electronic Engineering*, 18(4):519–534, Apr 2017.
- [155] C. McParland, S. Peisert, and A. Scaglione. Monitoring security of networked control systems: It’s the physics. *IEEE Security Privacy*, 12(6):32–39, Nov 2014.
- [156] Marco Caselli, Emmanuele Zambon, Johanna Amann, Robin Sommer, and Frank Kargl. Specification mining for intrusion detection in networked control systems. In *25th USENIX Security Symposium*, pages 791–806. USENIX Association, 2016.
- [157] Herson Esquivel-Vargas, Marco Caselli, and Andreas Peter. Automatic Deployment of Specification-based Intrusion Detection in the BACnet Protocol. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and Privacy (CPS’17)*, pages 25–36. ACM, 2017.
- [158] Chen Markman, Avishai Wool, and Alvaro A. Cardenas. A New Burst-DFA Model for SCADA Anomaly Detection. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and Privacy (CPS’17)*, pages 1–12. ACM, 2017.
- [159] Marwa Keshk, Elena Sitnikova, Nour Moustafa, Jiankun Hu, and Ibrahim Khalil. An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems. *IEEE Transactions on Sustainable Computing*, 2019.
- [160] Wenbin Yu, Yiyin Wang, and Lei Song. A two stage intrusion detection system for industrial control networks based on ethernet/ip. *Electronics*, 8(12):1545, 2019.
- [161] Manuel Cheminod, Luca Durante, Lucia Seno, and Adriano Valenzano. Detection of attacks based on known vulnerabilities in industrial networked systems. *Journal of Information Security and Applications*, 34:153–165, 2017.
- [162] Mitre. Common Vulnerabilities and Exposures. <https://cve.mitre.org/>, 2018. [Online; Accessed May 2018].
- [163] Ching-Tai Lin. Structural controllability. *IEEE Transactions on Automatic Control*, 19(3):201–208, 1974.
- [164] Mohammad Amin Rahimian and Amir G Aghdam. Structural controllability of multi-agent networks: Robustness against simultaneous failures. *Automatica*, 49(11):3149–3157, 2013.

- [165] Martin Roesch et al. Snort: Lightweight intrusion detection for networks. In *Lisa*, volume 99, pages 229–238, 1999.
- [166] Vern Paxson et al. The Bro Network Security Monitor. <https://www.bro.org/>, 2018. [Online; Accessed May 2018].
- [167] K. Wong, C. Dillabaugh, N. Seddigh, and B. Nandy. Enhancing Suricata intrusion detection system for cyber security in SCADA networks. In *IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE'17)*, pages 1–5, April 2017.
- [168] Khurum Nazir Junejo and Jonathan Goh. Behaviour-based attack detection and classification in cyber physical systems using machine learning. In *Proceedings of the 2Nd ACM International Workshop on Cyber-Physical System Security (CPSS'16)*, pages 34–43, New York, NY, USA, 2016. ACM.
- [169] Hamid Reza Ghaeini, Daniele Antonioli, Ferdinand Brasser, Ahmad-Reza Sadeghi, and Nils Ole Tippenhauer. State-aware anomaly detection for industrial control systems. In *Proceedings of Security Track at the ACM Symposium on Applied Computing (SAC'18)*, April 2018.
- [170] William Jardine, Sylvain Frey, Benjamin Green, and Awais Rashid. SENAMI: Selective Non-Invasive Active Monitoring for ICS Intrusion Detection. In *Proceedings of the 2Nd ACM Workshop on Cyber-Physical Systems Security and Privacy (CPS-SPC'16)*, pages 23–34, New York, NY, USA, 2016. ACM.
- [171] Matthias Niedermaier, Martin Striegel, Felix Sauer, Dominik Merli, and Georg Sigl. Efficient intrusion detection on low-performance industrial iot edge node devices. *arXiv preprint arXiv:1908.03964*, 2019.
- [172] L. Garcia, S. Zonouz, Dong Wei, and L. P. de Aguiar. Detecting PLC control corruption via on-device runtime verification. In *2016 Resilience Week (RWS)*, pages 67–72, Aug 2016.
- [173] J. Hong and C. C. Liu. Intelligent electronic devices with collaborative intrusion detection systems. *IEEE Transactions on Smart Grid*, 2017. In Press.
