UNIVERSITY OF WESTERN MACEDONIA
Faculty of Engineering
Department of Electrical and Computer Engineering

Security and Privacy in the Internet of Things

PhD Thesis

Panagiotis Radoglou-Grammatikis

Kozani, Greece
January 2023

Security and Privacy in the Internet of Things

The approval of the present Doctor of Philosophy Thesis from the Department of Electrical Engineering, Faculty of Engineering, University of Western Macedonia does not indicate the acceptance of the author's opinions

(L. 5343/1932, article 202, par. 2)

This PhD thesis has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No. 787011 (SPEAR), No. 833955 (SDN-microSENSE), No. 957406 (TERMINET) and No. 101021936 (ELECTRON).

Abstract

In the digital era of smart economies, the Internet of Things (IoT) plays a significant role in Critical Infrastructures (CIs), providing several benefits such as improved productivity, efficient operation management and self-healing. However, this rapid evolution raises severe security and privacy issues, especially in critical environments. In particular, on the one hand, IoT introduces a set of new and heterogeneous technologies that rely on the vulnerable Internet model (Transmission Control Protocol/Internet Protocol (TCP/IP) networks). On the other hand, Industrial IoT (IIoT) environments are characterised by the presence of legacy systems that are prone to a wide range of security weaknesses and vulnerabilities. For instance, industrial communication protocols, such as Modbus/TCP, Distributed Network protocol 2 (DNP3) and IEC 60870-5-104 do not include any authentication and authorisation mechanisms. Therefore, cyberattackers are able to execute unauthorised activities, such as Man In The Middle (MITM) and Denial of Service (DoS) attacks. In addition, the IIoT-based CIs are an attractive target for a growing number of cyberattackers and Advanced Persistent Threats (APTs). Characteristic APT campaigns are Industroyer (also known as Crashoverride), Stuxnet and TRITON. Finally, it is worth mentioning that IIoT is characterised by constrained computing resources that do not allow the deployment of conventional security mechanisms.

Based on the aforementioned remarks, it is evident that the presence of appropriate defensive mechanisms for the IoT paradigm is necessary. Therefore, the main goal of this PhD thesis is to investigate the security and privacy issues of IoT, providing effective security solutions. After the introductory chapter to this PhD thesis (Chapter 1), the security requirements, challenges and threats are examined in Chapter 2, paying special attention to the unique characteristics and constraints of the IoT entities. In particular, a new IoT threat taxonomy is provided, utilising the Common Attack Pattern Enumeration and Classification (CAPEC) system, while also the efficiency of the existing countermeasures is analysed. It is noteworthy that both academia and industry have provided sufficient solutions, such as the encryption and authorisation mechanisms of the IoT communication protocols. However, they cannot be adopted in IIoT ecosystems due to the role of Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) systems. Consequently, the presence of Intrusion Detection and Prevention Systems (IDPS) is necessary.

According to the countermeasure analysis of Chapter 2, next, Chapter 3 focuses on intrusion detection and mitigation mechanisms. In particular, a special emphasis is given to the architectural design and specifications of IDPS. In addition, the categories of IDPS are analysed, taking full advantage of Artificial Intelligence (AI) techniques. Moreover, mitigation and resilience mechanisms like honeypots and Software-Defined Networking (SDN) are discussed. Finally, this chapter analyses the role of Security Information and Event Management (SIEM) systems in IoT, paying special attention to the normalisation and correlation of the various security events.

Next, in Chapter 4, a thorough analysis of the intrusion detection and mitigation mechanisms in the smart electrical grid (or Smart Grid (SG)) follows. In particular, SG is the largest IoT application, including multiple architectural elements, such as the Advanced Metering Infrastructure (AMI), ICS/SCADA systems, substations and synchrophasors. Therefore, for each of the previous SG elements, relevant intrusion detection and mitigation solutions (such as IDPS, honeypots and SIEM) are studied in terms of their architecture and detection performance. Based on this comparative analysis, the strengths and limitations of the existing solutions are further discussed, thus guiding the implementation of the proposed SDN-enabled SIEM in the following chapter.

Subsequently, Chapter 5 presents the proposed SDN-enabled SIEM, which is composed of three IDPS, namely (a) Network Flow-based IDPS (NF-IDPS), (b) Host-based IDPS (H-IDPS) and (c) Visual-based IDPS (V-IDPS). First, NF-IDPS incorporates multiple ML and DL models that can discriminate cyberattacks and anomalies against a wide range of industrial communication protocols. Next, H-IDPS can recognise operational anomalies against four SG environments: (a) hydropower plant, (b) substation, (c) power plant and (d) smart home. Finally, V-IDPS can detect Modbus/TCP cyberattacks, using visual representations and Convolutional Neural Networks (CNNs). It is worth mentioning that the proposed V-IDPS follows a self-learning approach that can re-train the pre-trained CNN during the inference mode. The security events of the previous IDPS are normalised, correlated and mitigated by the Normalisation, Correlation and Mitigation Engine (NCME). For the correlation process, custom security rules are used, while mitigation relies on the SDN controller. In particular, NCME adopts a Reinforcement Learning (RL) agent, which guides the SDN controller about the appropriate mitigation actions. Finally, NCME includes a sophisticated honeypot deployment mechanism, which relies on a honeypot security game between two players: (a) attacker(s) and (b) defender(s). For the previous honeypot security game, the Nash Equilibrium (NE) is identified, while when NE is not available based on the parameters of the game, two alternative approaches are provided: (a) maxmin-based honeypot development and AI-powered honeypot deployment.

Then, after the description of the proposed detection and mitigation solutions, Chapter 6 summarises the respective evaluation results. In particular, first, a concrete evaluation framework is defined, introducing (a) the evaluation environments, (b) the datasets, (c) comparative methods and (d) the evaluation metrics. Next, the evaluation results for each component of the proposed SDN-enabled SIEM are summarised and discussed, while Appendices M-N provide a detailed comparative analysis.

Finally, Chapter 7 concludes this PhD thesis, providing also potential directions for future research work in this field. In particular, after discussing the key results studied and implemented in the context of this PhD programme, next, five research directions for future work are discussed: (a) Intrusion and Anomaly Detection using Federated Learning (FL), (b) Correlation Mechanisms using Association Learning, (c) RL-based Mitigation Strategies, (d) SDN-powered recovery mechanisms using Graph Neural Networks (GNN) and (e) Explainable AI (XAI) Techniques for AI Detection and Mitigation Models.

Keywords: Artificial Intelligence, Cybersecurity, Industrial Control Systems, Internet of Things, Intrusion Detection, Intrusion Prevention, Privacy, Software-Defined Networking

Περίληψη

Στη ψηφιακή εποχή των έξυπνων οικονομιών, το Διαδίκτυο των Πραγμάτων (ΔτΠ) (Internet of Things – IoT) διαδραματίζει σημαντικό ρόλο στον κύκλο ζωής των κρίσιμων υποδομών, παρέχοντας διάφορα πλεονεκτήματα, όπως η βελτιωμένη παραγωγικότητα, υπηρεσίες αυτοθεραπείας και δυνατότητες ακριβέστερου ελέγχου και διαχείρισης. Ωστόσο, η ραγδαία εξέλιξη των επικοινωνιακών συστημάτων εγείρει σοβαρά ζητήματα ασφάλειας, κυρίως σε βιομηχανικά οικοσυστήματα. Ειδικότερα, το ΔτΠ εισάγει ένα σύνολο ετερογενών τεχνολογιών, οι οποίες χρησιμοποιούν το συμβατικό μοντέλο του διαδικτύου (Internet), το όποιο χαρακτηρίζεται από ένα μεγάλο εύρος ευπαθειών. Επίσης, τα βιομηχανικά οικοσυστήματα χαρακτηρίζονται από την παρουσία συμβατικών συστημάτων, τα οποία είναι επιρρεπή σε ένα ευρύ φάσμα αδυναμιών και ευπαθειών ασφάλειας. Για παράδειγμα, τα βιομηχανικά πρωτόκολλα επικοινωνίας δεν περιλαμβάνουν μηχανισμούς πιστοποίησης ταυτότητας και εξουσιοδότησης. Επομένως, πιθανοί επιτιθέμενοι έχουν τη δυνατότητα να εκτελέσουν επιθέσεις άρνησης υπηρεσιών (Denial of Service – DoS) και «Ανθρώπου Στη Μέση» (Man In The Middle - MITM). Επίσης, οι κρίσιμες υποδομές αποτελούν έναν ελκυστικό στόχο για έναν αυξανόμενο αριθμό κυβερνοεπιτιθέμενων, οι οποίοι δύνανται να σχεδιάζουν και να εκτελούν προγραμματισμένα προηγμένες επίμονες απειλές (Advanced Persistent Threats - APT) . Χαρακτηριστικά παραδείγματα APT αποτελούν οι εκστρατείες: Industroyer, Stuxnet και TRITON. Τέλος, αξίζει να σημειωθεί πως το ΔτΠ χαρακτηρίζεται από περιορισμένους υπολογιστικούς πόρους, οι οποίοι δεν επιτρέπουν την ανάπτυξη συμβατικών μηχανισμών ασφάλειας.

Με βάση τις προαναφερθείσες παρατηρήσεις, είναι προφανής η απαραίτητη παρουσία κατάλληλων μηχανισμών ασφάλειας για το ΔτΠ. Συνεπώς, ο κύριος στόχος της παρούσας διδακτορικής διατριβής είναι η διερεύνηση των ζητημάτων ασφάλειας και ιδιωτικότητας στο ΔτΠ, παρέχοντας αποτελεσματικές λύσεις ασφάλειας. Επομένως, μετά το Κεφάλαιο 1, το οποίο αποτελεί την εισαγωγή στην παρούσα διδακτορική διατριβή, παρουσιάζοντας τους στόχους, τη μεθοδολογία, τη συνεισφορά και τη δομή της εργασίας, οι απαιτήσεις, προκλήσεις και οι απειλές ασφάλειας στο οικοσύστημα του ΔτΠ εξετάζονται αναλυτικά στο Κεφάλαιο 2, δίνοντας ιδιαίτερη προσοχή στα ιδιαίτερα χαρακτηριστικά και τους περιορισμούς των οντοτήτων στο οικοσύστημα του ΔτΠ. Συγκεκριμένα, παρέχεται μια νέα ταξινόμηση των απειλών του ΔτΠ, αξιοποιώντας το σύστημα (Common Attack Pattern Enumeration and Classification - CAPEC), ενώ αναλύεται επίσης η αποτελεσματικότητα των υφιστάμενων αντιμέτρων. Αξίζει να σημειωθεί ότι τόσο η ακαδημαϊκή κοινότητα όσο και η βιομηχανία έχουν αναπτύξει αποτελεσματικές λύσεις ασφάλειας, όπως οι μηχανισμοί κρυπτογράφησης και εξουσιοδότησης των πρωτοκόλλων επικοινωνίας του ΔτΠ. Ωστόσο, οι υφιστάμενοι μηχανισμοί ασφάλειας δεν έχουν τη δυνατότητα να υιοθετηθούν πλήρως σε βιομηχανικά περιβάλλοντα λόγω του αναγκαίου ρόλου των Συστημάτων Βιομηχανικού Ελέγχου (Industrial Control Systems - ICS) και των Συστημάτων Εποπτικού Ελέγχου και Συλλογής Δεδομένων (Supervision Control And Data Acquisition - SCADA). Κατά συνέπεια, η παρουσία Συστημάτων Ανίχνευσης και Πρόληψης Εισβολών (ΣΑΠΕ) (Intrusion Detection and Prevention Systems - IDPS) είναι απαραίτητη.

Με βάση την ανάλυση των αμυντικών μηχανισμών στο προηγούμενο κεφάλαιο, στη συνέχεια το Κεφάλαιο 3 επικεντρώνεται στην ανίχνευση και πρόληψη εισβολών. Ακριβέστερα, διερευνάται το αρχιτεκτονικό μοντέλο και οι προδιαγραφές των συστημάτων ΣΑΠΕ, αναλύοντας τις κατηγορίες τους με βάση τις μεθόδους ανίχνευσης και την τοποθέτηση τους. Επίσης, ιδιαίτερη έμφαση δίνεται στις τεχνικές Τεχνητής Νοημοσύνης (Artificial Intelligence – AI), οι οποίες δύνανται να εφαρμοστούν από τα συστήματα ΣΑΠΕ για την αναγνώριση κυβερνοεπιθέσεων και ανωμαλιών. Επιπλέον, αναλύονται μηχανισμοί πρόληψης εισβολών, όπως οι παγίδες εισβολών (honeypots) και οι μηχανισμοί καθοριζόμενοι από λογισμικό (Software-Defined Networking - SDN). Τέλος, στο συγκεκριμένο κεφάλαιο αναλύεται ο ρόλος των Συστημάτων Διαχείρισης Πληροφοριών και Συμβάντων ασφάλειας (ΣΔΠΣΑ) (Security Information and Event Management -SIEM) στα οικοσυστήματα ΔτΠ, εστιάζοντας στις μεθόδους κανονικοποίησης και συσχέτισης των διαφόρων συμβάντων ασφάλειας.

Στη συνέχεια, στο Κεφάλαιο 4, ακολουθεί μια διεξοδική ανάλυση των μηχανισμών ανίχνευσης και πρόληψη εισβολών σε περιβάλλοντα Έξυπνων Δικτύων Ηλεκτροδότησης (ΕΔΗ) (Smart Grid - SG). Συγκεκριμένα, τα συστήματα ΕΔΗ αποτελούν τη μεγαλύτερη εφαρμογή του ΔτΠ, περιλαμβάνοντας πολλαπλά αρχιτεκτονικά συστατικά, όπως η Προηγμένες Υποδομές Μέτρησης (Advanced Metering Infrastructure - AMI), τα συστήματα βιομηχανικού ελέγχου, οι υποσταθμοί και οι συγχρονιστές. Συνεπώς, για καθένα από τα προηγούμενα αρχιτεκτονικά στοιχεία των συστημάτων ΕΔΗ μελετώνται σχετικές λύσεις ανίχνευσης και πρόληψης εισβολών (όπως ΣΑΠΕ και παγίδες εισβολών), εστιάζοντας στην αρχιτεκτονική και την απόδοση τους. Επομένως, με βάση αυτή τη συγκριτική ανάλυση, διερευνώνται περαιτέρω τα πλεονεκτήματα και οι περιορισμοί των υφιστάμενων λύσεων, καθοδηγώντας έτσι την υλοποίηση του προτεινόμενου ΣΔΠΣΑ με δυνατότητες καθοριζόμενου λογισμικού και τεχνητής νοημοσύνης στο επόμενο κεφάλαιο.

Στη συνέχεια, στο Κεφάλαιο 5 παρουσιάζεται το προτεινόμενο ΣΔΠΣΑ, το οποίο αποτελείται από τρία ΣΑΠΕ: (α) ΣΑΠΕ με βάση δικτυακές ροές, (β) ΣΑΠΕ μεμονωμένων συστημάτων και (γ) ΣΑΠΕ με δυνατότητες οπτικοποίησης. Αρχικά, το πρώτο ΣΑΠΕ ενσωματώνει πολλαπλά μοντέλα μηχανικής και βαθιάς μάθησης, τα οποία δύνανται να διακρίνουν κυβερνοεπιθέσεις και ανωμαλίες έναντι ενός ευρέος φάσματος βιομηχανικών πρωτοκόλλων επικοινωνίας. Στη συνέχεια το δεύτερο ΣΑΠΕ μπορεί να αναγνωρίσει λειτουργικές ανωμαλίες έναντι τεσσάρων περιβαλλόντων ΕΔΗ: (α) υδροηλεκτρικό εργοστάσιο, (β) υποσταθμός, (γ) σταθμός παραγωγής ηλεκτρικής ενέργειας και (δ) έξυπνο σπίτι. Τέλος, το ΣΑΠΕ με δυνατότητες οπτικοποίησης δύναται να ανιχνεύσει κυβερνοεπιθέσεις κατά του πρωτοκόλλου επικοινωνίας (Modbus/TCP), χρησιμοποιώντας οπτικές αναπαραστάσεις και συνεπτυγμένα νευρωνικά δίκτυα. Αξίζει να σημειωθεί ότι το προτεινόμενο ΣΑΠΕ με δυνατότητες οπτικοποίησης ακολουθεί μια προσέγγιση αυτοεκμάθησης, η οποία δύναται να επανεκπαιδεύει το προ-εκπαιδευμένο συνεπτυγμένο νευρωνικό δίκτυο κατά τη διάρκεια της λειτουργίας προβλέψεων σε πραγματικό χρόνο. Τα συμβάντα ασφάλειας των προηγουμένων ΣΑΠΕ κανονικοποιούνται, συσχετίζονται και μετριάζονται από τη Μηχανή Κανονικοποίησης, Συσχέτισης και Μετριασμού (ΜΚΣΜ). Συγκεκριμένα, για τη διαδικασία συσχέτισης χρησιμοποιούνται προσαρμοσμένοι κανόνες ασφάλειας, ενώ ο μετριασμός βασίζεται στην τεχνολογία καθοριζόμενη από λογισμικό. Συγκεκριμένα, η ΜΚΣΜ

υιοθετεί έναν πράκτορα ενισχυτικής μάθησης, ο οποίος καθοδηγεί την τεχνολογία καθοριζόμενη από λογισμικό σχετικά με τις κατάλληλες αμυντικές ενέργειες. Τέλος, η ΜΚΣΜ περιλαμβάνει έναν εξελιγμένο μηχανισμό ανάπτυξης παγίδων εισβολών, ο οποίος βασίζεται σε ένα παιχνίδι ασφάλειας μεταξύ δύο παικτών: (α) επιτιθέμενος(-οι) και (β) αμυνόμενος(-οι). Για το προηγούμενο παίγνιο, προσδιορίζεται η ισορροπία (Nash), ενώ όταν η ισορροπία (Nash) δεν είναι διαθέσιμη με βάση τις παραμέτρους του παιγνίου, παρέχονται δύο εναλλακτικές προσεγγίσεις: (α) ανάπτυξη παγίδων εισβολών με βάση ανάλυση μεγιστοποίησης και ελαχιστοποίησης και (β) ανάπτυξη παγίδων εισβολών με βάση τεχνητή νοημοσύνη.

Στη συνέχεια, μετά την περιγραφή των προτεινόμενων λύσεων ανίχνευσης και πρόληψης, το Κεφάλαιο 6 συνοψίζει τα αντίστοιχα αποτελέσματα αξιολόγησης. Ειδικότερα, αρχικά ορίζεται ένα συγκεκριμένο πλαίσιο αξιολόγησης, παρουσιάζοντας (α) τα περιβάλλοντα αξιολόγησης, (β) τα σύνολα δεδομένων, (γ) τις συγκριτικές μεθόδους και (δ) τις μετρικές αξιολόγησης. Στη συνέχεια, συνοψίζονται και διερευνώνται τα αποτελέσματα αξιολόγησης για κάθε συστατικό στοιχείο του προτεινόμενου ΣΔΠΣΑ με δυνατότητα καθοριζόμενη από λογισμικό, ενώ τα παραρτήματα Μ-Ν παρέχουν τη λεπτομερή συγκριτική ανάλυση.

Τέλος, το Κεφάλαιο 7 ολοκληρώνει την παρούσα διδακτορική διατριβή, παρέχοντας επίσης πιθανές κατευθύνσεις για μελλοντικές ερευνητικές εργασίες στο συγκεκριμένο τομέα. Ειδικότερα, αφού συζητηθούν τα βασικά αποτελέσματα που μελετήθηκαν και υλοποιήθηκαν στο πλαίσιο του παρόντος διδακτορικού προγράμματος, στη συνέχεια, αναλύονται πέντε ερευνητικές κατευθύνσεις για μελλοντικές ερευνητικές προσπάθειες: (α) Ανίχνευση Εισβολών και Ανωμαλιών με Χρήση Ομοσπονδιακής Μάθησης, (β) Μηχανισμοί Συσχέτισης με Χρήση Μάθησης Συσχετίσεων, (γ) Στρατηγικές Μετριασμού με Τεχνικές Ενισχυτικής Μάθησης, (δ) Μηχανισμοί Ανάκτησης με χρήση Τεχνολογίας Καθοριζόμενης από Λογισμικό με Χρήση Νευρωνικών Δικτύων Γράφων και (ε) Τεχνικές Εξήγησης και Ανάλυσης Τεχνητής Νοημοσύνης για Μοντέλα Ανίχνευσης και Μετριασμού Τεχνητής Νοημοσύνης.

Λέξεις Κλειδιά: Ανίχνευση Εισβολών, Βιομηχανικά Συστήματα Ελέγχου, Διαδίκτυο των Πραγμάτων, Ιδιωτικότητα, Κυβερνοασφάλεια, Πρόληψη Εισβολών, Τεχνητή Νοημοσύνη, Τεχνολογία Καθοριζόμενη από Λογισμικό

Acknowledgements

I would like to thank my supervisor, Prof. Panagiotis Sarigiannidis, for his motivation, guidance and encouragement throughout my PhD. His strong experience in the field of cybersecurity, providing technical and psychological support, has been valuable in exploring and understanding new intrusion detection and mitigation mechanisms. Many thanks for all the help and support during the last four years.

It is worth mentioning that this research would not have been possible without the funding of the European Union in the context of the Horizon 2020 projects, namely SPEAR, SDN-microSENSE, TERMINET and ELECTRON.

I would also like to thank everyone at the Laboratory of Internet of THings & AppliCAtions (ITHACA) of the University of Western Macedonia, as they have contributed directly or indirectly to my progression, providing an exciting research environment that I have enjoyed being a part of.

A special word of gratitude goes to my family for the many ways in which they have contributed to my education. Last but not least, my best thanks to Evdoxia Manganopoulou for all the understanding, support, and love these years.

Contents

List of Figures

List of Tables

Abbreviations

Acronym	**Explanation**
6LoWPAN	IPv6 over Low -Power Wireless Personal Area Networks
ABOD	Angle-Based Outlier Detection
ACC	Accuracy
ACL	Access Control Lists
ADT	Attack Defence Tree
AES	Advanced Encryption Standard
AES-128-CMAC	AES-128 Cipher-based Message Authentication Code
AES-CBC	AES in the Cypher Block Chaining (AES-CBC)
AES-CCM	AES in the combined Counter with CBC
AES-CTR	AES in Counter
AH	Authentication Header
AI	Artificial intelligence
AIS	Artificial Immune System
AMI	Advanced Metering Infrastructure
AMQP	Advanced Message Queuing Protocol
ANN	Artificial Neural Networks
ANSI	American National Standards Institute
AODV	Ad hoc On-Demand Distance Vector
APCI	Application Protocol Control Information
APDU	Application Protocol Data Unit
API	Application Programming Interface
AppSKey	Application Key
APT	Advanced Persistent Threat
AR	Availability Requirement

ARIES	smArt gRid Intrusion dEtection System
ARP	Address Resolution Protocol
ASDU	Application Service Data Unit
ASH	Authentication Security Header
AUC	Area Under Curve
BAN	Business Area Network
BLE	Bluetooth Energy
bps	bits per second
CA	Certificate Authority
CAD	Channel Aware Detection
CAPEC	Common Attack Pattern Enumeration and Classification
CART	Classification and Regression Tree
CFP	Contention Free Period
CF	Control Field
CHAID	Chi-square Automatic Interaction Detector
CI	Critical Infrastructure
CIA	Confidentiality, Integrity and Availability
CNN	Convolutional Neural Networks
CoA	Common Address of ASDU
COAP	Constrained Application Protocol
CoT	Cause of Transmission
CPN	Coloured Petri Net
CPS	Cyber-Physical System
cps	connections per second
CPU	Central Processing Unit
CR	Confidentiality Requirement
CSRK	Connection Signature Resolving Key
CSV	Comma-Separated Values
CVE	Common Vulnerabilities and Exposure
CVSS	Common Vulnerability Scoring System
DAO	Destination Advertisement Object
DAO-ACK	DAO Acknowledgement
DDoS	Distributed Denial of Service

DDPG	Deep Deterministic Policy Gradient
DER	Distributed Energy Resources
DFA	Deterministic Finite Automaton
DIO	DODAG Information Object
DIS	DODAG Information Solicitation
DL	Deep Learning
DNN	Deep Neural Networks
DNP3	Distributed Network Protocol 3
DNS	Domain Name System
DODAG	Destination Oriented Directed Acyclic Graph
DoE	Department of Energy
DoS	Denial of Service
DSL	Digital Subscriber Line
DSM	Detection State Machine
dst	Destination
DTK	Deception Toolkit
DTLS	Datagram Transport Layer Security
ECDH	Elliptic Curve Diffie Hellman
EHR	Electronic Health Record
ELI5	Explain Like I am Five
EM	Expectation Maximisation
EPL	Event Processing Language
ERPI	Electric Power Research Institute
ESP	Encapsulating Security Payload
FL	Federated Learning
FN	False Negatives
FNR	False Negative Rate
FP	False Positives
FPR	False Positive Rate
FSM	Finite State Machines
FTDI	Future Technology Devices International
GAN	Generative Adversarial Network
GIS	Global Information System

GNN	Graph Neural Network
GOOSE	Generic Object Oriented Substation Event
GPIO	General Purpose Input/Output
GPS	Global Positioning System
GSAL	Generic Security Application
GSSE	Generic Substation State Events
GTS	Guaranteed Time Slot
GUI	Graphical User Interface
H-IDPS	Host-based Intrusion Detection and Prevention System
HAN	Home Area Network
HIDPS	Host-based Intrusion Detection and Prevention System
HIDS	Host Intrusion Detection System
HIH	High-Interaction Honeypots
HMI	Human Machine Interface
HSC	Honeynet Security Console
HTTP	Hyper Text Transfer Protocol
IAN	Industry Area Network
IANA	Internet Assigned Numbers Authority
ICS	Industrial Control System
ICT	Information Communication and Telecommunication
ID3	Iterative Dichotomiser 3
IDE	Intrusion Detection Engine
IDPS	Intrusion Detection and Prevention System
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IG	Information Gain
IIoT	Industrial Internet of Things
IKEv2	Internet Key Exchange version 2
ILP	Integer Linear Programming
IOA	Information Object Address

IoMT	Internet of Medical Things
IoT	Internet of Things
IP	Internet Protocol
IPS	Intrusion Prevention System
IPv4	Internet Protocol version 4
IR	Integrity Requirement
IRK	Identity Resolving Key
ISP	Internet Service Provider
IT	Information technology
ITACA	Internet Traffic and Content Analysis
ITS	Intelligent Transportation Systems
JSON	JavaScript Object Notation
KNN	K-Nearest Neighbors
KPI	Key Performance Indicator
L2L	Lan-to-Lan
LDA	Linear Discriminant Analysis
LEACH	Low-Energy Adaptive Clustering Hierarchy
LFI	Local File Inclusion
LIH	Low-Interaction Honeypot
LIME	Interpretable Model-Agnostic Explanations
LLNs	Low Power and Lossy Network
LN	Logical Node
LOF	Local Outlier Selection
LoRaWAN	LoRa Wide Area Network
LPWAN	Low Power Wide-Area Networks
LRD	Local Reachability Distance
LTK	Long-Term Key
MAB	Multi-Armed Bandit
MAC	Medium Access Control
MCD	Minimum Covariance Determinant
MIC	Message Integrity Code
MIH	Medium-Interaction Honeypots
MITM	Man In the Middle

ML	Industrial Internet of Things
MLP	Multi-Layer Perceptron
MLTK	Master LTK
MMS	Manufacturing Message Specifications
MOA	Massive Online Analysis
MQTT	MQ Telemetry Transport
MSE	Mean Squared Error
MTU	Master Terminal Unit
NAN	Neighbor Area Network
NCME	Normalisation, Correlation and Mitigation Engine
NE	Nash Equilibrium
NF-IDPS	Network Flow-based Intrusion Detection and Prevention System
NFECM	Network Flow Extraction and Clustering Module
NFEM	Network Flow Extraction Module
NIC	Network Interface Controller
NIDPS	Network-based Intrusion Detection and Prevention System
NIDS	Network Intrusion Detection System
NIST	National Institute of Standards and Technology
NM	Notification Module
NS	Network Simulator
NS2	Network Simulator 2
NTCM	Network Traffic Capturing Module
NTMCM	Network Traffic Monitoring and Capturing Module
NTP	Network Time Protocol
NwkSKey	Network Session Key
OKB	Ontology Knowledge Base
ONOS	Open Network Operating System
OOB	Out of Band
ORG	Originator Address
OS	Operating System
OS-ELM	Online Sequence Extreme Learning Machine
OSI	Open Systems Interconnection
OSSIM	Open Source SIEM

OTX	Open Threat Intelligence
OVS	Open Virtual Switch
OWASP	Open Web Application Security Project
OWASP-RR	OWASP Project Risk Rating
PCA	Principal Component Analysis
PCAP	Packet Capture
PDC	Phasor Data Concentrator
PDF	Probability Density Function
PDML	Packet Description Markup Language
PER	Packet Error Rate
PHP	Hypertext Preprocessor
PICOM	Piece of Information for COMmunication
PLC	Programmable Logic Controller
PMU	Phasor Measurement Unit
pps	packets per second
PRDSA	Probe Route based Defense Sinkhole Attack
QDA	Quadratic Discriminant Analysis
QoS	Quality of Service
QUEST	Quick, Unbiased, Efficient, Statistical Tree
R2L	Remote-to-Local
RAM	Random Access Memory
RBF	Radial Basis Function
RD	Reachability Distance
REST	REpresentational State Transfer
RFC	Request for Comments
RFI	Remote File Inclusion
RL	Reinforcement Learning
ROC	Receiver Operating Characteristic
RPL	Routing Protocol for Low Power and Lossy Networks
RSS	Received Signal Strength
RTOS	Real-Time Operating System
RTU	Remote Terminal Unit
SAP	Sequential Attack Pattern

SCADA	Supervisory Control and Data Acquisition
SDN	Software-Defined Networking
SDN-C	Software-Defined Networking Controller
SEB	Security Enabled Bit
SEP	Smart Energy Profile
SFC	Specific Fuel Consumption
SG	Smart Grid
SHAP	SHapley Additive exPlanation
SIEM	Security Information and Event Management
SIG	Special Interest Group
SLTK	Slave LTK
SMTP	Simple Mail Transfer Protocol
SMV	Sampled Measure Value
SNMP	Simple Network Management Protocol
SOC	Security Operation Centre
SOS	Stochastic Outlier Selection
SPAN	Switched Port Analyser
SPI	Serial Peripheral Interface
SQ	Structure Qualifier
SQL	Structured Query Language
src	Source
SSH	Secure Shell
SSL	Secure Sockets Layer
STK	Short-Term Key
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, DoS, and Elevation of Privilege
SVM	Support Vector Machine
TC	Trust Centre
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TK	Temporal Key
TLS	Transport Layer Security
TN	True Negatives
TNR	True Negative Rate

TP	True Positives
TPR	True Positive Rate
TS	Thompson Sampling
ttl	time to leave
UART	Universal Asynchronous Receiver Transmitter
UCB	Upper Confident Bound
UDP	User Datagram Protocol
UNIX	Uniplexed Information and Computing System
URL	Uniform Resource Locator
US	United States
USB	Universal Serial Bus
UWB	Ultra-Wideband
V-IDPS	Visual-based Intrusion Detection and Prevention System
VRGM	Visual Representation Generation Module
WAN	Wide Area Network
WH	Wireless Honeypot
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WPANs	Wireless Personal Area Networks
WSN	Wireless Sensor Network
XAI	Explainable AI
XSS	Cross-Site Scripting
ZPAN	ZigBee Personal Area Network

Chapter 1

Introduction

This chapter provides an introduction to this PhD thesis, describing (a) the motivation, (b) the objectives, (c) the methodology and (d) the contributions and finally, (e) the structure of this thesis. Therefore, first, the motivation behind this PhD program is provided, considering the wide range of threats and vulnerabilities that characterise the overall Internet of Things (IoT) ecosystem. Next, the objectives of this thesis are enumerated and described. In particular, four main objectives are determined, guiding the definition and implementation of the relevant security solutions developed in this thesis. Next, the methodological framework is presented. Then, based on the implementation and validation activities, the contributions of this thesis are briefly described. Moreover, it is worth highlighting that a list of publications ((a) published papers in international scientific peer-reviewed journals, (b) published papers in international scientific peer-reviewed conference proceedings, (c) book chapters and (d) datasets) that support the achievements of this thesis is provided. Finally, the structure of this thesis is presented.

1.1 Motivation and Objectives

The technological leap of the IoT leads the Critical Infrastructures (CIs) and, in general, the operational environments into a new digital era with multiple benefits, such as self-monitoring, self-healing and pervasive control. However, this evolution raises severe cybersecurity and privacy issues due to the heterogeneous nature of smart and legacy entities. In particular, the operation of legacy systems relies on insecure communication protocols. On the other hand, the smart technologies can result in new security threats and vulnerabilities. In addition, it is worth mentioning that the vast amount of data generated by smart devices, such as sensors and actuators, is an attractive target for potential cyberattackers, while making harder the security and information management of various entities. A cybersecurity incident against a sensitive IoT environment can result in disastrous consequences. A characteristic example was the Advanced Persistent Threat (APT) against a Ukrainian substation, leading to a power outage for more than $225,000$ people [180]. Other relevant APTs were Stuxnet [23], Duqu [23], Flame [23], Gaus [23], DragonFly [23], WannaCry [11] and TRITON [98].

Consequently, it is evident that the presence of reliable detection and mitigation mechanisms is necessary. First, despite the efficiency of the conventional methods, it is evident that they cannot predict and discriminate zero-day cyberattacks. On the other hand, although Artificial Intelligence (AI) and particularly Machine Learning (ML) and Deep Learning (DL) solutions have already proved their capability to detect unknown cyberattacks and anomalies, the operational characteristics and constraints of the IoT environments make the use of ML and DL models challenging [139]. In particular, the IoT ecosystem includes multiple sensitive infrastructures and entities that cannot allow the generation and distribution of the necessary datasets for the ML and DL solutions. Furthermore, these datasets are rarely available, especially in the case of Industrial IoT (IIoT) environments. This fact complicates the cybersecurity analysts to construct appropriate intrusion detection datasets and train their models. Moreover, the heterogeneous nature of the IoT ecosystem makes the implementation of such models harder since each IoT environment is characterised by different attributes. Therefore, based on the aforementioned remarks, the following objectives of this thesis are defined.

Objective #1: Threat Identification in the Internet of Things

The first objective of this thesis refers to identifying and studying the security threats in the IoT. First, the security requirement and challenges are identified, taking into account the unique nature and operational characteristics of the IoT entities. Next, depending on the architectural layers of the IoT (Perception Layer, Communication Layer, Support Layer and Business Layer) and the characteristics of the IoT applications, the corresponding security threats are listed and analysed. In particular, after a study of the IoT services and applications, special emphasis is given to malicious activities against the smart electrical grid (Smart Grid (SG)), which is the largest application in the IoT ecosystem [183]. In addition, complex cyberattacks are investigated in terms of identifying the multiple attack steps. For this purpose, threat and risk analysis techniques (such as Attack Defence Trees (ADT), Common Vulnerability Scoring System (CVSS) and Open Web Application Security Project (OWASP) risk rating methodology), security rules and specifications are also investigated in order to estimate the severity of actual cyberattacks supported by available and custom penetration testing tools (e.g., Smod, Metasploit, Nmap and THC Hydra).

Objective #2: Countermeasure Analysis in the Internet of Things

After the identification and analysis of the security threats in the IoT, the second objective of this thesis focuses on identifying and studying relevant security solutions and countermeasures. A similar approach is utilised by identifying the appropriate countermeasures for each architectural layer. In particular, for each layer, the strong and weak points of each countermeasure are described. A special emphasis is given to the security mechanisms of the IoT communication protocols, such as the encryption methods of Institute of Electrical and Electronics Engineers (IEEE) 802.15.4, IPv6 over Low-Power

Wireless Personal Area Networks (6LoWPAN), Routing Protocol for Low-Power and Lossy Networks (RPL) and Datagram Transport Layer Security (DTLS). In addition, the critical role of intrusion detection and prevention mechanisms in the IoT is highlighted. More specifically, a comprehensive analysis of the current Intrusion Detection and Prevention Systems (IDPS) for the IoT paradigm is conducted by identifying their requirements and weaknesses. Finally, the role of novel technologies, such as Software Defined Networking (SDN), AI, honeypots and Security Information and Event Management (SIEM) systems, is provided, highlighting how they can be used to secure and protect the IoT applications.