- [174] Mohamad Houssein Monzer, Kamal Beydoun, and Jean-Marie Flaus. Model based rules generation for intrusion detection system for industrial systems. In *2019 International Conference on Control, Automation and Diagnosis (ICCAD)*, pages 1–6. IEEE, 2019.
- [175] Long Cheng, Ke Tian, and Danfeng (Daphne) Yao. Orpheus: Enforcing Cyber-Physical Execution Semantics to Defend Against Data-Oriented Attacks. In *Proceedings of the 33rd Annual Computer Security Applications Conference (ACSAC'17)*, pages 315–326, 2017.

- [176] F. Adamsky, M. Aubigny, F. Battisti, M. Carli, F. Cimorelli, T. Cruz, A. Di Giorgio, C. Foglietta, A. Galli, A. Giuseppi, F. Liberati, A. Neri, S. Panzieri, F. Pascucci, J. Proenca, P. Pucci, L. Rosa, and R. Soua. Integrated Protection of Industrial Control Systems from Cyber-attacks: the ATENA Approach. *International Journal of Critical Infrastructure Protection*, 2018. In Press.
- [177] Hamid Reza Ghaeini and Nils Ole Tippenhauer. Hamids: Hierarchical monitoring intrusion detection system for industrial control systems. In *Proceedings of the 2Nd ACM Workshop on Cyber-Physical Systems Security and Privacy (CPS-SPC'16)*, pages 103–111, New York, NY, USA, 2016. ACM.
- [178] J. Yang, C. Zhou, S. Yang, H. Xu, and B. Hu. Anomaly Detection Based on Zone Partition for Security Protection of Industrial Cyber-Physical Systems. *IEEE Transactions on Industrial Electronics*, 65(5):4257–4267, May 2018.
- [179] Jinping Liu, Wuxia Zhang, Tianyu Ma, Zhaohui Tang, Yongfang Xie, Weihua Gui, and Jean Paul Niyoyita. Toward security monitoring of industrial cyber-physical systems via hierarchically distributed intrusion detection. *Expert Systems with Applications*, 158:113578, 2020.
- [180] Daesung Moon, Hyungjin Im, Ikkyun Kim, and Jong Hyuk Park. DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks. *The Journal of Supercomputing*, 73(7):2881–2895, Jul 2017.
- [181] Niv Goldenberg and Avishai Wool. Accurate modeling of modbus/tcp for intrusion detection in {SCADA} systems. *International Journal of Critical Infrastructure Protection*, 6(2):63 – 75, 2013.
- [182] Amit Kleinmann and Avishai Wool. Automatic Construction of Statechart-Based Anomaly Detection Models for Multi-Threaded Industrial Control Systems. *ACM Trans. Intell. Syst. Technol.*, 8(4):55:1–55:21, Feb 2017.
- [183] Kagermann Henning. Recommendations for implementing the strategic initiative industrie 4.0, 2013.
- [184] Siemens. SIMATIC OPC UA. <http://www.industry.siemens.com/topics/global/en/tia-portal/software/details/pages/opc-ua.aspx>, 2018. [Online; Accessed May 2018].
- [185] A. Terai, S. Abe, S. Kojima, Y. Takano, and I. Koshijima. Cyber-attack detection for industrial control system monitoring with support vector machine based on communication profile. In *IEEE European Symposium on Security and Privacy Workshops (EuroS PW'17)*, pages 132–138, April 2017.

- [186] Mario Hildebrandt, Kevin Lamshöft, Jana Dittmann, Tom Neubert, and Claus Vielhauer. Information hiding in industrial control systems: An opc ua based supply chain attack and its detection. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, pages 115–120, 2020.
- [187] L. Bayou, N. Cuppens-Boulahia, D. Espès, and F. Cuppen. Towards a CDS-based Intrusion Detection Deployment Scheme for Securing Industrial Wireless Sensor Networks. In *2016 11th International Conference on Availability, Reliability and Security (ARES)*, pages 157–166, Aug 2016.
- [188] Fal Sadikin, Ton van Deursen, and Sandeep Kumar. A zigbee intrusion detection system for iot using secure and efficient data collection. *Internet of Things*, 12:100306, 2020.
- [189] Mehmet Bozdal, Mohammad Samie, and Ian Jennions. A survey on can bus protocol: attacks, challenges, and potential solutions. In *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, pages 201–205. IEEE, 2018.
- [190] Ralph E Mackiewicz. Overview of iec 61850 and benefits. In *2006 IEEE Power Engineering Society General Meeting*, pages 8–pp. IEEE, 2006.