Objective #3: Development of AI-powered Intrusion Detection Mechanisms

According to the countermeasure and threat analysis, the third objective of this thesis refers to the efficient and timely detection of IoT-related cyberattacks and anomalies. For this purpose, novel AI mechanisms are adopted, taking full advantage of ML and DL methods. Multiple ML and DL-based intrusion and anomaly detection models are implemented, focusing on the largest IIoT application, i.e., the SG [183]. In particular, three types of intrusion detection mechanisms are implemented: (a) network flow-based intrusion detection, (b) host-based intrusion detection and (c) visual-based intrusion detection. Each of the previous types includes multiple intrusion and anomaly detection models depending on the industrial communication protocols and the operational data of each IIoT/SG environment. Furthermore, it is worth mentioning that the security events generated by the previous mechanisms are normalised into a specific format, while also security rules are defined in order to correlate the security events with each other, thus synthesising security alerts. Based on the aforementioned remarks, a SIEM system is implemented, focusing on IIoT/SG environments.

Objective #4: Implementation of Sophisticated Mitigation and Prevention Mechanisms

After the detection mechanisms, the mitigation and prevention of the corresponding cyberattacks and anomalies follow. For this purpose, novel technologies, such as SDN and honeypots, are investigated and utilised, taking into consideration the unique characteristics and constraints of the IIoT/SG environments. In particular, SDN is used to stop the cyberattacks in a timely and reliable manner, while honeypots are used to increase the resilience of the underlying IIoT/SG infrastructure. In both cases, sophisticated methods are studied and provided in order to optimise the SDN-based mitigation and the deployment of honeypots.

1.2 Methodology

Fig. 1.1 illustrates the methodological framework used in this PhD thesis. In particular, five main steps are followed in order to implement the proposed security solutions and validate their efficiency. Initially, the IoT security and privacy requirements are defined, taking into consideration the essential security principles and the special characteristics and constraints of the IoT entities. Next, the relevant security threats are analysed, utilising a layered approach based on the architecture stack of IoT. In addition, the corresponding security solutions are investigated, thus identifying their strong and weak points. Next, based on the previous countermeasure analysis, it is evident that the presence of reliable IDPS is necessary in a timely manner. Therefore, a detailed literature review takes place, focusing on IDPS systems that monitor and protect IIoT/SG environments. Next, based on this literature review, the overall architecture of the proposed detection and mitigation solutions (provided by this thesis) is designed. Then, their implementation takes place, taking full advantage of novel technologies, such as AI, SDN and honeypots. Finally, the evaluation of the proposed solutions follows, identifying the respective evaluation metrics, comparative methods and simulation experiments.

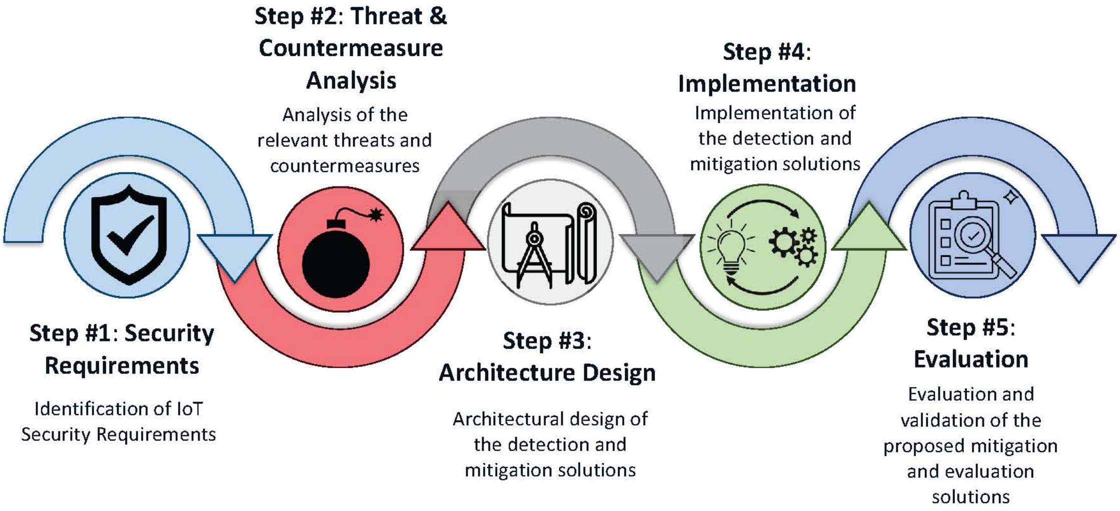

Step #1: Security Requirements
Identification of IoT Security Requirements
Step #2: Threat & Countermeasure Analysis
Analysis of the relevant threats and countermeasures
Step #3: Architecture Design
Architectural design of the detection and mitigation solutions
Step #4: Implementation
Implementation of the detection and mitigation solutions
Step #5: Evaluation
Evaluation and validation of the proposed mitigation and evaluation solutions

FIGURE 1.1: Methodology

1.3 Contributions

According to the previous objectives, the contributions of this thesis are defined as follows.

- **Contribution #1 - New IoT Threat Taxonomy**: A new threat taxonomy was specified for IoT ecosystems. Based on Common Attack Pattern Enumeration and Classification (CAPEC), this taxonomy follows a layered approach, identifying the relevant security threats and CAPEC codes. Additionally, the APT campaigns against the energy sector are identified, utilising the MITRE ATT&CK.
- **Contribution #2 - Comprehensive Review of Intrusion Detection and Prevention Systems**: A comprehensive review of IDPS was conducted, focusing on the energy domain.
- **Contribution #3 - SDN-enabled SIEM Implementation**: The third contribution refers to the implementation of an SDN-enabled SIEM system capable of detecting, normalising and correlating multiple security events that are related to IIoT/SG environments. In addition, the proposed SIEM system can mitigate security alerts, taking full advantage of SDN and AI.

- **Contribution #4 - Implementation of custom ML/DL-based Network flow-based Intrusion Detection Models**: The fourth contribution refers to the implementation of custom ML and DL models for detecting cyberattacks and anomalies against IIoT communication protocols.
- **Contribution #5 - Implementation of custom ML/DL-based Host Intrusion Detection Models for IIoT/SG Environments**: A Host-based Intrusion Detection and Prevention System (H-IDPS) was implemented, recognising anomalies based on operational data (related to four IIoT/SG use cases) and custom ML/DL models.
- **Contribution #6 - Implementation of Visual-based Intrusion Detection and Prevention System**: A Visual-based Intrusion Detection and Prevention System (V-IDPS) was developed, focusing on Modbus/Transmission Control Protocol (TCP) cyberattacks. The proposed V-IDPS follows a custom self-learning approach in order to train the pre-trained Convolutional Neural Network (CNN) used for the detection process.
- **Contribution #7 - New Intrusion Detection Datasets**: In this thesis, three new labelled intrusion detection datasets were generated. They are related respectively to three IIoT communication protocols: (a) Modbus/TCP, (b) International Electrotechnical Commission (IEC) 60870-5-104 and (c) Distributed Network Protocol 3 (DNP3). They are/will be publicly available in IEEE Dataport and Zenodo. These datasets can be used to implement other ML/DL models.
- **Contribution #8 - Honeypot Security Game**: A new honeypot security game was implemented, considering how many production honeypots can be deployed in an IIoT/SG environment in order to increase its resilience. In addition, the Nash Equilibrium (NE) of this security game is identified and proved.
- **Contribution #9 - MaxMin-based Honeypot Deployment**: Based on the previous honeypot security game, a max-min solution is provided in order to calculate the appropriate number of honeypots when the NE is not available.
- **Contribution #10 - AI-powered Honeypot Deployment**: Based on the previous honeypot security game, a Reinforcement Learning (RL)-based solution is provided in order to calculate the appropriate number of honeypots when the NE is not available.

Based on the previous contributions, the results of this thesis are also supported by the following publications, honors and awards. It is worth mentioning that the content of the following publications was used appropriately to structure and compose this PhD thesis.

Publications in International Scientific Peer-Reviewed Journals:

[J1] **P. Radoglou-Grammatikis**, P. G. Sarigiannidis, and I. D. Moscholios, “Securing the internet of things: challenges, threats and solutions”, Internet of Things, vol. 5, pp. 41–70, 2019, doi: 10.1016/j.iot.2018.11.003. - **Published**

[J2] **P. Radoglou-Grammatikis** and P. G. Sarigiannidis, "Securing the Smart Grid: A Comprehensive Compilation of Intrusion Detection and Prevention Systems", in IEEE Access, vol. 7, pp. 46595-46620, 2019, doi: 10.1109/ACCESS.2019.2909807. - **Published**

[J3] **P. Radoglou-Grammatikis**, P. Sarigiannidis, T. Lagkas and I. Moscholios, "A Compilation of UAV Applications for Precision Agriculture", Computer Networks, pp. 1-39, 2019, doi: 10.1016/j.comnet.2020.107148. - **Published**

[J4] P. Diamantoulakis, C. Dalamagkas, **P. Radoglou-Grammatikis**, P. Sarigiannidis, and G. Karagiannidis, "Game Theoretic Honeypot Deployment in Smart Grid", Sensors, vol. 20, no. 15, p. 4199, Jul. 2020, doi: 10.3390/s20154199. - **Published**

[J5] **P. Radoglou-Grammatikis**, P. Sarigiannidis, G. Efstathopoulos, and E. Panaousis, "ARIES: A Novel Multivariate Intrusion Detection System for Smart Grid", Sensors, vol. 20, no. 18, p. 5305, Sep. 2020, doi: 10.3390/s20185305. - **Published**

[J6] **P. Radoglou-Grammatikis**, P. Sarigiannidis, E. Iturbe, E. Rios, S. Martinez, A. Sarigiannidis, G. Eftathopoulos, I. Spyridis, A. Sesis, N. Vakakis, D. Tzovaras, E. Kafetzakis, I. Giannoulakis, M. Tzifas, A. Giannakoulias, M. Angelopoulos, and F. Ramos, "SPEAR SIEM: A Security Information and Event Management System for the Smart Grid", Computer Networks, p. 108008, 2021, doi: 10.1016/j.comnet.2021.108008. - **Published**

[J7] I. Siniosoglou, **P. Radoglou-Grammatikis**, G. Efstathopoulos, P. Fouliras and P. Sarigiannidis, "A Unified Deep Learning Anomaly Detection and Classification Approach for Smart Grid Environments", in IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1137-1151, June 2021, doi: 10.1109/TNSM.2021.3078381. - **Published**

[J8] **P. Radoglou-Grammatikis**, K. Rompolos, P. Sarigiannidis, V. Argyriou, T. Lagkas, A. Sarigiannidis, S. Goudos and S. Wan, "Modeling, Detecting, and Mitigating Threats Against Industrial Healthcare Systems: A Combined Software Defined Networking and Reinforcement Learning Approach", in IEEE Transactions on Industrial Informatics, vol. 18, no. 3, pp. 2041-2052, March 2022, doi: 10.1109/TII.2021.3093905. - **Published**

[J9] **P. Radoglou-Grammatikis**, P. Sarigiannidis, C. Dalamagkas, Y. Spyridis, T. Lagkas, G. Efstathopoulos, A. Sesis, I. L. Pavon, R. T. Burgos, R. Diaz, A. Sarigiannidis, D. Papamartzivanos, S. A. Menesidou, G. Ledakis, A. Pasias, T. Kotsiopoulos, A. Drosou, O. Mavropoulos, A. C. Subirachs, P. P. Sola, J. L. Domínguez-García, M. Escalante, M. M. Alberto, B. Caracuel, F. Ramos, V. Gkioulos, S. Katsikas, H. C. Bolstad, D.-E. Archer, N. Paunovic, R. Gallart, T. Rokkas, and A. Arce, "SDN-Based Resilient Smart Grid: The SDN-microSENSE Architecture", Digital, vol. 1, no. 4, pp. 173–187, Sep. 2021, doi: 10.3390/digital1040013. - **Published**

[J10] I. Nwankwo, M. Stauch, **P. Radoglou-Grammatikis**, P. Sarigiannidis, G. Lazaridis, A. Drosou and D. Tzovaras, "Data Protection and Cybersecurity Certification Activities and Schemes in the Energy Sector", Electronics, vol. 11, no. 6, p. 965, Mar. 2022, doi: 10.3390/electronics11060965. - **Published**

[J11] **P. Radoglou-Grammatikis**, P. Sarigiannidis, P. Diamantoulakis, T. Lagkas, T. Saoulidis, E. Fountoukidis and G. Karagiannidis, "Strategic Honeypot Deployment in Ultra-Dense Beyond

5G Networks: A Reinforcement Learning Approach", in IEEE Transactions on Emerging Topics in Computing, 2022, doi: 10.1109/TETC.2022.3184112. - **Published**

[J12] **P. Radoglou-Grammatikis**, P. Sarigiannidis and G. Efstathopoulos, T. Lagkas, A. Sarigiannidis, V.Mladenov, N. Siaxabanis, "Defending Industrial Internet of Things Against Modbus/TCP Threats: A Combined AI-Based Detection and SDN-Based Mitigation Solution", in Internet of Things, 2022. - **Submitted**

Publications in International Scientific Peer-Reviewed Conference Proceedings:

[C1] **P. I. Radoglou-Grammatikis** and P. G. Sarigiannidis, "Flow anomaly based intrusion detection system for android mobile devices", in 20176th International Conference on Modern Circuits and Systems Technologies (MOCAST), May 2017, pp. 1–4, doi: 10.1109/MOCAST.2017.7937625. - **Published**

[C2] **P. I. Radoglou-Grammatikis** and P. G. Sarigiannidis, "An Anomaly-Based Intrusion Detection System for the Smart Grid Based on CART Decision Tree", 2018 Global Information Infrastructure and Networking Symposium (GIIS), 2018, pp. 1-5, doi: 10.1109/GIIS.2018.8635743. - **Published**

[C3] **P. Radoglou-Grammatikis**, P. Sarigiannidis, T. Liatifis, T. Apostolakos and S. Oikonomou, "An Overview of the Firewall Systems in the Smart Grid Paradigm", 2018 Global Information Infrastructure and Networking Symposium (GIIS), 2018, pp. 1-4, doi: 10.1109/GIIS.2018.8635747. - **Published**

[C4] **P. Radoglou-Grammatikis**, P. Sarigiannidis, I. Giannoulakis, E. Kafetzakis and E. Panaousis, "Attacking IEC-60870-5-104 SCADA Systems", 2019 IEEE World Congress on Services (SERVICES), 2019, pp. 41-46, doi: 10.1109/SERVICES.2019.00022. - **Published**

[C5] G. Efstathopoulos, **P. Radoglou-Grammatikis**, P. Sarigiannidis, V. Argyriou, A. Sarigiannidis, K. Stamatakis, M. Angelopoulos and S. Athanasopoulos "Operational Data Based Intrusion Detection System for Smart Grid", 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2019, pp. 1-6, doi: 10.1109/CAMAD.2019.8858503. - **Published**

[C6] **P. Radoglou-Grammatikis**, P. Sarigiannidis, E. Iturbe, E. Rios, A. Sarigiannidis, O. Nikolis, D. Ioannidis, V. Machamint, M. Tzifas, A. Giannakoulias, M. K. Angelopoulos, A. Papadopoulos, F. Ramos, "Secure and Private Smart Grid: The SPEAR Architecture", IEEE Conference on Network Softwarization (NetSoft), Paris, France, 2020, doi: 10.1109/NetSoft48620.2020.9165420. - **Published**

[C7] **P. Radoglou-Grammatikis**, P. Sarigiannidis, G. Efstathopoulos, P.-A.Karypidis, and A. Sarigiannidis, "DIDEROT: An Intrusion Detection and Prevention System for DNP3-based SCADA Systems", in Proceedings of the 15th International Conference on Availability, Reliability and Security, ser. ARES '20.New York, NY, USA: Association for Computing Machinery, 2020, doi: 10.1145/3407023.3409314. - **Published**

[C8] **P. Radoglou-Grammatikis**, P. Sarigiannidis, A. Sarigiannidis, D. Margounakis, A. Tsiakalos and G. Efstathopoulos, "An Anomaly Detection Mechanism for IEC 60870-5-104", 2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST), Bremen, Germany, 2020, pp. 1-4, doi: 10.1109/MOCAST49295.2020.9200285. - **Published**

[C9] **P. Radoglou-Grammatikis**, I. Siniosoglou, T. Liatifis, A. Kourouniadis, K. Rompolos and P. Sarigiannidis, "Implementation and Detection of Modbus Cyberattacks", 2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST), Bremen, Germany, 2020, pp. 1-4, doi: 10.1109/MOCAST49295.2020.9200287. - **Published**

[C10] V. Mladenov, V. Chobanov, P. Sarigiannidis, **P. I. Radoglou-Grammatikis**, A. Hristov and P. Zlatev, "Defense against cyber-attacks on the Hydro Power Plant connected in parallel with Energy System", 2020 12th Electrical Engineering Faculty Conference (BulEF), Varna, Bulgaria, 2020, pp. 1-6, doi: 10.1109/BulEF51036.2020.9326016. - **Published**

[C11] **P. Radoglou-Grammatikis**, P. Sarigiannidis, G. Efstathopoulos, T. Lagkas, G. Fragulis and A. Sarigiannidis, "A Self-Learning Approach for Detecting Intrusions in Healthcare Systems", ICC 2021 – IEEE International Conference on Communications, 2021, pp. 1-6, doi: 10.1109/ICC42927.2021.9500354. - **Published**

[C12] **P. Radoglou-Grammatikis**, A. Liatifis, E. Grigoriou, T. Saoulidis, A. Sarigiannidis, T. Lagkas and P. Sarigiannidis, "TRUSTY: A Solution for Threat Hunting Using Data Analysis in Critical Infrastructures", 2021 IEEE International Conference on Cyber Security and Resilience (CSR), 2021, pp. 485-490, doi: 10.1109/CSR51186.2021.9527936. - **Published**

[C13] **P. Radoglou-Grammatikis**, C. Dalamagkas, T. Lagkas, M. Zafeiropoulou, M. Atanasova, P. Zlatev, A-A. Boulogeorgos, V. Argyriou, E. K. Markakis, I. Moscholios and P. Sarigiannidis, "False Data Injection Attacks against Low Voltage Distribution Systems", 2022 IEEE Conference and Exhibition on Global Telecommunications (GLOBECOM), 2022. - **Presented**

[C14] **P. Radoglou-Grammatikis**, M. Zafeiropoulou, M. Atanasova, P. Zlatev, S. Giannakidou, T. Lagkas, A-A. Boulogeorgos, V. Argyriou, E. K. Markakis, I. Moscholios and P. Sarigiannidis, "False Data Injection Attacks against High Voltage Transmission Systems", 2023 IEEE International Conference on Communications (ICC), 2023. - **Submitted**

Book Chapters:

[B1] **P. Radoglou-Grammatikis** and P. Sarigiannidis, "Chapter 5: Network Threats", "Book: Cyber-Security Threats, Actors, and Dynamic Mitigation", 2021. - **Published**

Other Publications:

[O1] **P. Radoglou-Grammatikis** and Panagiotis Sarigiannidis, "Secure and Private Smart Grid: The SPEAR Project", in The Project Repository Journal (PRj), 2021. - **Published**

[O2] **P. Radoglou-Grammatikis**, T. Lagkas and Panagiotis Sarigiannidis, "Secure and resilient electrical power and energy systems: the SDN-microSENSE project", in The Project Repository Journal (PRj), 2021. - **Published**

[O3] **P. Radoglou-Grammatikis**, T. Lagkas and Panagiotis Sarigiannidis, "Next Generation IoT Reference Solution: The TERMINET Project", in Open Access Government, 2021. - **Published**

Datasets:

[D1] **P. Radoglou-Grammatikis**, T. Lagkas, V. Argyriou and P. Sarigiannidis, September 23, 2022, "IEC 60870-5-104 Intrusion Detection Dataset", IEEE Dataport, doi: 10.21227/fj7s-f281. - **Published**

[D2] **Panagiotis Radoglou-Grammatikis**, Vasiliki Kelli, Thomas Lagkas, Vasileios Argyriou, Panagiotis Sarigiannidis, November 22, 2022, "DNP3 Intrusion Detection Dataset", IEEE Dataport, doi: https://dx.doi.org/10.21227/s7h0-b081. - **Published**

Honors and Awards:

[A1] **Best Paper Award** - G. Efstathopoulos, **P. Radoglou-Grammatikis**, P. Sarigiannidis, V. Argyriou, A. Sarigiannidis, K. Stamatakis, M. Angelopoulos and S. Athanasopoulos "Operational Data Based Intrusion Detection System for Smart Grid", 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2019, pp. 1-6, doi: 10.1109/CAMAD.2019.8858503.

[A2] **Best Paper Award - P. Radoglou-Grammatikis**, A. Liatifis, E. Grigoriou, T. Saoulidis, A. Sarigiannidis, T. Lagkas and P. Sarigiannidis, "TRUSTY: A Solution for Threat Hunting Using Data Analysis in Critical Infrastructures", 2021 IEEE International Conference on Cyber Security and Resilience (CSR), 2021, pp. 485-490, doi: 10.1109/CSR51186.2021.9527936.

[A3] **Editor's Choice Article** - The paper entitled "ARIES: A Novel Multivariate Intrusion Detection System for Smart Grid" was selected by the MDPI Sensors Editors-in-Chief as a work of particular interest, and was deemed to be highly important in its research area.

[A4] **Top 2% of Scientists in the World for 2022 in Stanford University's List** - Ioannidis, John P.A. (2022), "September 2022 data-update for Updated science-wide author databases of standardized citation indicators", Mendeley Data, V4, doi: 10.17632/btchxktzyw.4.

1.4 Thesis Structure

This thesis is as structured as follows.

- **Chapter #1 - Introduction**: The first chapter provides an introduction to this PhD thesis.

- **Chapter #2 - Security and Privacy in the Internet of Things: A New Threat Taxonomy and Countermeasure Analysis**: The second chapter focuses on the new IoT threat taxonomy, discussing the security requirements, challenges, threats and countermeasures.
- **Chapter #3 - Intrusion Detection and Prevention**: The third chapter provides a background on intrusion detection and prevention mechanisms.
- **Chapter #4 - Review of Intrusion Detection and Prevention Systems for Smart Grid**: Chapter 4 provides a comprehensive literature review of the IDPS systems in the energy domain (i.e., SG).
- **Chapter #5 - Detection and Mitigation of Cyberattacks and Anomalies against Smart Grid**: Next, the fifth chapter focuses on the proposed SDN-enabled SIEM system, describing in detail its detection and mitigation mechanisms.
- **Chapter #6 - Evaluation Analysis**: Chapter 6 focuses on the evaluation results related to the detection and mitigation mechanisms of the proposed SDN-enabled SIEM.
- **Chapter #7 - Conclusions & Future Work**: Finally, Chapter 7 concludes this report.

Chapter 2

Security and Privacy in the Internet of Things: A New Threat Taxonomy and Countermeasure Analysis

The IoT reflects an optimistic digital era, where the objects take full advantage of the typical Internet model and create intelligent synergies with each other, anywhere and anytime. In particular, the IoT incorporates a wide range of technologies, such as sensors, actuators, cloud/edge computing and numerous communication infrastructures and protocols. While the IoT was born in 1999 by K. Ashton [145], the idea of this technology was envisioned many years ago. N. Tesla, in the Colliers magazine in 1926, stated that: "When wireless is perfectly applied the whole earth will be converted into a huge brain, which in fact it is, all things being particles of a real and rhythmic whole and the instruments through which we shall be able to do this will be amazingly simple compared with our present telephone". Similarly, in 1950, the British scientist A. Turing said: "It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. This process could follow the normal teaching of a child" [145]. Today, IoT is adopted by several and critical technological areas such as energy, health and transportations. It is expected that over than 28 billion objects will be able to connect to the Internet by 2025. Many standardisation bodies both from academia and industry have defined the IoT term. For instance, according to the International Telecommunication Union (ITU-T Y.4000/Y.2060 (06/2012)): "A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving, interoperable information and communication technologies" [145].

However, as in any communication network, the IoT is prone to various kinds of security threats and vulnerabilities originating from the conventional Internet model and other technologies such as Wireless Sensor Networks (WSN), 5G and legacy industrial systems. Moreover, it is noteworthy that the objects can interact with each other and their environment in an automatic and autonomous manner, thus increasing the security and privacy concerns. In addition, the multiple IoT communications generate a vast amount of valuable and sensitive data, representing an attractive goal for a growing number of cyberattackers. This chapter provides a comprehensive overview of the security and privacy issues in the IoT, paying special attention to requirements, challenges, threats and countermeasures.

2.1 Entities in the Internet of Things

The IoT can include multiple communication networks where the various hardware or virtualised entities can interact with each other. In particular, in the IoT paradigm, the entities are commonly named ”things”, ”objects”, ”entities”, or ”nodes”. As illustrated in Fig. 2.1, they are characterised by six attributes: (a) Identification, (b) Sensing, (c) Communication, (d) Computation, (e) Services and (f) Semantics.

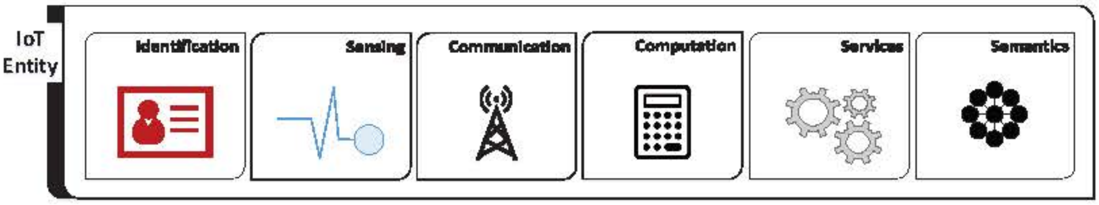


FIGURE 2.1: Attributes of the IoT entities [145]

Each of the above attributes is described briefly below, while in [5], A. Al-Fuqaha et al. provide more details.

- **Identification**: Each IoT entity holds an identifier, such as an Internet Protocol version 4 (IPv4) or version 6 (IPv6) address used for its communications.
- **Sensing**: Sensing denotes that an IoT entity can retrieve useful data from the physical environment through sensors or actuators.
- **Communication**: Communication refers to the interconnection means utilised for the interaction with other IoT entities or users.
- **Computation**: Computation refers to the computing resources for processing the information originating from the (a) physical environment, (b) the user and (c) other IoT entities.
- **Services**: Services denote core functions provided by the IoT entities to the users based on their computations.
- **Semantics**: Semantics implies that IoT entities are capable of receiving the appropriate information and providing the required services.

Characteristic examples of IoT entities are Rasberry Pi [177], Arduino [177], BeagleBone [33], Omega [124], Particle Photon [177], Tessel [124], CubieBoard [124], PLCs [3] and Remote Terminal Units (RTUs) [207]. The first seven devices refer to generic IoT-related boards, while PLCs and RTUs refer to industrial logic controllers utilised in IIoT environments. The aforementioned devices usually hold a microcontroller, memory and several analog and digital General Purpose Input/Output (GPIO) pins. Moreover, they use sensors and actuators, such as temperature sensors, proximity sensors, accelerometers, potentiometers, vibration sensors and moisture sensors. Finally, the functionality of the IoT entities relies on a Real-Time Operating System (RTOS) responsible for managing the communication, computation and storage services. Indicative IoT-related operating systems are Contiki [57, 75], TinyOS [57, 75], FreeRTOS [57], Mbed OS [57], Brillo [57], Windows CE [57] and RIOT [57, 75]. Since this thesis focuses on IIoT environments, a special emphasis will be given to RTU and Programmable Logic Controller (PLC) devices composing Supervisory Control and Data Acquisition (SCADA) systems.

2.2 Communication Architecture in the Internet of Things

As in the case of any Information and Communication Technology (ICT) network, the IoT can be separated into communication layers. Although an official communication stack has not been specified yet by a standardisation body [72], the research community has introduced several suggestions, including three, four or five layers. As depicted in Fig. 2.2, this thesis adopts a communication architecture composed of four layers, namely: (a) Perception Layer, (b) Communication Layer, (c) Support Layer and (d) Business Layer. The first layer focuses on the IoT entities and their sensing capabilities. Next, the Communication Layer is devoted to the data transmission, including six sublayers: (a) Physical Sublayer, (b) Data Link Sublayer, (c) Network Layer, (d) Transport Layer, (e) Session Layer and (f) Application Layer. Each of the aforementioned sublayers adopts respective communication protocols. In the context of this thesis, particular importance will be attached to industrial protocols, such as Modbus/TCP [146], DNP3 [138], IEC 60870-5-104 [141], IEC 61850 (Manufacturing Message Specification (MMS)) [71] and IEC 61850 (Generic Object Oriented Substation Event (GOOSE)) [71]. Next, the Support Layer refers to cloud and edge computing resources, thus facilitating and enhancing the operation of the other layers. Finally, the Business Layer denotes the business applications implemented according to the end-user needs and requirements.

2.3 Services and Applications in the Internet of Things

Based on the attributes of the IoT entities, the relevant services can be classified into four main categories: (a) identity-related services, (b) information aggregation services, (c) collaborative-aware services and (d) ubiquitous services [5]. The first category is the most significant since any physical or

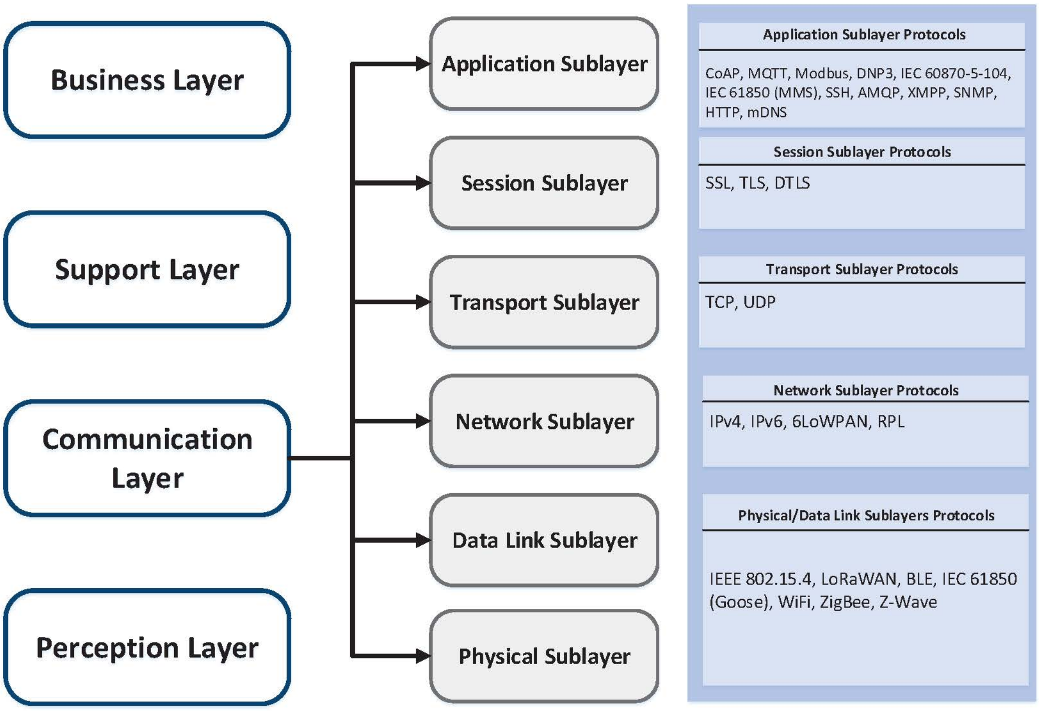

Business Layer
Support Layer
Communication Layer
Perception Layer
Application Sublayer
Session Sublayer
Transport Sublayer
Network Sublayer
Data Link Sublayer
Physical Sublayer
Application Sublayer Protocols
CoAP, MQTT, Modbus, DNP3, IEC 60870-5-104, IEC 61850 (MMS), SSH, AMQP, XMPP, SNMP, HTTP, mDNS
Session Sublayer Protocols
SSL, TLS, DTLS
Transport Sublayer Protocols
TCP, UDP
Network Sublayer Protocols
IPv4, IPv6, 6LoWPAN, RPL
Physical/Data Link Sublayers Protocols
IEEE 802.15.4, LoRaWAN, BLE, IEC 61850 (Goose), WiFi, ZigBee, Z-Wave

FIGURE 2.2: IoT Communication Architecture and Protocols [145]

virtual object should be identified first in order to participate in the digital reality of IoT. Next, the information aggregation services refer to retrieving and aggregating operational data from the physical environment. The operational data is used by the collaborative-aware services for decision making and responding appropriately. Finally, the ubiquitous services denote that the collaborative-aware service can be provided anytime to anyone and anywhere. Although the main goal of the IoT applications is to provide ubiquitous services, most of the current IoT applications can support identity-related, information aggregation and collaborative-aware services. The smart electrical grid and healthcare environments are more related to the information aggregation service, while smart home and Intelligent Transportation Systems (ITS) refer mainly to the collaborative-aware services. Subsequently, a short description is provided about smart home, ITS, smart healthcare and smart cities, while special emphasis is given to the smart electrical grid, which constitutes the biggest IoT application [183].

In the context of a smart home [88], the IoT services intend to optimise the personal quality of life, offering a convenient and easy manner to monitor and adjust automatically and remotely home appliances, such as heating systems, smart meters and air conditioners. For instance, a smart home can close the windows and adjust the temperature based on the weather conditions. The IoT entities within a smart home are needed to interact with their internal and external environment. The internal environment refers to the home appliances, while the external environment includes entities that are not under the control of the smart home, such as the electrical grid. On the other side, ITS [133, 163] refers to intelligent communication and computation services that monitor and control the transportation network. ITS intends to increase the reliability, availability and safety of the transportation ecosystem. An ITS consists of four main subsystems, namely: (a) vehicle subsystem, (b) station subsystem, (c) monitoring centre and (d) security subsystem. Audi, Google and Volvo have provided remarkable outcomes in this area [183]. Given the COVID-19 pandemic, smart healthcare services, such as personalised healthcare, remote monitoring, health education, preventive care and faster diagnosis, are necessary than ever. The evolution of the Internet of Medical Things (IoMT) [214] leads healthcare organisations to adopt medical telemetry solutions, such as wearables, medical implantables and intelligent Electronic Health Records (EHRs). Finally, a smart city [221] takes full advantage of the previous applications and implies a set of collaboration and ubiquitous services interconnected to each other, improving the quality of life in the city.

2.4 Smart Electrical Grid: The Biggest IoT Application

The smart electrical grid or differently SG has been determined by several organisations, such as the United States (US) Department of Energy (DoE), the European Commission Task Force for Smart Grid and the Electric Power Research Institute (ERPI). SG is considered the interconnection between the existing electrical grid and ICT, allowing the two-way communication between the energy consumers and utilities. The main goal of SG is to ensure the appropriate sustainability with respect to energy generation, transmission and distribution through distributed generation, storage and smart measurement. Moreover, it is worth mentioning the ability of SG to form and create microgrids or islands, taking full advantage of Distributed Energy Resources (DERs) relying on renewable, such as solar and wind energy. The main architectural components of the SG are SCADA systems, Advanced Metering Infrastructure (AMI), substations and synchrophasors [148]. Each of them is described further below.

The goal of a SCADA system is to monitor and control the automatic function of other components. In particular, a SCADA system is composed of five main ingredients, namely: (a) measuring instruments, (b) logic controllers, (c) Master Terminal Unit (MTU), (d) communication network and (e) Human Machine Interface (HMI) [148]. First, the measuring instruments refer to sensors and actuators capable of monitoring and retrieving operational measurements, such as pressure, temperature and voltage. Next, based on the data of the measuring instrument, the logical controllers recognise potential abnormalities with respect to the system behaviour, thus activating or deactivating appropriate controlling mechanisms. Characteristic examples of logical controllers are PLCs and RTUs. The logical controller communicates with MTU, which refers to a centralised server through which the system operator can send various commands to the logical controllers. The communication network refers to the communication between the logical controllers and MTU, utilising industrial protocols, such as Modbus, Profinet and BACnet. Finally, HMI is a Graphical User Interface (GUI) used by the system operator in order to contact the logical controllers.

The AMI provides the necessary services behind the two-way communication between the energy consumers and utilities. The AMI is composed of three primary components: (a) smart meters, (b) Data

Collectors and (c) the AMI headend [147]. First, the smart meters are responsible for monitoring the energy consumption and other electricity measurements. Next, the data collectors store and pre-process the data generated by several smart meters belonging to different geographic areas. Finally, the AMI headend usually refers to the computing resources of the utility company, such as centralised or decentralised servers. The AMI headend receives, processes, stores and manages the information of each data collector, thus providing the ability to the utility company to re-consider or apply different policies. On the other hand, the energy consumers are able to monitor and control in real-time their energy consumption. It is worth mentioning that the components mentioned above can belong to different geographic regions with varying characteristics and constraints. Consequently, each component uses different communication technologies and mediums that will be further analysed subsequently.