- [191] FortiNet. FortiGate Enterprise Firewall. <https://www.fortinet.com/products/next-generation-firewall.html>, 2018. [Online; Accessed May 2018].
- [192] Harman. Harman Shield. <https://services.harman.com/solutions/automotive-cybersecurity>, 2018. [Online; Accessed May 2018].
- [193] Advenica. Security Solutions for Critical Infrastructures. <https://advenica.com/>, 2018. [Online; Accessed May 2018].
- [194] BAE Systems. Data Loss Prevention. <https://www.baesystems.com/en/product/data-loss-prevention>, 2018. [Online; Accessed May 2018].
- [195] Nexor. Nexor Border Gateway. <https://www.nexor.com/nexor-border-gateway/>, 2018. [Online; Accessed May 2018].
- [196] Fox IT. Fox Data Diode. <https://www.fox-it.com/datadiode/>, 2018. [Online; Accessed May 2018].
- [197] Waterfall Security. FLIP. <https://waterfall-security.com/products/flip>, 2018. [Online; Accessed May 2018].
- [198] Nextnine. ICS Shield. <https://nextnine.com/solutions/ics-shield/>, 2018. [Online; Accessed May 2018].

- [199] AlgoSec. AlgoSec Security Policy Management Solution. <https://www.algosec.com/>, 2018. [Online; Accessed May 2018].
- [200] Sigmaflow. NERC CIP Compliance. <http://www.sigmaflow.com/>, 2018. [Online; Accessed May 2018].
- [201] Positive Technologies. MaxPatrol. <https://www.ptsecurity.com/ww-en/products/maxpatrol/>, 2018. [Online; Accessed May 2018].
- [202] Amenaza Technologies LTD. SecurITree. <https://www.amenaza.com>, 2018. [Online; Accessed May 2018].
- [203] CISCO Systems. CISCO: Protecting ICS with Industrial Signatures. <https://tools.cisco.com/security/center/>, 2018. [Online; Accessed May 2018].
- [204] Cyberbit. SCADAShield. <https://www.cyberbit.net/solutions/ics-scada-security-continuity/>, 2018. [Online; Accessed May 2018].
- [205] AlertEnterprise. Sentry CyberSCADA. <http://www.alertenterprise.com/products-EnterpriseSentryCybersecuritySCADA.php>, 2018. [Online; Accessed May 2018].
- [206] WurldTech (GE). OPShield. <https://www.ge.com/digital/cyber-security>, 2018. [Online; Accessed May 2018].
- [207] Attivo Networks. BOTsink. <https://attivonetworks.com/product/attivo-botsink/>, 2018. [Online; Accessed May 2018].
- [208] Control-See. UCME-OPC. <http://www.controlsee.com/u-c-me-opc/>, 2018. [Online; Accessed May 2018].
- [209] SIGA. SIGA Guard. <http://www.sigasec.com>, 2018. [Online; Accessed May 2018].
- [210] Mission Secure. MSi Secure Sentinel Platform. <http://www.missionsecure.com/solutions/>, 2018. [Online; Accessed May 2018].
- [211] CyberX. XSense. <https://cyberx-labs.com/en/xsense/>, 2018. [Online; Accessed May 2018].
- [212] Halo Digital. Halo Vision. <https://www.halo-digital.com/>, 2018. [Online; Accessed May 2018].
- [213] DarkTrace. Enterprise Immune System. <https://www.darktrace.com/technology/#enterprise-immune-system>, 2018. [Online; Accessed May 2018].

- [214] Leidos. Insider Threat Detection Platform - Wisdom ITI. <https://cyber.leidos.com/products/insider-threat-detection>, 2018. [Online; Accessed May 2018].
- [215] CyberArk. Privileged Account Security Solution. <https://www.cyberark.com/products/>, 2018. [Online; Accessed May 2018].
- [216] ICS2. ICS2 On-Guard. <http://ics2.com/product-solution/>, 2018. [Online; Accessed May 2018].
- [217] ThetaRay. ThetaRay Analysis Platform. <https://www.thetaray.com/platform/>, 2018. [Online; Accessed May 2018].
- [218] Juan E. Rubio, Cristina Alcaraz, Rodrigo Roman, and Javier Lopez. Current cyber-defense trends in industrial control systems. *Computers & Security Journal*, 07/2019 2019.