The conventional model of the electrical grid relies on three main actions: (a) energy generation, (b) transmission and (c) distribution. The role of a substation in the electrical grid is crucial with respect to the energy transmission and distribution. In particular, both transmission and distribution substations handle the energy produced, configure the transmission or distribution process, respectively and control the power increase [148]. They include several hardware and software components, such as Intelligent Electronic Devices (IEDs), Global Position System (GPS), a Global Information System (GIS), RTUs, PLCs and HMI. Usually, the IEC 61850 protocol [71] is adopted with respect to the IEDs within a substation.

A synchrophasor system refers to an emerging technology critical for the reliability and sustainability of the modern electrical grid. Similarly to the SCADA systems, a synchrophasor system consists of Phasor Measurement Units (PMUs), Phasor Data Concentrators (PDCs), a communication network and HMI [148]. First, a Phasor Measurement Unit (PMU) refers to a device responsible for calculating various measurements from current/voltage waveforms like phase angle, frequency, reactive power and active power. Next, a Phasor Data Concentrator (PDC) plays the role of MTU, collecting and converting the data of multiple PMUs into a single flow. Usually, IEEE C37.118.2 [70] is utilised for the communications between PDC and PMUs. Finally, HMI visualises the outcomes of PDC.

The SG ecosystem consists of different geographic areas utilising different communication means and protocols. Fig. 2.3 illustrates a high-level architectural diagram of SG in terms of the main communication features. The first layer includes three types of are networks: (a) Home Area Network (HAN), (b) Business Area Network (BAN) and (c) Industry Area Network (IAN). The common characteristic of HAN, BAN and IAN is the presence of smart meters monitoring the energy consumption and similar measurements. In particular, HAN denotes a network comprising home-related appliances, such as a washing machine, fridge and microwave. On the other side, BAN refers to a network including devices and technologies used by small and medium organisations and enterprises, such as desktop computers. Finally, IAN refers to networks related to large-scale organisations using multiple servers and heavy computing resources. The various devices in the aforementioned networks usually adopt ZigBee, Z-Wave, IEEE 802.11 (i.e., Wireless Fidelity (Wi-Fi)) and rarely power line communications.

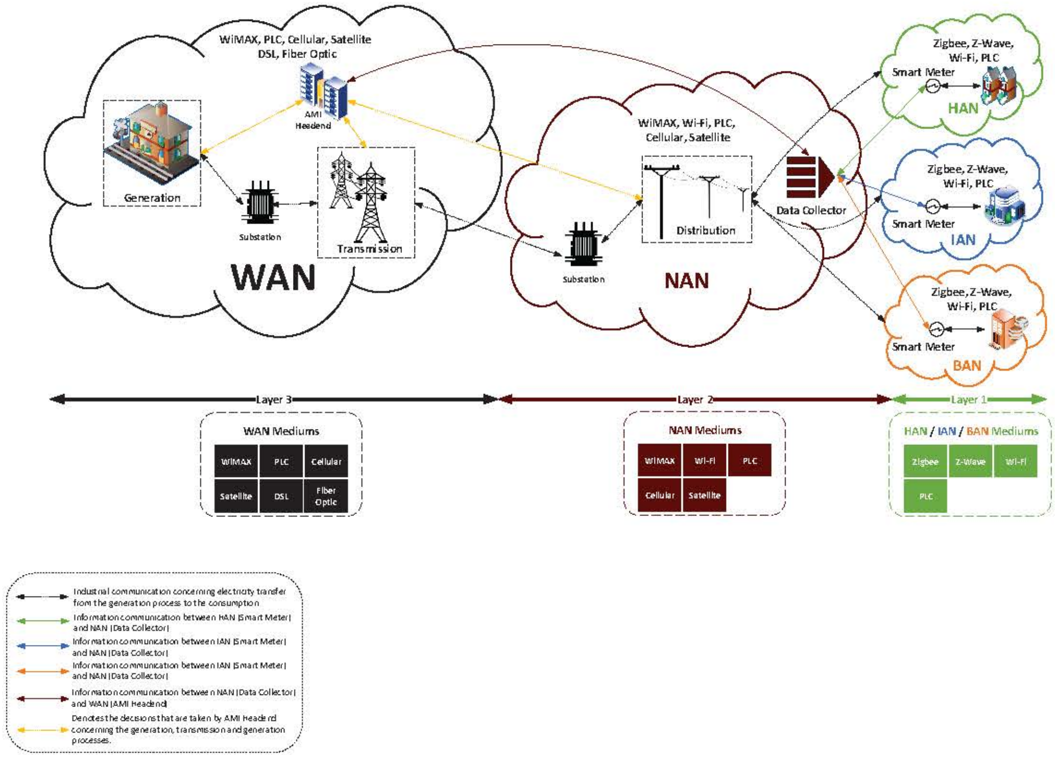

WiMAX, PLC, Cellular, Satellite
DSL, Fiber Optic
AMI Headend
Generation
Substation
Transmission
WAN
WiMAX, Wi-Fi, PLC, Cellular, Satellite
Distribution
Data Collector
Substation
NAN
Zigbee, Z-Wave, Wi-Fi, PLC
Smart Meter
HAN
Zigbee, Z-Wave, Wi-Fi, PLC
Smart Meter
IAN
Zigbee, Z-Wave, Wi-Fi, PLC
Smart Meter
BAN
Layer 3
Layer 2
Layer 1
WAN Mediums
WiMAX
PLC
Cellular
Satellite
DSL
Fiber Optic
NAN Mediums
WiMAX
Wi-Fi
PLC
Cellular
Satellite
HAN / IAN / BAN Mediums
Zigbee
Z-Wave
Wi-Fi
PLC
Industrial communication concerning electricity transfer from the generation process to the consumption
Information communication between HAN (Smart Meter) and NAN (Data Collector)
Information communication between IAN (Smart Meter) and NAN (Data Collector)
Information communication between IAN (Smart Meter) and NAN (Data Collector)
Information communication between NAN (Data Collector) and WAN (AMI Headend)
Denotes the decisions that are taken by AMI Headend concerning the generation, transmission and generation processes.

FIGURE 2.3: The SG architecture in terms of communication mediums [148]

Next, the second and third layers are devoted to several Neighbor Area Networks (NANs) and Wide Area Networks (WANs), respectively. First, a NAN is related to a small geographic area comprising multiple HANs, BANs and IANs, respectively. The main characteristic of a NAN is the presence of one or more data collectors responsible for aggregating the data originating from the smart meters of the first layer. The communication between the first and the second layer usually relies on IEEE 802.16 (i.e., Worldwide Interoperability for Microwave Access (WiMAX)) and IEEE 802.11 (i.e., Wi-Fi). On the other side, a WAN is responsible for connecting the data of the various NANs with AMI headends in order to optimise the processes related to the energy generation, transmission and distribution. The communication mechanisms behind a WAN can include IEEE 802.16, power line communications, cellular, satellite and Digital Subscriber Line (DSL) communications.

Based on the aforementioned remarks, it is evident that a cyberattack against the SG can produce disastrous consequences. A characteristic example was the APT against a Ukrainian substation in December 2016, resulting in a power outage for more than 225000 households. Appendix B and Appendix C summarise APT campaigns and well-known malware targeting energy-related organisations and Industrial Control Systems (ICS), respectively. In Appendix B, for each APT campaign, the corresponding techniques and malware are given, while similarly, in Appendix C, for each malware, the respective techniques are enumerated.

2.5 Security Requirements in the Internet of Things

Before investigating the various security threats related to the IoT and the appropriate countermeasures, first, the security requirements have to be identified. The security requirements intend to specify a set of security principles that should be guaranteed in the context of the IoT applications. Several studies have already defined precisely the security requirements for the IoT. For instance, some of them are listed in [8, 56, 73, 82, 111, 126, 145, 185]. Therefore, based on them, the following security principles are considered.

- **Confidentiality**: This term refers to two interrelated terms. First, Confidentiality implies that unauthorised users, entities and services must not access private information. Second, Confidentiality ensures that privacy and proprietary information are protected.
- **Integrity**: Integrity denotes that the attributes of the IoT entities and their interchangeable information shall not be violated, modified and used by unauthorised users, entities and services.
- **Availability**: Availability denotes that the computing resources, information and services shall be available when needed. This means that the IoT entities, the communication channels and the computation mechanisms should operate properly based on the user and business requirements of each IoT application.
- **Authenticity**: Authenticity refers to the fact that the various information and transactions must be genuine. This security principle means that the parties participating in a transaction must be the ones that they claim to be.
- **Accountability**: Accountability means that each IoT entity must be able to be identified and mapped in a unique way. To this end, non-repudiation, fault isolation, deterrence, intrusion detection and prevention and recovery mechanisms can be supported. Given that absolutely secure and private IoT entities and applications are not yet an attainable goal, the security breaches must be recognised with respect to the affected IoT entities.

2.6 Security Challenges in the Internet of Things

The IoT constitutes an evolutionary paradigm of the typical Internet model, incorporating its security weaknesses and vulnerabilities but also a heterogeneous set of technologies allowing the physical world to meet the digital era. Therefore, the security mechanisms should consider the unique nature and the functionality of the IoT entities and their communications. Several studies have already identified the security and privacy challenges in the IoT. Indicative examples are [56, 73, 75, 145]. According to them, the following challenges are listed and described.

- **Interoperability**: The security mechanisms should not significantly limit and impact the functionality of the IoT entities and applications.
- **Limited Computing and Storage Resources**: The IoT entities are characterised by limited computing, memory and storage resources. Therefore, they cannot fully support heavy security mechanisms like conventional asymmetric encryption.
- **Resilience against Physical Attacks and Natural Disasters**: The entities in the IoT are usually small without including any protection measure with respect to the physical security. For instance, a mobile device or sensor can be stolen, while fixed entities and facilities can be destroyed by natural disasters and environmental threats.
- **Automated and Autonomous Control**: The conventional information systems are configured by the users. However, on the other side, the IoT entities have the ability to configure and adjust their operation by themselves.
- **Big Data**: The IoT entities and applications generate, process and handle a massive amount of sensitive data that is an attractive target for a growing number of cyberattackers.
- **Privacy**: Several IoT entities and applications are related to sensitive data that must not be identifiable, traceable and linkable. A typical example is the IoT entities and applications processing medical and financial data.
- **Scalability**: The IoT applications can include and expand multiple networks, including numerous IoT entities and computing systems. Consequently, the security and privacy mechanisms should also be scalable.

2.7 Security Threats in the Internet of Things

This section introduces a threat taxonomy with respect to the layers mentioned above in subsection 2.2. Consequently, for each layer, the various threats are described and linked to the entries of the CAPEC catalogue established by the US Department of Homeland Security. In particular, Appendix A provides a summary of the various IoT threats and corresponds them to the relevant CAPEC codes. The CAPEC catalogue provides attack patterns explaining how the adversaries can take full advantage of potential weak points in cyber-physical applications in order to violate their security. Moreover, a CAPEC attack pattern provides the challenges of the cyberattacker and includes information with respect to the design and execution of the cyberattacks. Consequently, according to (a) the nature of each layer, (b) the role of the IoT entities and (c) the relevant interfaces, the corresponding threats are discussed. For instance, at the perception layer, the adversaries focus on the physical security, while at the communication layer, the attackers usually aim to exploit the weaknesses of the communication protocols through network and routing attacks. Furthermore, subsection 2.7.5 is dedicated to threats belonging to one or more layers. Finally, it is noteworthy that this threat taxonomy is industry and vendor agnostic.

2.7.1 Security Threats at the Perception Layer

The security measures at the perception layer focus mainly on the physical security of the IoT entities based on two complement requirements. First, the security measures should ensure the availability of the physical infrastructure, preventing and mitigating potential accidents and disasters. Second, integrity and authenticity should be guaranteed in terms of preventing misuse of the physical infrastructure that can lead to abuse or fraudulent use of sensitive information. Based on the aforementioned remarks, the main threats of this layer are discussed below.

Natural Disasters and Environment Threats: Characteristic examples of natural disasters are floods, tornadoes, earthquakes, hurricanes and ice storms. Such phenomena can destroy the physical facilities behind the IoT applications. On the other side, environmental threats like chemical and water accidents or inappropriate values of temperature and humidity can also affect the physical security of the IoT applications. Although the likelihood of such events is not high and effective prevention and detection measures are in place, their impact can be disastrous.

Human-caused Physical Threats: In contrast to the previous case, the human-caused physical threats represent a more challenging category since the malicious activities can bypass potential detection and mitigation mechanisms. Vandalisms, eavesdropping, misuse and device tampering are indicative examples of this category. The human-caused physical threats can affect and violate all the security requirements discussed in section 2.5, thus resulting in devastating consequences. Despite the fact that there are effective detection and prevention mechanisms, their probability of occurrence is high.

2.7.2 Security Threats at the Communication Layer

The attack vectors related to the network services are mainly due to the weaknesses of the respective communication protocols. Many of them were designed without including sufficient cybersecurity measures, such as authentication and authorisation. Characteristic examples are the Address Resolution Protocol (ARP), Domain Name System (DNS), Modbus, DNP3 and various routing protocols. Therefore, the potential cyberattackers have the capability to take advantage of the various weak points and compromise the security requirements of the involved entities. For example, the unauthorised access attacks against many application-layer protocols, such as Modbus, can lead a cybercriminal to cause disastrous consequences against an industrial environment. On the other side, the weaknesses of the ARP protocol can result in Man-In-The-Middle (MITM) attacks, which in turn can cause replay, Denial of Service (DoS) and data modification attacks. This subsection is devoted to the threats at the communication layer. First, various kinds of reconnaissance and DoS attacks are discussed. Next, a special attention is given to cyberattacks against routing protocols like RPL and Ad hoc On-Demand Distance Vector (AODV). Finally, the MITM attacks are discussed.

Reconnaissance Attacks: Through the reconnaissance attacks, the cyberattacker intends to gather valuable information about the victim, such as potential weak points and vulnerabilities. Usually, a reconnaissance attack is the first step before other attacks. A typical example in this category is port scanning, exploring what ports and services are available. Moreover, through port scanning, the cyberattacker can retrieve useful information about the operating system and the version of the services running on the target system. In addition, vulnerability scanning and analysis is another example falling into this category. Through vulnerability scanning and analysis, the cyberattacker can identify potential Common Vulnerabilities and Exposures (CVEs) and exploits.

Denial of Service Attacks: The DoS attacks target the availability of the involved systems and mainly the network services running on them. Based on the National Institute of Standards and Technology (NIST) Computer Security Incident Handling Guide, a DoS attack is defined as an action exhausting the computing resources like the Central Processing Unit (CPU), bandwidth, memory and disk space in order to prevent or impair the authorised use of systems, networks and applications. Based on this definition, three main DoS attacks can be distinguished targeting network bandwidth, system resources and application resources, respectively. Moreover, DoS attacks can be classified based on the number of potential attackers. A single cyberattacker or a small number of them may launch a DoS attack. On the other side, several cyberattackers can collaborate in order to form a Distributed Denial of Service (DDoS) or an amplification attack. The network bandwidth refers to the capacity of the network links that connect a server with the Internet. In most cases, this is the connection between the organisations and their Internet Service Provider (ISP). Typically, this connection has a lower capacity than ISPs. This means that over such higher-capacity connections, more traffic can arrive at the ISP's routers than can be transported over the connection to the organisation. Therefore, the ISP's routers should discard some packets, transmitting only those that can be supported by the communication links. In a normal scenario, this behaviour is usually noticed when popular servers receive a large number of requests, resulting in not supporting a random portion of users. On the other side, in the case of a DoS attack targeting the network bandwidth, the cyberattackers generate a plethora of malicious requests that exceed the normal ones. Thus, the legitimate users cannot access the available services. The goal of the DoS attacks targeting the system resources is to overload or crash the network services by using specific network packets that usually take advantage of the limited resources or the network protocols' weaknesses. More specifically, in contrast to the DoS attacks consuming network bandwidth, this kind of DoS either uses packets that consume limited resources, such as temporary buffers, tables of open connections and similar memory data structures or exploits network protocols' vulnerabilities. SYN spoofing and ping of death attacks are characteristic examples, respectively. The DoS attacks against a software application, such as a web server, usually are conducted by transmitting several malicious, but valid network packets so that the server cannot respond to the legitimate requests. For instance, a web server might provide the ability to access a specific database via appropriate queries. In this case, the attacker aims at generating and transmitting continuously multiple queries that will not allow the server to respond to the legitimate requests. Finally, another DoS attack of this category can target a potential vulnerability of a software application that will result in its termination.

Sybil Attacks: In the Sybil attack, the malicious nodes forge or create multiple identities to deceive other nodes, in order to monitor various parts of the network [135]. According to P. Radoglou-Grammatikis and P. Sarigiannidis in [135], a general model of the Sybil attack is presented in Fig. 2.4, where nodes X, Y, and Z forge the identities of the various nodes. This attack can be divided into three types: SA-1, SA-2 and SA-3 [135]. In general, SA-1 attackers create connections inside a Sybil group, as shown in Fig. 2.5. In this case, the Sybil nodes are closely related to other Sybil nodes. The SA-1 Sybil attacks are usually performed against sensing environments or mobile sensing systems. For instance, a voting system can be significantly impacted since an SA-1 Sybil attack will try to forge a large number of identities, thus affecting the final vote outcome. On the other side, SA-2 and SA-3 Sybil attacks (Fig. 2.6 and Fig. 2.7) are capable of creating connections not only with the malicious nodes but also with the legitimate ones. Both of them attempt to imitate the behaviour of legitimate nodes by transmitting appropriate messages. The difference between SA-2 and SA-3 is that SA-3 focuses on mobile networks, where the connections among the nodes cannot exist for a long time. However, this characteristic of the mobile networks makes it difficult to detect SA-3 attacks since the network topology is changed frequently, and the nodes' behaviour patterns cannot be identified. Hence, based on the aforementioned remarks, Sybil attacks can compromise the confidentiality and authenticity of a network. Their impact is considered important; however, IDPS can detect and mitigate them. In [200], L. Wallgren et al. simulate such attacks, using the Contiki Operating System (OS) and Cooja simulator. On the other side, K. Zhang et al. in [220] study relevant detection methods devoted to SA-1, SA-2 and SA-3. Finally, P. Sarigiannidis et al. in [165] focus on Sybil attacks against WSN, providing a relevant detection method, using Ultra-Wideband (UWB) ranging-based information.

Selective Forwarding Attacks: A selective forwarding attack is a routing threat designed to compromise the availability and integrity of the network by corrupting selectively or not the network packets [135]. Fig. 2.8 illustrates a general model of this attack, where the node Z arbitrarily drops those packets coming from the nodes A and Z. In particular, there are two main types of selective forwarding attacks, namely (a) blackhole and (b) grayhole. In the first category, blackhole constitutes a kind of DoS attack at the routing layer, where the attacker drops all the packets. A notable survey related to blackhole attacks is presented by F. Tseng et al. in [192]. Similarly, L. Wallgren et al. [200]

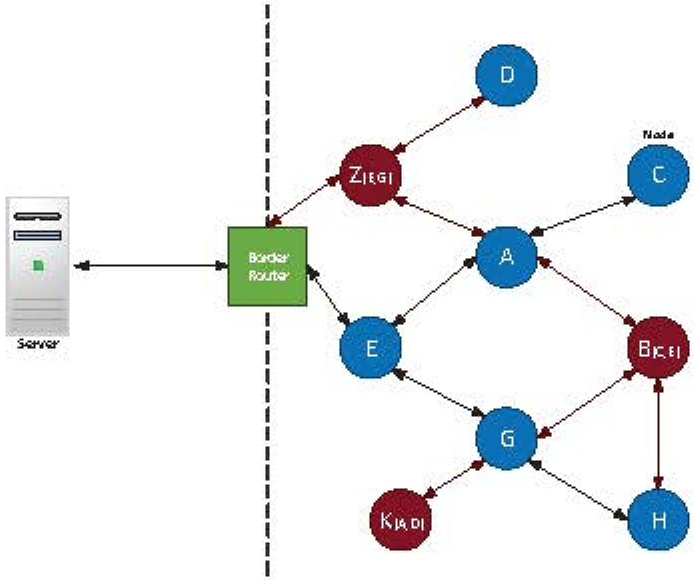


FIGURE 2.4: Typical Sybil Attack

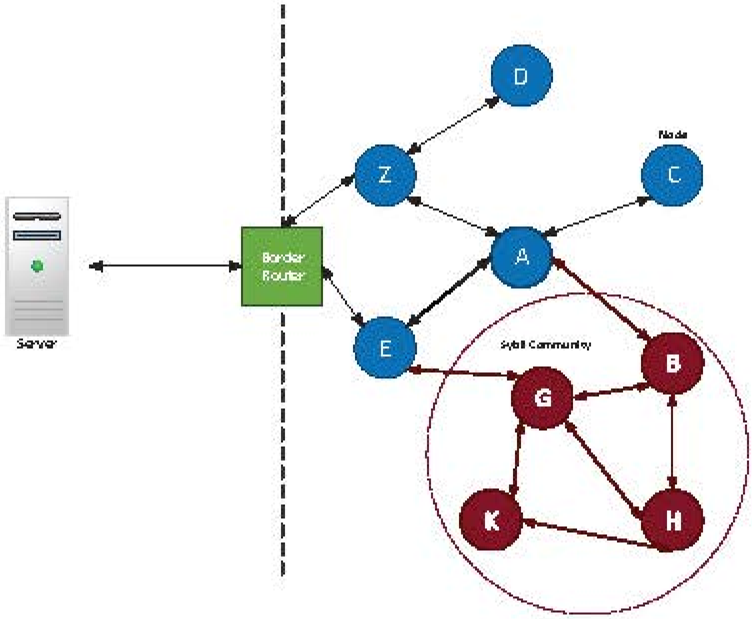


FIGURE 2.5: SA-1 Sybil Attack

emulate such an attack against RPL. On the contrary, grayholes corrupt arbitrarily only some packets either coming from particular nodes or choosing a time interval, where the packets will be discarded. Moreover, grayholes can operate randomly, deciding which packet will be dropped or not, thus making their mitigation more difficult. In [191], M. Tripathi et al. emulate grayhole attacks against Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol, using the Network Simulator 2 (NS2). On the other side, regarding the potential countermeasures against this kind of threat, many remarkable research papers have been proposed. In particular, E. Karapistoli et al. in [80], focus their attention on the detection of selective forwarding attacks by presenting a visualisation system called SRNET. The functionality of SRNET relies on the network traffic analysis as well as on visualisation methods that aim

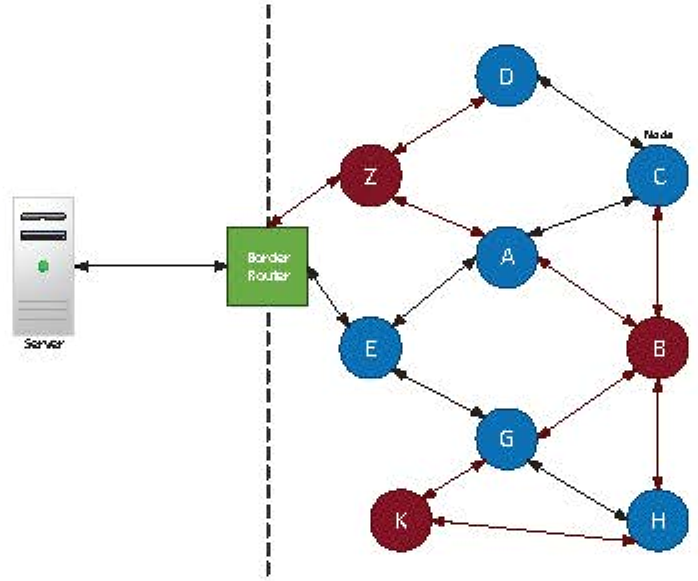


FIGURE 2.6: SA-2 Sybil Attack

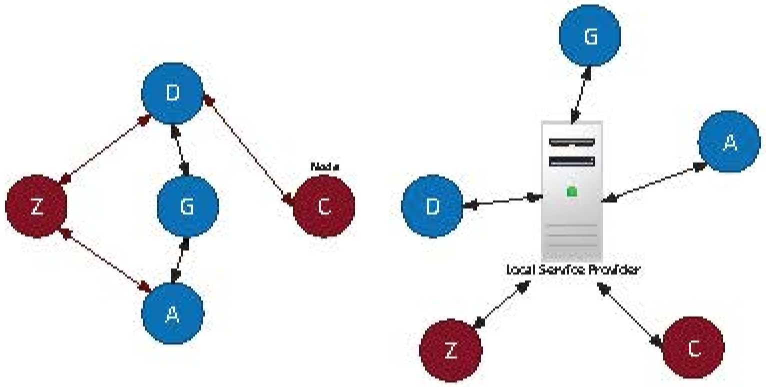


FIGURE 2.7: SA-3 Sybil Attack

to identify the root cause of these attacks. Similarly, D. Shila et al. in [172] presented a Channel Aware Detection (CAD) algorithm against grayhole attacks. The proposed algorithm relies on two strategies, namely channel estimation and traffic monitoring. Specifically, if the monitored loss rate overcomes the estimated one, the involved nodes are considered as cyberattackers.

Sinkhole Attacks: In sinkhole attacks, the goal of the attacker is to forward the network traffic to a specific node [135, 172]. In particular, the attacker promotes a particular route and attempts to persuade the other members of the network to utilise it. Usually, this route is formed via a wormhole attack, which is analysed subsequently. Fig. 2.9 depicts a sinkhole attack where the node E is the attacker, while the nodes A, B, K and Z are affected. The node E tries to advertise itself in order to receive the network packets of the other ones. The specific attack type is not very hazardous; however, when it is combined with other routing attacks, such as a wormhole attack, it can have a significant impact. In particular, a sinkhole attacker has the ability to violate all the essential security principles since it can

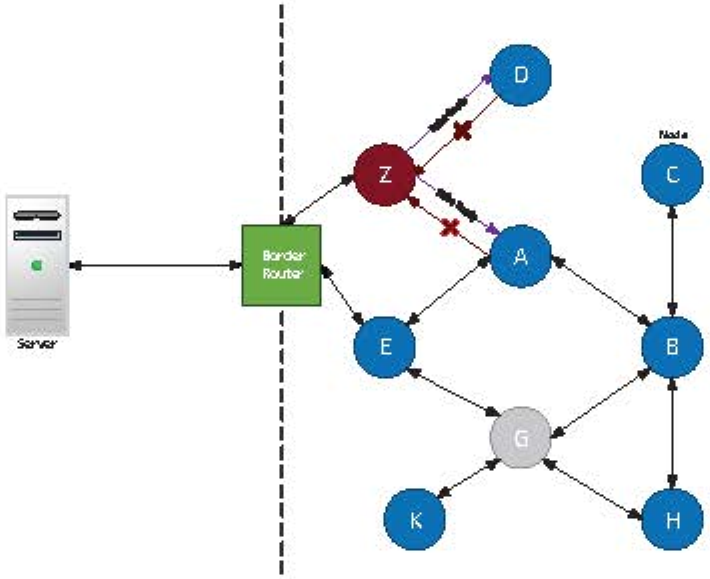


FIGURE 2.8: Selective Forwarding Attack

modify, drop or delay the various packets. A sinkhole attack can be classified into three categories: (a) Sinkhole Message Modification, (b) Sinkhole Message Dropping and (c) Sinkhole Message Delay. In the first category, the attacker modifies the packet before re-transmitting them. Accordingly, in the second category, the attacker drops the packets entirely or selectively. Finally, the third sinkhole attack delays the packet forwarding. In [200], L. Wallgren et al. emulates a sinkhole attack against RPL, usually adopted in the IoT networks. On the other side, S. Raza et al. in [151] presents an IDPS called SVELTE, which can detect such kinds of attacks in IoT networks. Finally, Y. Li et al. in [113], present the Probe Route based Defense Sinkhole Attack (PRDSA) scheme, which is capable of detecting, locating and bypassing a potential sinkhole. More specifically, PRDSA combines minimum hop routing, equal-hop routing and far-sink reverse routing, thus circumventing sinkhole attacks and discovering a safe route.

Wormhole Attacks: In a wormhole attack, the goal of the intruder is to capture the network packets, transmit ("tunnel") them in a specific node (destination node) and then selectively drop or replay them to the network [135]. In order to establish a wormhole, the attackers should construct with each other a direct communication link through which the packets will be transmitted with a better efficiency compared to the normal communication paths in terms of various network metrics, such as throughput, latency and network speed [66]. Fig. 2.10 depicts a wormhole attack formed between the nodes H and Z. It is worth mentioning that if the two collaborating members of a wormhole do not intend to compromise the network security, then the wormhole does not refer to a cyberthreat and can be used for useful purposes. On the other side, it should be noted that a potential attacker is in an advantageous position with the ability to manipulate the network packets maliciously in a variety of ways. For instance, due to the nature of the wireless networks, the attacker is able to monitor and transmit maliciously the packets exchanged among the other nodes. Therefore, wormhole attacks constitute a critical threat, especially for the ad hoc networks. In [200], L. Wallgren simulates a wormhole attack

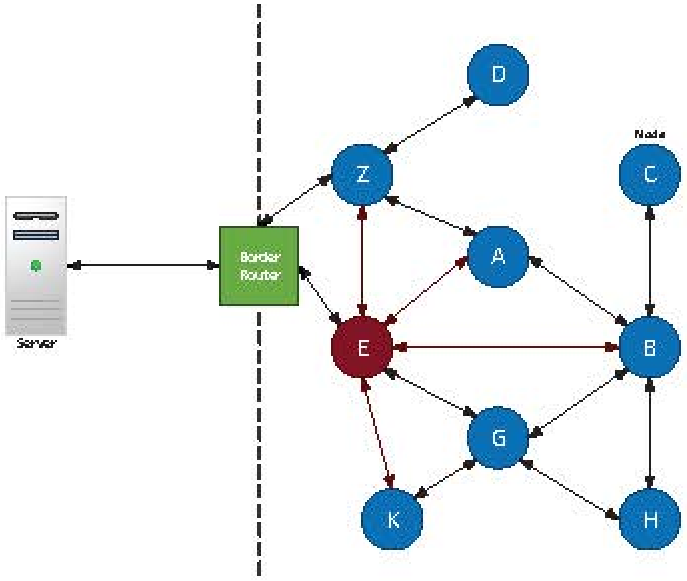


FIGURE 2.9: Sinkhole Attack

against RPL based on Contiki OS and Cooja Simulator. On the other hand, in [194], N. Tsitsiroudi et al. present EyeSim, a visual-based IDPS capable of detecting wormholes. Similarly, in [81], E. Karapistoli et al. describe another visualisation-based anomaly detection method named VA-WAD, which adopts routing dynamics in order to expose potential wormhole attackers.

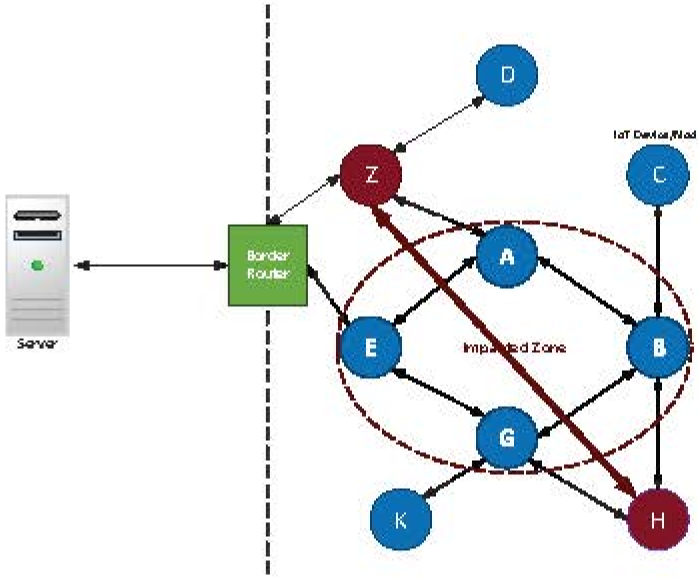


FIGURE 2.10: Wormhole Attack

HELLO Flood Attacks: Typically, the HELLO messages are used by a node in order to introduce or advertise itself to the other nodes of the network. Nevertheless, this kind of message can also be used in a malicious manner, aiming either to exhaust the computing resources of the nodes or to mislead them, considering the attacker as a neighbour [135]. Fig. 2.11 illustrates a HELLO flood attack, where the node Z plays the role of attacker, sending HELLO messages to the other nodes. L. Wallgren et al. in [200] simulate HELLO flood attacks against RPL. Based on their experimental results, although at the beginning, the HELLO flood attack was successful, after the activation of the RPL self-healing mechanism, the attack was mitigated fast. Hence, this attack cannot last for a long time, as the routing protocols include sufficient mitigation services. Similarly, in [171], M. Sharma et al. emulate also routing attacks against RPL, including also the HELLO flood attacks. Thus, a labelled dataset was constructed and used for ML and DL-based detection mechanisms. Finally, T. Srivinas and S. Manivannan in [178] provide an ML model capable of countering HELLO flood attacks. In particular, their model adopts (a) k-paths generation, (b) Cluster head selection, (c) HELLO flooding attack detection and (d) optimal shortest path selection.

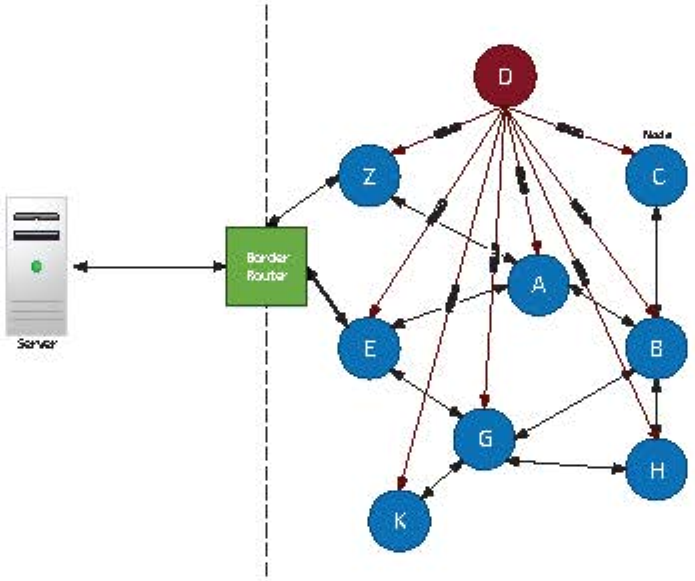


FIGURE 2.11: Hello Flood Attack

Passive Network Traffic Analysis: A passive network traffic analysis attack includes the capturing and analysis of the network packets exchanged in a network. In particular, this kind of attack requires from the attacker to activate the promiscuous mode of the Network Interface Controller (NIC) in order not to ignore those packets that are not destined to the attacking machine [135]. There are many software applications that can be used for implementing this attack, such as Wireshark, Tcpdump and Scapy. The previous applications are composed usually of two main elements called (a) sniffer and (b) protocol analyser. The sniffer undertakes to capture and copy the network traffic data while the protocol analyser decodes, processes and analyses the various packets.

Man In The Middle Attacks: The MITM attacks refer to a kind of network traffic eavesdropping. In particular, the attacker can monitor and handle the network packets exchanged between two or more parties. Characteristic examples of MITM attacks are ARP poisoning, session hijacking, and Secure Sockets Layer (SSL)/TL MITM cyberattacks. With respect to ARP poisoning, although ARP is widely used in any internal computer network, it does not include authorisation mechanisms. In particular, a potential attacker can change the victims' ARP tables, associating the IP address of a system with another forged Medium Access Control (MAC) address. Thus, the attacker can access confidential information. Session hijacking refers to the malicious activities that allow a potential attacker to impersonate a party of a session by sniffing the relevant network traffic. Finally, according to M. Conti et al. in [36], SSL/Transport Layer Security (TLS) MITM attacks can be discriminated into two main categories: (a) MITM based on a certificate and (b) MITM based on the private key. Regarding the first category, the attacker either possesses a certificate of the target system by compromising the respective Certificate Authority (CA) or differently an invalid certificate can be used. In the second case, the victim should ignore the relevant security warnings, which is a common phenomenon.

2.7.3 Security Threats at the Support Layer

The support layer offers key technologies, such as cloud and edge computing supporting heavy computational services in terms of data storage and computing power. However, despite their valuable benefits, both of them are related to critical security threats, such as unauthorised access, malicious insiders, insecure software services and unknown risk profile. The aforementioned threats are further analysed below.

Unauthorised Access and Malicious Insiders: The unauthorised access refers to accessing and/or using illegally the computing resources of an organisation or environment. Based on the nature of the cloud and edge computing, it is necessary for the involved users and organisations to provide an unusual level of trust to the cloud/edge providers [145]. Consequently, this kind of threat can compromise all the security requirements discussed above with critical consequences depending on the actions of the malicious users.