- [219] Gonzalo De La Torre, Paul Rad, and Kim-Kwang Raymond Choo. Implementation of deep packet inspection in smart grids and industrial internet of things: Challenges and opportunities. *Journal of Network and Computer Applications*, 2019.
- [220] Hadeli Hadeli, Ragnar Schierholz, Markus Braendle, and Cristian Tuduce. Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration. In *2009 IEEE Conference on Emerging Technologies & Factory Automation*, pages 1–8. IEEE, 2009.
- [221] Luying Zhou and Huaqun Guo. Anomaly detection methods for iiot networks. In *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pages 214–219. IEEE, 2018.
- [222] Robin Berthier and William H Sanders. Specification-based intrusion detection for advanced metering infrastructures. In *IEEE 17th Pacific Rim International Symposium on Dependable Computing*, pages 184–193, 2011.
- [223] Steven Cheung, Bruno Dutertre, Martin Fong, Ulf Lindqvist, Keith Skinner, and Alfonso Valdes. Using model-based intrusion detection for scada networks. In *Proceedings of the SCADA security scientific symposium*, volume 46, pages 1–12. Citeseer, 2007.
- [224] Aditya Ashok, Manimaran Govindarasu, and Jianhui Wang. Cyber-physical attack-resilient wide-area monitoring, protection, and control for the power grid. *Proceedings of the IEEE*, 105(7):1389–1407, 2017.
- [225] Hui Lin, Adam Slagell, Zbigniew Kalbarczyk, Peter W Sauer, and Ravishankar K Iyer. Semantic security analysis of scada networks to detect malicious control commands in power grids. In *Proceedings of the first ACM workshop on Smart energy grid security*, pages 29–34. ACM, 2013.

- [226] Saurabh Amin, Xavier Litrico, S Shankar Sastry, and Alexandre M Bayen. Cyber security of water scada systems—part ii: Attack detection using enhanced hydrodynamic models. *IEEE Transactions on Control Systems Technology*, 21(5):1679–1693, 2012.
- [227] Antoine Lemay, Joan Calvet, François Menet, and José M Fernandez. Survey of publicly available reports on advanced persistent threat actors. *Computers & Security*, 72:26–59, 2018.
- [228] Andrew Vance. Flow based analysis of advanced persistent threats detecting targeted attacks in cloud computing. In *2014 First International Scientific-Practical Conference Problems of Infocommunications Science and Technology*, pages 173–176. IEEE, 2014.
- [229] Mirco Marchetti, Fabio Pierazzi, Michele Colajanni, and Alessandro Guido. Analysis of high volumes of network traffic for advanced persistent threat detection. *Computer Networks*, 109:127–141, 2016.
- [230] Guillaume Brogi and Valérie Viet Triem Tong. Terminaptor: Highlighting advanced persistent threats through information flow tracking. In *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2016.
- [231] Ibrahim Ghafir, Mohammad Hammoudeh, Vaclav Prenosil, Liangxiu Han, Robert Hegarty, Khaled Rabie, and Francisco J Aparicio-Navarro. Detection of advanced persistent threat using machine-learning correlation analysis. *Future Generation Computer Systems*, 89:349–359, 2018.
- [232] Patrick Rosenberger and Detlef Gerhard. Context-awareness in industrial applications: definition, classification and use case. *Procedia CIRP*, 72:1172–1177, 2018.
- [233] Cristina Alcaraz and Javier Lopez. Wide-area situational awareness for critical infrastructure protection. *Computer*, 46(4):30–37, 2013.
- [234] Nour Moustafa, Erwin Adi, Benjamin Turnbull, and Jiankun Hu. A new threat intelligence scheme for safeguarding industry 4.0 systems. *IEEE Access*, 6:32910–32924, 2018.
- [235] Behrad Bagheri, Shanhu Yang, Hung-An Kao, and Jay Lee. Cyber-physical systems architecture for self-aware machines in industry 4.0 environment. *IFAC-PapersOnLine*, 48(3):1622–1627, 2015.
- [236] Juan E. Rubio, Mark Manulis, Cristina Alcaraz, and Javier Lopez. Enhancing security and dependability of industrial networks with opinion dynamics. In *European Symposium on Research in Computer Security (ESORICS2019)*, volume 11736, pages 263–280, 2019.
- [237] Seokcheol Lee and Taeshik Shon. Open source intelligence base cyber threat inspection framework for critical infrastructures. In *2016 Future Technologies Conference (FTC)*, pages 1030–1033. IEEE, 2016.

- [238] Juan E. Rubio, Cristina Alcaraz, Rodrigo Roman, and Javier Lopez. Analysis of intrusion detection systems in industrial ecosystems. In *14th International Conference on Security and Cryptography (SECRYPT'17)*, 2017.
- [239] HeSec. HeSec Smart Agents. <http://he-sec.com/products/>, June 2017.
- [240] Sen Nie, Xuwen Wang, Haifeng Zhang, Qilang Li, and Binghong Wang. Robustness of controllability for networks based on edge-attack. *PloS one*, 9(2):e89066, 2014.
- [241] Teresa W Haynes, Sandra M Hedetniemi, Stephen T Hedetniemi, and Michael A Henning. Domination in graphs applied to electric power networks. *SIAM Journal on Discrete Mathematics*, 15(4):519–529, 2002.
- [242] Joachim Kneis, Daniel Mölle, Stefan Richter, and Peter Rossmanith. Parameterized power domination complexity. *Information Processing Letters*, 98(4):145–149, 2006.
- [243] Giuliano Andrea Pagani and Marco Aiello. The power grid as a complex network: a survey. *Physica A: Statistical Mechanics and its Applications*, 392(11):2688–2700, 2013.
- [244] Christopher R Palmer and J Greg Steffan. Generating network topologies that obey power laws. In *Globecom'00-IEEE. Global Telecommunications Conference. Conference Record (Cat. No. 00CH37137)*, volume 1, pages 434–438. IEEE, 2000.
- [245] Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’ networks. *nature*, 393(6684):440, 1998.
- [246] Paul D O’Brien and Richard C Nicol. Fipa—towards a standard for software agents. *BT Technology Journal*, 16(3):51–59, 1998.
- [247] S2Grupo. Emas SOM – Monitoring System for Industrial Environments, 2018. <https://s2grupo.es/es/emas-ics/>, last retrieved in April 2018.
- [248] Wei Ren and Randal W Beard. *Distributed consensus in multi-vehicle cooperative control*, volume 27. Springer, 2008.
- [249] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.
- [250] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226. 2019.
- [251] Vincent D Blondel, Julien M Hendrickx, Alex Olshevsky, and John N Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 2996–3000. IEEE, 2005.

- [252] Rainer Hegselmann, Ulrich Krause, et al. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of artificial societies and social simulation*, 5(3), 2002.
- [253] Solomon E Asch and Harold Guetzkow. Effects of group pressure upon the modification and distortion of judgments. *Organizational influence processes*, pages 295–303, 1951.
- [254] John RP French Jr. A formal theory of social power. *Psychological review*, 63(3):181, 1956.
- [255] Robert P Abelson. Mathematical models of the distribution of attitudes under controversy. *Contributions to mathematical psychology*, 1964.
- [256] Morris H DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.
- [257] Hossein Noorazar. Recent advances in opinion propagation dynamics: a 2020 survey. *The European Physical Journal Plus*, 135(6):1–20, 2020.
- [258] Noah E Friedkin and Eugene C Johnsen. Social influence and opinions. *Journal of Mathematical Sociology*, 15(3-4):193–206, 1990.
- [259] Sergey E Parsegov, Anton V Proskurnikov, Roberto Tempo, and Noah E Friedkin. Novel multidimensional models of opinion dynamics in social networks. *IEEE Transactions on Automatic Control*, 62(5):2270–2285, 2016.
- [260] Guillaume Deffuant, David Neau, Frederic Amblard, and Gérard Weisbuch. Mixing beliefs among interacting agents. *Advances in Complex Systems*, 3(01n04):87–98, 2000.
- [261] Serge Galam. Sociophysics: A review of galam models. *International Journal of Modern Physics C*, 19(03):409–440, 2008.
- [262] Katarzyna Sznajd-Weron and Jozef Sznajd. Opinion evolution in closed community. *International Journal of Modern Physics C*, 11(06):1157–1165, 2000.
- [263] Vishal Sood and Sidney Redner. Voter model on heterogeneous graphs. *Physical review letters*, 94(17):178701, 2005.
- [264] Michael T Gastner and Kota Ishida. Voter model on networks partitioned into two cliques of arbitrary sizes. *Journal of Physics A: Mathematical and Theoretical*, 52(50):505701, 2019.