Insecure Services and Unknown Risk Profile: Both cloud and edge computing provide a wide range of computing services, such as applications, virtual machines, storage services and containerised applications. Such services might be compromised by a cyberattack or malware. Furthermore, it is noteworthy that the various services can be controlled by external providers. For instance, a cloud provider can use the computing and storage resources of another cloud provider. It is obvious that the security level of the cloud/edge services can rely on the security measures of other organisations, services and entities. Therefore, insecure services and unknown risks can occur.

business

2.7.4 Security Threats at the Business Layer

The business layer focuses on applications based on the users' needs and requirements. Therefore, the respective threats intend to exploit the security gaps of the business applications. Characteristic examples are buffer overflows, backdoors, social engineering techniques and web attacks. Although the social engineering attacks do not exploit a weakness or vulnerability, they adopt and use information about the nature of the application in order to mislead and violate potential victims. The aforementioned cyberattacks are further detailed below.

Buffer Overflow: According to NIST, a buffer overflow allows the intruder to insert more data in a buffer than the capacity limit allows. The attacker aims to overwrite the existing information in the buffer in order to insert a malicious code that will take control of the overall system. Some indicative examples of buffer overflow attacks are stack overflow, global data area overflow, format strings overflow, heap overflow and integer overflow. Commonly, a cyberattacker adopts assembly in order to perform a buffer overflow attack.

Backdoor: A backdoor is a code segment in software that enables a cyberattacker to bypass potential security controls. A backdoor is usually activated when particular credentials are used or a specific sequence of events is performed. It is noteworthy that a backdoor is not necessarily a security threat since the security administrator can use a backdoor to bypass time-consuming procedures and control or restore the normal operation expeditiously. However, extremely adverse effects can be caused if a cyberattacker is aware of the specific block of code. The malicious backdoors usually act as a network service allowing the cyberattacker to connect to an unusual network port and execute malicious activities.

Social Engineering Attacks: Social engineering is a psychological attack aiming to mislead the users to reveal confidential information or unwittingly perform malicious activities. A phishing attack is the most productive social engineering technique. The goal of the attacker is to gain the trust of the user by using spoofed emails, instant messages and DNS spoofing processes. Usually, the users are re-directed to a fake website that prompts them to provide sensitive information. Spear-phishing is a more dangerous case of this attack. In this kind of attack, the attacker has investigated the recipients thoroughly, and each fake message is carefully generated and sent in order to suit the recipient profile. With respect to the security requirements, the social engineering techniques focus mainly on confidentiality, integrity and authenticity. Although there are efficient countermeasures against such techniques like security management policies and training processes, their probability of occurrence is very high, and their consequences can be destructive.

Web Application Attacks: As in the case of all software, web applications can present severe security issues if they are not properly sanitised. For example, misconfigured authentication and authorisation web services can lead a cyberattacker to violate important unauthorised information. Characteristic examples of web-application attacks are (a) malicious proxies, (b) Structured Query Language (SQL) injection, (c) Local File Inclusion (LFI), (d) Remote File Inclusion (RFI), (e) command execution attacks and (f) Cross-Site Scripting (XSS) attacks. A proxy is a hardware or a software component placed between two communication parties to monitor and control their communications. In particular, the role of the proxy is to receive the messages coming either from the client or the server and forward them, respectively. Therefore, the proxy has the capability to capture and control the exchanged network traffic between the parties. If a proxy has not been instantiated by a potential cyberattacker, then it can enhance the overall security and Quality of Service (QoS) of this interaction. However, on the other hand, since a proxy operates as an intermediary, it can be used for MITM attacks. Next, the main goal of SQL injection attacks is to bulk extraction of data. For instance, an attacker can try to dump database tables, including customers' personal information. However, SQL injection attacks can also be used to modify or delete the content of a database, execute DoS attacks or launch malicious operating system commands. In particular, such attacks can occur when SQL commands are filtered wrongfully for escaped characters or the types of the various fields in the SQL database are not very strong, thus allowing the attackers to create combinations capable of returning or modifying unauthorised content. Another web application attack is LFI, which allows a cyberattacker to access files without appropriate permissions. Moreover, this vulnerability can induce more dangerous consequences, such as the creation of a reverse shell for the attacker, thus providing him/her with the overall control of the infected target system. RFI is similar to LFI, enabling the cyberattacker to perform malicious scripts located everywhere in the target system. Next, a command execution attack is another web attack, giving the ability to a cyberattacker to execute remotely malicious commands. For instance, an IoT application with a registration service could execute specific commands organising the content of each user. If the appropriate security measures have not been applied, a malicious user could exploit this vulnerability by introducing a suitable code block, which in turn will allow to perform various operations, such as the creation of a reverse shell. Finally, XSS allows the attacker to inject a malicious Javascript code into a web application. When the malicious code is executed, it will affect the client using the web application. There are three main types of XSS: (a) persistent/sored XSS, (b) reflected XSS and (c) DOM-based XSS. The persistent/sored XSS is stored persistently in the web application. Therefore, each time, the legitimate user accesses and uses the web application, the malicious code is executed. On the other side, the malicious code of a reflected XSS is executed when the victim accesses a specific Uniform Resource Locator (URL) created by the cyberattacker. Finally, the malicious code in the case of the DOM-based XSS is executed by the client side without interacting with the web server.

2.7.5 Multi-Layer Security Threats

This subsection is devoted to describing threats that can be executed in more than one layer. In particular, three kinds of threats are described: (a) Cryptanalytic attacks, (b) Malware and (c) APTs.

Cryptanalytic Attacks: Cryptanalysis refers to an attacker trying to discover the original message (i.e., plaintext) from the scrambled message (i.e., ciphertext). Although the cryptanalytic attacks focus mainly on violating confidentiality, they can also affect the integrity and authenticity of the target system. Indicative examples of cryptanalytic attacks are: (a) Known-plaintext attack, (b) Ciphertext-only attack, (c) Chosen-plaintext attack, (d) Chosen-ciphertext attack and (e) Chosen-ciphertext attack. It is noteworthy that despite the fact that communication protocols adopt efficient encryption mechanisms at the session sublayer, the evolution of quantum computing threatens and solves easily the mathematical problems behind the existing encryption mechanisms. In [1], D. Aggarwal discusses quantum attacks against Bitcoin, while in [46], T. M. Fernadez-Xaramez provides a survey discussing quantum-resistant cryptosystems for securing the communications of the IoT.

Malware Attacks: Malware refers to a malicious program inserted into a system, targeting all the security requirements mentioned earlier. There are various kinds of malware, such as adware, attack kits, downloaders, exploits, spyware, botnets, rootkits and ransomware. In particular, adware refers to a malicious program, which advertises its presence through pop-up ads or re-directing a browser to a particular webpage. An attack kit denotes a set of attacking tools used to generate malware in an automatic manner. Next, a downloader installs other items on a computing system under attack. An exploit refers to a malicious program leveraging specific vulnerabilities. Spyware is a kind of malware devoted to collecting and propagating useful information from the target system, such as credentials, screenshots and keystrokes. A bot intends to take under control the resources of a computing system and use them to perform malicious activities. For example, the infected system can participate in a group of bots (i.e., botnet) executing a DDoS attack. Subsequently, a rootkit allows the attacker to get access to the infected system with administrator permissions. Finally, ransomware is a kind of malware encrypting the files of a device, thus making it unusable. Characteristic examples of ransomware are Petya, WannaCry and Locky. In [20], S. Aurangzeb presents a survey about ransomware, while in [123], A. Costin et al. provide a comprehensive analysis of IoT-related malware.

Advanced Persistent Threats: An APT does not refer to a particular threat but denotes organised and persistent multi-step malicious campaigns against a particular target over a long period. Usually, several attackers cooperate with each other in order to execute an APT, using a big number of computing resources. The targets originate mainly from the political and business area. Appendix B summarises the main APT campaigns against the energy sector based on MITRE ATT&CK. Moreover, in [103], A. Lemay et al. present a survey about APT campaigns, while in [13], A. Alshamrani et al. provide a detailed analysis with respect to modelling, detecting and mitigating APTs.

2.8 Existing Security Solutions in the Internet of Things

Based on the aforementioned threats, this section describes relevant countermeasures. While the ideal solution is to prevent and mitigate the various attacks in a timely manner, this objective is generally not feasible given the presence of zero-day vulnerabilities and unknown anomalies. Therefore, for each layer, the respective countermeasures are presented. In particular, with respect to the perception layer, physical security measures are described. On the other side, regarding the communication layer, the encryption mechanisms of the respective protocols, and other defensive mechanisms, such as firewalls and IDPS are discussed. In the same way, security measures are foreseen regarding the support and business layer.

2.8.1 Countermeasures at the Perception Layer

The security measures at the perception layer intend to prevent and mitigate the corresponding threats, such as natural disasters, environmental threats and human-caused malicious activities. According to OWASP, physical security remains one of the top ten IoT weaknesses. On the one side, physical security measures like infrastructure design, mitigation plans, restoration mechanisms and personnel training can effectively handle the natural disasters and the environmental threats. On the other side, the first step to counter the human-caused malicious activities is to ensure that only legitimate users can access the IoT entities and their information. Consequently, authentication, access control and trust management mechanisms are necessary. Characteristic authentication schemes are password-based schemes, token-based schemes (e.g., electronic keycards, smart cards) and static or dynamic biometric systems (e.g., recognition by fingerprints, retina, iris, facial characteristics, hand geometry, voice). Next, access control defines the access permissions of the authenticated users and entities.Finally, trust management aims to establish a secure environment consisting only of trusted entities and users.

2.8.2 Countermeasures at the Communication Layer

The security of the communication layer relies mainly on the encryption mechanisms used by the IoT protocols. Therefore, this subsection aims to provide an overview of how such mechanisms operate in the context of the following protocol: (a) IEEE 802.15.4, (b) ZigBee, (c) Z-Wave, (d) Bluetooth Energy (BLE), (e) LoRa Wide Area Network (LoRaWAN), (f) 6LoWPAN, (g) RPL, and (h) DTLS. Based on Fig. 2.2, each protocol mentioned above is related to a particular sublayer. It is worth mentioning that application-sublayer protocols are not discussed in this chapter since depending on their nature and attributes, the data encryption relies mainly on protocols of the session sublayer. Finally, the role of additional countermeasures, such as firewall and IDPS is described.

IEEE 802.15.4 Security: The IEEE 802.15.4 protocol is a typical option for short-range communications in IoT environments. In particular, it controls and ensures the data transmission at the Physical and Data Link sublayers. First, at the Physical sublayer, it controls the radio frequency, the energy consumption, the signal management and the communication channel. With respect to the Data Link sublayer, apart from the data processing, IEEE 802.15.4 supports additional services, such as node association, security mechanisms and packets validation. The security mechanisms of IEEE 802.15.4 are not mandatory and related only to the Data Link sublayer. In particular, the Frame Control field includes a bit named Security Enabled Bit (SEB), which specifies whether the security services of the Authentication Security Header (ASH) will be activated or not. ASH is responsible for constructing the encryption key and combining the encryption algorithms depending on the security requirements of the IoT applications. For instance, the IoT applications demanding only confidentiality can use the security mode: Advanced Encryption Standard (AES) in Counter (AES-CTR). Next, the IoT applications demanding both confidentiality and integrity can use the security mode: AES in the Cypher Block Chaining (AES-CBC). Finally, the IoT applications requiring confidentiality, integrity and authenticity can utilise the security mode: AES in the combined Counter with CBC (AES-CCM). It is also noteworthy that IEEE 802.15.4 is resilient against replay attacks since the sender can split the original message into 16 blocks encrypted through a nonce or an initialisation vector. Finally, IEEE 802.15.4 incorporates Access Control Lists (ACLs) defining the access permissions. However, IEEE 802.15.4 cannot handle forged acknowledgement packets, thus allowing a cyberattacker to execute DoS attacks. Moreover, the ACLs cannot efficiently manage records using the same encryption key. Consequently, a cyberattacker can re-use a nonce resulting in the plaintext. Finally, IEEE 802.15.4 cannot support all keying models.

ZigBee Security: As IEEE 802.15.4, ZigBee is utilised in short-range communications, paying special attention to energy consumption. ZigBee is composed of four layers: (a) Physical layer, (b) Data Link Layer, (c) Network Layer and (d) Application Layer corresponding to the sublayers of the communication layer. Moreover, ZigBee uses three main entities, namely (a) Coordinator, (b) Routers and (c) end-nodes. The coordinator undertakes to establish and initialise the ZigBee network, configuring the communication channel and handling the permissions of the other entities. In addition, the coordinator is responsible for orchestrating the security mechanisms, thus determining continuously which end-node can access the ZigBee network. The routers are responsible for the intermediate communication between the end-nodes and between the coordinator and the end nodes. Finally, the end-nodes refer to the IoT entities. They can communicate with each other only through the routers, and they can also operate in a sleep mode, thus minimising the energy consumption.

With respect to the security mechanisms of ZigBee, the IEEE 802.15.4 encryption is adopted, using the AES-CCM* mode, which is a variation of AES-CCM. In particular, AES-CCM* offers the capability to select either encryption or authentication while both of them are applied necessarily in AES-CCM. Reagrding the other ZigBee layers, different security models can be chosen. The centralised security model is the most reliable option, using five encryption keys. First, regarding the network layer, a 128-bit key is adopted and shared with all entities either without encryption or using additional encryption measures. The first case is a well-known ZigBee weakness. In the second case, a global link key is used to encrypt the network key. The global link key is predefined between the coordinator and the end nodes. This key is also used when a new node enters the ZigBee network. In a similar manner, a unique link key is utilised with respect to the communication of the coordinator and an end node that is not a member of the established ZigBee network. This key is predefined between the coordinator and the end node. Furthermore, the coordinator or otherwise the Trust Centre (TC) uses a different key, called TC link key regarding its communication with the end nodes. The TC link key is randomly generated by the coordinator. Finally, with respect to the application layer, an application key is utilised and encrypted via the network key. The application key is also generated by the coordinator. It is worth mentioning that Message Integrity Code (MIC) is adopted, thus ensuring the data integrity. Finally, regarding the replay attacks, the ZigBee entities use a frame counter which is increased when a new frame is received.

Despite the aforementioned defensive mechanisms, it is evident that they cannot fully address DoS and replay attacks. For instance, the AVR RZ Raven Universal Serial Bus (USB) can be used either as a ZigBee Personal Area Network (ZPAN) or end node to sniff and capture the ZigBee network traffic and, accordingly, the network key if it is not appropriately encrypted. Also, Killerbee can be used to intercept and analyse the ZigBee packets. In particular, Killerbee consists of three modules: (a) zbdsniff, (b) zbstumbler and (c) zbassocflood. zbdsniff is responsible for capturing and analysing the ZigBee network traffic. zbstumbler is a ZigBee network discovery tool and zbassocflood can flood a ZigBee entity with multiple packets. Finally, based on N. Vidgren et al. in [198], another usual attack against the ZigBee entities is a DoS targetting the battery lifetime of the end nodes.

Z-Wave Security: Z-Wave is a proprietary technology, which is designed for short-range IoT communications. Z-Wave is provided by the Z-Wave Alliance, which includes more than 600 companies. Characteristic examples are Huawei and Siemens. In particular, Z-Wave is deployed in a mesh network utilising a four-layer architecture: (a) Physical Layer, (b) Data Link Layer, (c) Network Layer and (d) Transport Layer [117]. The Physical and Data Link layers have been standardised as the G.9959 standard by ITU. Z-Wave can interconnect 232 devices. In particular, the IoT entities in a Z-Wave network are divided into two categories: (a) controllers and (b) slaves. The controllers are responsible for the network management by determining the respective specifications and controlling whether a new device can join the network or not. Moreover, a primary controller specifies a unique identifier for the network. On the other hand, the slaves represent typical IoT devices. With respect to the security, Z-Wave categorises the security measures into two main classes: (a) Security 0 (S0) and (b) Security 2 (S2) [145]. Furthermore, S2 consists of three subclasses: (a) S2-Access Control subclass, (b) S2-Authenticated subclass and (c) S2-Unauthenticated subclass [117]. S2-Access control is considered the most secure option, while S2-Unauthenticated and S0 focus on very constrained and legacy devices, respectively. Apart from S2-Unauthenticated subclass, the security of the previous classes and subclasses use AES-128 CCM encryption and authentication processes. Regarding the key exchange process, S2 allows sharing only between the devices of the same subclass. For example, a device belonging to S2-Access Control cannot exchange the network key with a device of the S2-Unauthenticated subclass. The key exchange process of the S2 class is conducted via the Curve25519 model, which is considered a safe option. However, a side-channel attack against this model was recently discovered [145]. On the other side, the Elliptic Curve Diffie Hellman (ECDH) scheme is utilised for the key exchange process of the S0 class. Finally, S2 provides AES-128 Cipher-based Message Authentication Code (AES-128-CMAC) and predetermined nonces, thereby ensuring the data integrity and the protection against the replay attacks, respectively.

So far, there are no specific security issues against Z-Wave. There are only some successful cyberattacks against specific implementations. In particular, Z-Wave allows the communication with legacy devices that may not include sufficient security mechanisms. This fact can lead to various security threats, such as replay attacks. Moreover, although Z-Wave integrates AES-128 encryption, in many cases, the manufacturers do not activate this kind of encryption. In [145], the authors tested various Z-Wave devices. They state that only 9 of 33 incorporate the available security measures. Moreover, they demonstrate a successful cyberattack against a Z-Wave-based door lock application by exploiting a vulnerability of the key sharing process. The severity score of this vulnerability is 6.5 according to the CVSS [29]. Finally, in [87], M. Smith introduces a tool called EZ-Wave. EZ-Wave can perform various penetration tests against Z-Wave. The efficiency of this tool is demonstrated by turning on and off various bulbs of a Z-Wave network, thus leading to their destruction. The CVSS score of this vulnerability is 6.5. In conclusion, Z-Wave provides valuable security mechanisms that can largely guarantee the security of the network. The manufacturers and vendors should always follow the security updates, configuring appropriately the corresponding devices.

BLE Security: BLE is a variation of Bluetooth to support short-range communications, especially for constrained IoT devices, providing them with the ability to form wireless networks, called piconets. Bluetooth was introduced under a nonprofit consortium of many organisations and companies, called Bluetooth Special Interest Group (SIG). More specifically, BLE was generated from the Bluetooth 4.0 specification and subsequently, the specifications 4.1 and 4.2 updated its characteristics. The architecture of a BLE piconet mainly consists of two kinds of entities: (a) master nodes and (b) slave nodes. The master node is responsible for initiating the network, while the slave nodes are power-constrained entities sensing the physical environment. It is noteworthy that a slave node can be a master node in a different piconet. A chain of piconets is named scatternet [145]. Moreover, BLE allows the existence of broadcasters and observers that periodically broadcast and listen to messages, respectively. Finally, BLE allows the communication up to 50m, while the maximum data rate is calculated to 1Mbps.