- [265] Jimit R Majmudar, Stephen M Krone, Bert O Baumgaertner, and Rebecca C Tyson. Voter models and external influence. *The Journal of Mathematical Sociology*, 44(1):1–11, 2020.
- [266] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

- [267] Xi Chen, Xiao Zhang, Yong Xie, and Wei Li. Opinion dynamics of social-similarity-based hegselmann–krause model. *Complexity*, 2017, 2017.
- [268] Duc Truong Pham, Stefan S Dimov, and Chi D Nguyen. Selection of k in k-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(1):103–119, 2005.
- [269] Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734, 2000.
- [270] Jeff Bilmes, Amin Vahdat, Windsor Hsu, and Eun-Jin Im. Empirical observations of probabilistic heuristics for the clustering problem. *Technical Report TR-97-018, International Computer Science Institute*, 1997.
- [271] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.
- [272] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584, 2001.
- [273] Satu Elisa Schaeffer. Graph clustering. *Computer science review*, 1(1):27–64, 2007.
- [274] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.
- [275] Jorge M Santos and Mark Embrechts. On the use of the adjusted rand index as a metric for evaluating supervised classification. In *International conference on artificial neural networks*, pages 175–184. Springer, 2009.
- [276] Cristina Alcaraz and Javier Lopez. Safeguarding structural controllability in cyber-physical control systems. In *European Symposium on Research in Computer Security*, pages 471–489. Springer, 2016.
- [277] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
- [278] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [279] Chin Yang Lee. An algorithm for path connections and its applications. *IRE transactions on electronic computers*, (3):346–365, 1961.
- [280] Bryan Ek, Caitlin VerSchneider, and Darren A Narayan. Global efficiency of graphs. *AKCE International Journal of Graphs and Combinatorics*, 12(1):1–13, 2015.
- [281] Hakim Badis and Khaldoun Al Agha. Qolsr, qos routing for ad hoc wireless networks using olsr. *European Transactions on Telecommunications*, 16(5):427–442, 2005.

- [282] Eric Crawley, Raj Nair, Bala Rajagopalan, and Hal Sandick. A framework for qos-based routing in the internet. Technical report, 1998.
- [283] Chunhung Richard Lin and Jain-Shing Liu. Qos routing in ad hoc wireless networks. *IEEE Journal on selected areas in communications*, 17(8):1426–1438, 1999.
- [284] Lei Chen and Wendi B Heinzelman. A survey of routing protocols that support qos in mobile ad hoc networks. *IEEE Network*, 21(6), 2007.
- [285] Francis Joseph Ogwu, Mohammad Talib, Ganiyu A Aderounmu, and Adedayo Adetoye. A framework for quality of service in mobile ad hoc networks. *Int. Arab J. Inf. Technol.*, 4(1):33–40, 2007.
- [286] Shigang Chen and Klara Nahrstedt. An overview of quality of service routing for next-generation high-speed networks: problems and solutions. *IEEE network*, 12(6):64–79, 1998.
- [287] Afreen Begum Sana, Farheen Iqbal, and Arshad Ahmad Khan Mohammad. Quality of service routing for multipath manets. In *Signal Processing And Communication Engineering Systems (SPACES), 2015 International Conference on*, pages 426–431. IEEE, 2015.
- [288] Ying Ge, Thomas Kunz, and Louise Lamont. Quality of service routing in ad-hoc networks using olsr. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pages 9–pp. IEEE, 2003.
- [289] Thomas Clausen and Philippe Jacquet. Optimized link state routing protocol (olsr). Technical report, 2003.
- [290] Hakim Badis and Khaldoun Al Agha. A distributed algorithm for multiple-metric link state qos routing problem. In *Mobile And Wireless Communications Networks: (With CD-ROM)*, pages 141–144. World Scientific, 2003.
- [291] Hakim Badis and Khaldoun Al Agha. Quality of service for the ad hoc optimized link state routing protocol (qolsr). 2005.
- [292] Zheng Wang and Jon Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on selected areas in communications*, 14(7):1228–1234, 1996.
- [293] Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.
- [294] Mahesh K Marina and Samir R Das. On-demand multipath distance vector routing in ad hoc networks. In *Network Protocols, 2001. Ninth International Conference on*, pages 14–23. IEEE, 2001.

- [295] Bernard Fortz and Mikkel Thorup. Optimizing ospf/is-is weights in a changing world. *IEEE journal on selected areas in communications*, 20(4):756–767, 2002.
- [296] Yan W Chen, Lu F Zhang, and Jian P Huang. The watts–strogatz network model developed by including degree distribution: theory and computer simulation. *Journal of Physics A: Mathematical and Theoretical*, 40(29):8237, 2007.
- [297] Albert-László Barabási, Erzsébet Ravasz, and Tamas Vicsek. Deterministic scale-free networks. *Physica A: Statistical Mechanics and its Applications*, 299(3-4):559–564, 2001.
- [298] Luis A Nunes Amaral, Antonio Scala, Marc Barthelemy, and H Eugene Stanley. Classes of small-world networks. *Proceedings of the national academy of sciences*, 97(21):11149–11152, 2000.
- [299] Albert-László Barabási and Eric Bonabeau. Scale-free networks. *Scientific american*, 288(5):60–69, 2003.
- [300] Juan E. Rubio, Rodrigo Roman, Cristina Alcaraz, and Yan Zhang. Tracking advanced persistent threats in critical infrastructures through opinion dynamics. In *European Symposium on Research in Computer Security*, volume 11098, pages 555–574, Barcelona, Spain, 08/2018 2018. Springer, Springer.
- [301] Cristina Alcaraz, Giuseppe Bernieri, Federica Pascucci, Javier Lopez, and Roberto Setola. Covert channels-based stealth attacks in industry 4.0. *IEEE Systems Journal.*, In Press.
- [302] Ramyar Rashed Mohassel, Alan Fung, Farah Mohammadi, and Kaamran Raahemifar. A survey on advanced metering infrastructure. *International Journal of Electrical Power & Energy Systems*, 63:473–484, 2014.
- [303] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.
- [304] Wenye Wang and Zhuo Lu. Cyber security in the smart grid: Survey and challenges. *Computer Networks*, 57(5):1344–1371, 2013.
- [305] NIST Framework. Roadmap for smart grid interoperability standards. *National Institute of Standards and Technology*, 26, 2010.
- [306] C.-T. Lin. Structural controllability. *IEEE Transactions on Automatic Control*, 19(3):201–208, 1974.
- [307] T. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, and M. A. Henning. Domination in graphs applied to electric power networks. *SIAM Journal on Discrete Mathematics*, 15(4):519–529, 2002.

- [308] Réka Albert, István Albert, and Gary L Nakarado. Structural vulnerability of the north american power grid. *Physical review E*, 69(2):025103, 2004.
- [309] P. Ranganathan and E. Nygard Kendall. *A Distributed Linear Programming Model in a Smart Grid*. Power Electronics and Power Systems. Springer, 2017.
- [310] Lucia Martins, Rita Girao-Silva, Luisa Jorge, Alvaro Gomes, Francesco Musumeci, and Jacek Rak. Interdependence between power grids and communication networks: A resilience perspective. In *DRCN 2017-Design of Reliable Communication Networks; 13th International Conference; Proceedings of*, pages 1–9. VDE, 2017.
- [311] Ajoy K Palit and Dobrivoje Popovic. *Computational intelligence in time series forecasting: theory and engineering applications*. Springer Science & Business Media, 2006.
- [312] Vipin Kumar. Algorithms for constraint-satisfaction problems: A survey. *AI Mag.*, 13(1):32–44, April 1992.
- [313] Cristina Alcaraz and Javier Lopez. Safeguarding structural controllability in cyber-physical control systems. In *The 21st European Symposium on Research in Computer Security (ESORICS 2016)*, volume 9879, pages 471–489, Crete, Greece, 2016. Springer, Springer.
- [314] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [315] Rob Hyndman. Forecast: Forecasting functions for time series and linear models in R. <https://cran.r-project.org/web/packages/forecast/index.html>, last retrieved in October 2017.
- [316] W.D. Stevenson. *Elements of power system analysis*. McGraw-Hill series in electrical engineering: Power and energy. McGraw-Hill, 1982.
- [317] Aristeidis Farao, Juan E. Rubio, Cristina Alcaraz, Christoforos Ntantogian, Christos Xenakis, and Javier Lopez. Sealedgrid: A secure interconnection of technologies for smart grid applications. In *14th International Conference on Critical Information Infrastructures Security (CRITIS 2019)*, In Press.
- [318] Younis A Younis, Kashif Kifayat, and Madjid Merabti. An access control model for cloud computing. *Journal of Information Security and Applications*, 19(1):45–60, 2014.
- [319] Aparna Kumari, Sudeep Tanwar, Sudhanshu Tyagi, Neeraj Kumar, Mohammad S Obaidat, and Joel JPC Rodrigues. Fog computing for smart grid systems in the 5g environment: Challenges and solutions. *IEEE Wireless Communications*, 26(3):47–53, 2019.

- [320] Chang Choi, Christian Esposito, Haoxiang Wang, Zhe Liu, and Junho Choi. Intelligent power equipment management based on distributed context-aware inference in smart cities. *IEEE Communications Magazine*, 56(7):212–217, 2018.
- [321] Adil Rasheed, Omer San, and Trond Kvamsdal. Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access*, 8:21980–22012, 2020.
- [322] Calin Boje, Annie Guerriero, Sylvain Kubicki, and Yacine Rezgui. Towards a semantic construction digital twin: Directions for future research. *Automation in Construction*, 114:103179, 2020.
- [323] Cristina Alcaraz, Juan E. Rubio, and Javier Lopez. Blockchain-assisted access for federated smart grid domains: Coupling and features. *Journal of Parallel and Distributed Computing*, 144:124–135, 06/2020 2020.
- [324] Anne Anderson, Anthony Nadalin, B Parducci, D Engovatov, H Lockhart, M Kudo, P Humenn, S Godik, S Anderson, S Crocker, et al. extensible access control markup language (xacml) version 1.0. *OASIS*, 2003.
- [325] Aissam Outchakoucht, ES Hamza, and Jean Philippe Leroy. Dynamic access control policy based on blockchain and machine learning for the internet of things. *Int. J. Adv. Comput. Sci. Appl*, 8(7):417–424, 2017.
- [326] Riaz Ahmed Shaikh, Kamel Adi, and Luigi Logrippo. A data classification method for inconsistency and incompleteness detection in access control policy sets. *International Journal of Information Security*, 16(1):91–113, 2017.
- [327] Lu Zhou, Chunhua Su, Zhen Li, Zhe Liu, and Gerhard P Hancke. Automatic fine-grained access control in scada by machine learning. *Future Generation Computer Systems*, 93:548–559, 2019.
- [328] Sankardas Roy, Charles Ellis, Sajjan Shiva, Dipankar Dasgupta, Vivek Shandilya, and Qishi Wu. A survey of game theory as applied to network security. In *2010 43rd Hawaii International Conference on System Sciences*, pages 1–10. IEEE, 2010.
- [329] Kong-wei Lye and Jeannette M Wing. Game strategies in network security. *International Journal of Information Security*, 4(1-2):71–86, 2005.
- [330] Kien C Nguyen, Tansu Alpcan, and Tamer Basar. Security games with incomplete information. In *2009 IEEE International Conference on Communications*, pages 1–6. IEEE, 2009.
- [331] Animesh Patcha and J-M Park. A game theoretic approach to modeling intrusion detection in mobile ad hoc networks. In *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.*, pages 280–284. IEEE, 2004.

- 
- [332] Marten Van Dijk, Ari Juels, Alina Oprea, and Ronald L Rivest. Flipit: The game of “stealthy takeover”. *Journal of Cryptology*, 26(4):655–713, 2013.
- [333] Tansu Alpcan and Tamer Basar. A game theoretic analysis of intrusion detection in access control systems. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 2, pages 1568–1573. IEEE, 2004.
- [334] Juan E. Rubio, Cristina Alcaraz, and Javier Lopez. Preventing advanced persistent threats in complex control networks. In *European Symposium on Research in Computer Security*, volume 10493, pages 402–418. 22nd European Symposium on Research in Computer Security (ESORICS 2017), 09/2017 2017.
- [335] Marwaan Simaan and Jose B Cruz. On the stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications*, 11(5):533–555, 1973.
- [336] OpenWeatherMap. Malaga weather, 2019. <https://openweathermap.org/>, last retrieved in February 2019.
- [337] Antoine Lemay, José M Fernandez, and Scott Knight. A modbus command and control channel. In *2016 Annual IEEE Systems Conference (SysCon)*, pages 1–6. IEEE, 2016.
- [338] Carlos Leonardo and Daryl Johnson. Modbus covert channel. In *Proceedings of the International Conference on Security and Management (SAM)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2014.