With respect to security, BLE focuses on authentication, confidentiality, integrity and pairing. Pairing refers to the secret key generation and storage used for the encryption and authentication procedures of BLE. There are three keys distributed: (a) Long-Term Key (LTK), (b) Identity Resolving Key (IRK) and (c) Connection Signature Resolving Key (CSRK). LTK is utilised for the encryption mechanisms. IRK and CSRK are responsible for determining private addresses and data signing, respectively. LTK is divided into Master LTK (MLTK) and Slave LTK (SLTK). In particular, two security modes are defined. The first security mode (Security Mode 1) includes four levels. The first level does not integrate any security mechanism. The second level includes encrypted communication, but it does not require any authenticated pairing. The third level requires both authenticated pairing and encryption processes. Finally, level 4 introduces upgraded encryption and authentication processes, called Secure Connections. On the other side, the second security mode (Security Mode 2) comprises two levels related to the signing processes. In particular, the first level defines data signing with non-authenticated pairing, while the second demands authenticated pairing and data signing. Regarding the pairing process, there are four models: (a) Numeric Comparison, (b) Passkey Entry, (c) Just Works and (d) Out of Band (OOB). The devices following the specifications 4.0 and 4.1 use a legacy pairing process. Thus, the devices use first a Temporal Key (TK) to exchange some random values and then, based on TK, they generate a Short Term Key (STK) used to distribute securely LTK, IRK and CSRK. On the other side, the devices following the specification 4.2 use the pairing process defined by the Secure Connections. In this case, the LTK is not distributed but is generated autonomously in each device utilising AES-128-CMAC. Subsequently, LTK is used to distribute securely IRK and CSRK. It is worth mentioning that in contrast to specifications 4.0 and 4.1, specification 4.2 enhances the security of the pairing process through the addition of AES-128-CMAC as well as the P-256 Elliptic curve. Finally, concerning the data confidentiality, BLE adopts AES-CCM, while there is not an explicit authentication mechanism, as the encryption of the link satisfies the authentication process.

Despite the above security mechanisms, BLE presents various security vulnerabilities. First, through a replay attack, the attackers can violate the legacy pairing process by capturing LTK, IRK and CSRK. In [97], G. Kwon et al. demonstrate this vulnerability by predicting and identifying TK within 20 seconds. The CVSS score of this vulnerability is rated at 7.4. Furthermore, a crucial issue is that the first level of Security Mode 1 does not incorporate any security mechanism [145]. In addition, although specification 4.2 introduced mechanisms ensuring several security requirements, the manufacturers and vendors have the ability to adjust and change the security level, thus leading to potential weaknesses [145]. Finally, the specifications themselves are characterised by high complexity, resulting in several security issues and weaknesses [145].

LoRaWan Security: LoRaWan was initially adopted to enhance the functionality of Low Power Wide-Area Networks (LPWAN) regarding mainly the consumption capability, storage capacity, long-range communication and transmission cost. The architecture of LoRaWan relies on four main entities: (a) end nodes, (b) gateways, (c) network server and (d) application server. The end nodes are usually

IoT entities collecting information from the physical environment and transmitting it to gateways via the LoRa physical layer. In turn, the gateways send this data to a network server. This communication is achieved through various protocols, such as IEEE 802.3 (Ethernet) and IEEE 802.11 (Wi-Fi). The network server is responsible for controlling the data by executing the appropriate security operations and checking redundant packets. Finally, it transmits the data to the application servers representing software applications.

Regarding the security measures of LoRaWan, it includes two security layers. The first one undertakes to authenticate the end nodes. In particular, the authentication process is conducted through an AES-CTR 128 secret key, called Network Session Key (NwkSKey). This key is utilised between the end nodes and the network server. On the other side, the second layer is responsible for assuring the privacy of end nodes by utilising an AES-CTR 128 secret key called Application Key (AppSKey). This key is used by the end nodes and the application servers. Consequently, a crucial issue for the LoRaWAN technology is the safety of the above keys. If a cyberattacker manages to steal them, then the respective data is exposed. Furthermore, concerning the communication between the end nodes and the gateways, it is worth mentioning that the payload length remains unchanged before and after the encryption process. An attacker can take advantage of this issue, trying to restore NwkSKey from the encrypted messages [145]. Moreover, an attacker with physical access to the end nodes has the ability to extract the aforementioned keys. In particular, an end node includes a LoRa radio module and an MCU. The LoRa radio module interacts with MCU, utilising Universal Asynchronous Receiver Transmitter (UART) and Serial Peripheral Interface (SPI) interfaces. However, LoRa radio module does not include embedded encryption mechanisms, thus allowing the attacker to extract the keys. To this end, external means, such as a Future Technology Devices International (FTDI) interface can be used [16]. In addition, it is noteworthy that the LoRaWAN packets do not integrate time information to verify the integrity of the messages. This issue can lead to replay and wormhole attacks. In [16], E. Aras et al. describe the process of a possible wormhole attack against LoRaWAN. Finally, in [156], B.Reynders et al. demonstrate that the LoRa transmissions are prone to jamming attacks.

6LoWPAN Security: Based on IEEE 802.15.4, the Low Power WPANs can use only 102 bytes for the data transmission with respect to the other communication layers. However, the value of the maximum transmission unit required for IPv6 is equal to 1280 bytes. The purpose of 6LoWPAN is to solve this issue by deploying an adaptation layer between IEEE 802.15.4 and IPv6. This adaptation layer takes advantage of compression, fragmentation and encapsulation mechanisms and transmits the IPv6 packets to the Data Link Layer.

Currently, 6LoWPAN does not provide any security mechanism, such as IPSec. However, research efforts [145] investigate how security solutions can be adopted in 6LoWPAN, by designing, for instance, compressed security headers for 6LoWPAN, as in the case of IPSec, which adopts Encapsulating Security Payload (ESP) and Authentication Header (AH). Moreover, some studies [69, 86] investigate security mechanisms against 6LoWPAN fragmentation attacks. In particular, [86] H. Kim discusses the addition of a timestamp and a nonce field to the 6LoWPAN fragmentation header in order to address such attacks. Furthermore, in [69], R. Hummer et al. present a mechanism supporting fragment authentication and preventing suspicious messages. Finally, a significant security addition to the 6LoWPAN standard is the key management since the security keys must be regularly renewed in order to assure the principles of confidentiality, integrity and authenticity. Towards this end, the Internet Key Exchange version 2 (IKEv2) protocol could be used.

RPL Security: The RPL protocol was created by the Internet Engineering Task Force (IETF) for routing messages in Low Power and Lossy Networks (LLNs). Its operation relies on a Destination Oriented Directed Acyclic Graph (DODAG) utilising an objective function [200]. In particular, DODAG consists of a set of nodes possessing oriented edges in order not to create loops. The creation of a DODAG starts when the root node transmits a DODAG Information Object (DIO) message to its neighbours. The neighbouring nodes receive the DIO message and take the decision whether they can join the graph or not. If a node does join the graph, then the corresponding path to the root node is created. Next, using the objective function, the new node of the graph calculates a value called "rank". This procedure is repeated for each node within the graph. Finally, it is worth mentioning that the nodes can transmit a DODAG Information Solicitation (DIS) message in order to discover new DODAGs. On the other side, they can also send DODAG Destination Advertisement Object (DAO) messages to advertise a routing path.

The security mechanisms of RPL rely mainly on the variations of the RPL messages, such as DIS, DIO, DAO, DAO-Acknowledgement (DAO-ACK). The variations can guarantee message integrity, replay protection, delay protection and confidentiality. In particular, the cryptographic methods behind RPL are identified by the security field. Moreover, RPL supports three security modes: (a) unsecured, (b) preinstalled and (c) authenticated. The first one is the default choice without including any security mechanism. In the second mode, the nodes hold a pre-configured key used to connect to DODAG as a router or host. Finally, the third mode adopts a pre-configured key as in the previous case and another one from a validation authority. Despite the fact that RPL includes valuable security mechanisms, the routing attacks discussed above remain a severe issue.

DTLS Security: DTLS is a variation of the TLS protocol assuring the presence of confidentiality, integrity and authenticity in IoT environments at the session sublayer. DTLS operates over datagrams that can be lost, duplicated or received in a wrong order. For this reason, DTLS supports some additional measures compared to TLS. First, the TLS record protocol is enhanced with two additional fields: (a) an epoch and (b) a sequence number. Second, DTLS does not allow stream cyphers. Finally, the TLS handshake protocol is improved with the addition of a stateless cookie, thus addressing potential fragmentation, message loss and reordering issues. The Request for Comments (RFC) 6347 document [145] describes in detail the aforementioned measures.

Several IoT application-layer protocols rely on the security mechanisms of DTLS. However, DTLS is characterized by some limitations. First, DTLS cannot support some IoT communications since large messages have to be fragmented at the 6LoWPAN adaptation sublayer. Thus, some packets have to be re-transmitted. Moreover, the preparation and transmission of the finished DTLS messages are computationally expensive with respect to the resources of the IoT entities. Furthermore, DTLS cannot be used yet with some IoT protocols, such as Modbus and DNP3. Finally, it is also noteworthy that the current DTLS version cannot support multicast communications.

Firewalls: A firewall is a protection system in the form of hardware or software that continuously monitors and controls the network activities based on predefined rules. A firewall can monitor the network traffic data with respect to the attributes of the communication protocols at the various sublayers. This choice relies on the firewall policy defined by risk management and assessment procedures. In particular, a firewall can be classified either by its operation mode or its placement. Regarding the first case, four categories are defined: (a) packet filtering firewall, (b) status inspection firewall, (c) application-level gateway and (d) circuit-level gateway. Each category mentioned earlier is further analysed in [145]. On the other side, a firewall can be installed in various locations. For instance, the IoT entities can carry a lightweight firewall system. Otherwise, a centralised IoT entity can also play the role of a firewall monitoring the activities of the entire IoT network. In [143], P. Radoglou-Grammatikis et al. provide an overview of the firewall systems with respect to the SG ecosystem. In [55], N. Gupta et al. present a firewall using a Raspberry Pi device as a gateway, which adopts appropriate heuristic functions and signature rules.

Intrusion Detection and Prevention Systems: An IDPS system aims to detect, record and mitigate timely potential cyberattacks. For this purpose, an IDPS can monitor and process information originating from multiple sources, such as system logs, network traffic data and operational data. As in the case of a firewall, an IDPS can be categorised either by its operation mode or its placement. In the first case, three kinds of IDPS are defined: (a) signature-based IDPS, (b) anomaly-based IDPS and (c) specification-based IDPS. The signature-based IDPS utilise a predefined set of malicious patterns. The second category uses statistics and AI methods, while the specification-based IDPS adopt a set of patterns defining the normal behaviour. On the other side, there are three types of IDPS: (a) Host-based IDPS (HIDPS), (c) Network-based IDPS (NIDPS) and Hybrid IDPS. HIDPS is responsible for monitoring the system logs and activities of an individual entity. In contrast, NIDPS undertakes to monitor the entire network. Finally, the last category combines the characteristics of the previous ones. The following chapter describes in detail the various kinds of IDPS and how they work. Several papers investigate how IDPS can protect the IoT entities and apllications. For instance, in [151], S. Raza et al. present “Svelte”, an IDPS capable of detecting sinkhole, sybil and selective forwarding attacks. In [181], M. Surendar and A. Umamakeswari provide an IDPS, which focuses on sinkhole attacks. In a similar manner, in [118] D. Midi et al. present “Kalis”, an IDPS system for the IoT, combining signature-based and anomaly-based detection. Finally, in [30], C. Cervantes et al. presents “INTI”, an IDPS for 6LoWPAN. Finally, in [217], B. Zarpelao et al. provide a comprehensive survey of various IoT-related IDPS systems, while Chapter 3 presents a similar analysis about SG-related IDPS.

2.8.3 Countermeasures at the Support and Business Layers

The respective countermeasures at the support layer have to counter unauthorised access activities, malicious insiders, insecure services and potential unknown threats. Consequently, first, only the legitimate users and IoT entities should be able to access and use the services and information of the support layer. For this purpose, the presence of an authentication and access control system is necessary. Moreover, a trust management framework is paramount of importance. Next, secure programming, firewall and IDPS can prevent data loss or leakage. Finally, stringent and transparent policies with respect to the security and safety rules are essential to address and mitigate potential malicious insiders. On the other side, the security threats of the business layer are mainly related to the insecure programming and the human factor. Therefore, high-level programming languages could be adopted in order to minimise potential security gaps and vulnerabilities. Moreover, cybersecurity training and certification activities are necessary since they can protect the users from social engineering techniques.

2.9 Chapter Summary

This chapter provides an introduction to IoT, giving particular emphasis to the relevant security and privacy issues. In particular, first, the entities of IoT are briefly described, providing characteristic examples like PLCs and RTUs. Next, despite the fact that an IoT communication stack has not been standardised yet, a proposed one is discussed in terms of the various layers: (a) Perception Layer, (b) Communication Layer, (c) Support Layer and (d) Business Layer. Next, the applications and services in the IoT paradigm are presented, paying special attention to SG, the largest application in the IoT era. Subsequently, the IoT security requirements and challenges are enumerated and discussed, considering the unique characteristics and limitations of the participating entities. Next, a threat taxonomy is presented based on the aforementioned layers and the CAPEC catalogue. Finally, in a similar manner, for each layer, the relevant security solutions and countermeasures are organised and discussed.

Chapter 3

Intrusion Detection and Prevention

It is evident that despite the benefits of the IoT, the corresponding entities and applications are prone to a wide range of threats and cyberattacks. While the optimal security policy is to prevent the various attacks in a timely manner, this goal is generally not achievable due to zero-day vulnerabilities and unexpected security events. A realistic solution is the timely detection of cyberattacks and anomalies without affecting the normal operation of the IoT applications. Therefore, the presence of intrusion detection mechanisms is necessary. This chapter provides an overview about intrusion detection and prevention mechanisms. According to RFC 2828 (Internet Security Glossary), intrusion detection is related to monitoring, auditing and evaluating security events in order to detect any malicious or anomalous behaviour in a timely manner. In 1980, the term ”IDS” was coined, referring to a hardware and/or software system that automates the aforementioned activities. In 1980, James Anderson [148] concluded that the log files could be an efficient source for monitoring the health status of a computing system and how the involved users interact with it. In 1978, D. Denning et al. defined the first concrete intrusion detection model [41]. On this basis, several engineers start implementing the first IDS. This chapter summarises (a) the objectives and requirements of IDPS, (b) an IDPS reference architecture, (c) the intrusion detection techniques, (d) the role and types of honeypots, (e) the intrusion prevention techniques and finally, (f) the role of the SIEM systems and the Security Operation Centres (SOCs).

3.1 Objectives and Requirements of Intrusion Detection and Prevention Systems

According to RFC 2828 (Internet Security Glossary), a security intrusion is defined as: “A security event, or a combination of multiple security events, that constitutes a security incident in which an intruder gains, or attempts to gain, access to a system (or system resource) without having authorisation to do so.”. On the other side, intrusion detection is defined as: “A security service that monitors and analyzes system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner.”. Although the intruders intend to take advantage of new weaknesses and bypass potential countermeasures, they usually adopt a common attack methodology consisting of six steps: (a) Target Acquisition, (b) Initial Access, (c) Privilege Escalation, (c) System Exploit, (d) Maintaining Access and (e) Covering Tracks [179]. Despite the fact that the main goal of IDPS is to detect and mitigate potential intrusions in a timely manner, the constrained nature of the IoT entities and applications leads to new requirements about their role in IoT [148]. According to P. Radoglou-Grammatikis et al. in [148], these requirements are given below.

- **Detection of a wide range of cyberattacks and anomalies**: Based on the threats discussed in Chapter 2, the IDPS shall be able to detect and discriminate a wide range of cyberattacks and anomalies.
- **Timely Intrusion Detection**: Depending on the criticality of each IoT application, the corresponding cyberattacks and anomalies should be detected in near real-time.
- **High Detection Accuracy**: The detection efficiency is the most critical challenge of an IDPS in terms of recognising the various cyberattacks and anomalies in an accurate manner.
- **Lightweight Resource Scaling**: Due to the constrained nature of IoT entities and applications, an IDPS should be able to operate in an accurate and timely manner without consuming a lot of computing resources and affecting the core operation of the IoT entities and applications.
- **Scalability**: Given the size of the IoT applications consisting of multiple IoT entities, a relevant IDPS should be capable of monitoring and controlling them efficiently.
- **Resiliency against Cyberattacks**: The IDPS should be able to detect and counter cyberattacks targeting itself.
- **Friendly User Interface**: Similarly, given the large amount of data and security events in IoT applications, the respective IDPS should be able to visualise and correlate the various security events and alerts in a clear and organised manner.

3.2 Reference Architecture of Intrusion Detection and Prevention Systems

As illustrated in Fig. 3.1, a typical IDPS consists of three main components, namely (a) Agent(s), (b) Analysis Engine and (c) Response Module. The agents are responsible for monitoring the activities of the various entities, thus collecting and pre-processing the necessary data. Next, this data is sent to the Analysis Engine. It is worth mentioning that based on the placement of an agent, an IDPS can be classified into three categories: (a) HIDPS, (b) NIDPS and (c) Hybrid IDPS. In particular, an HIDPS monitors the characteristics and the activities of a single entity, such as system calls and log files. On the other hand, a NIDPS is capable of monitoring the network traffic data of a network segment. Finally, the Hybrid IDPS combines the aforementioned types (i.e., HIDPS and NIDPS). Next, the Analysis Engine receives the data from the various agents and is responsible for recognising the presence of a cyberattack or anomaly. For this purpose, three main detection techniques can be utilised, namely (a) signature-based detection, (b) anomaly-based detection and (c) specification-based detection. Each of them is further detailed below. Finally, the Response Module receives the security events and alerts generated by the Analysis Engine and notifies the user. In some cases, the Response Module can perform some mitigation and prevention activities, such as activating the appropriate firewall rules. Usually, the terms Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) are used depending on the mitigation and prevention actions provided by the Response Module. In the first case, the Response Module can generate some security messages without executing any mitigation/prevention actions. In the second case, the Response Module can perform some of the mitigation/prevention measures discussed in subsection 3.5. In this thesis, the term IDPS will be used for both options.

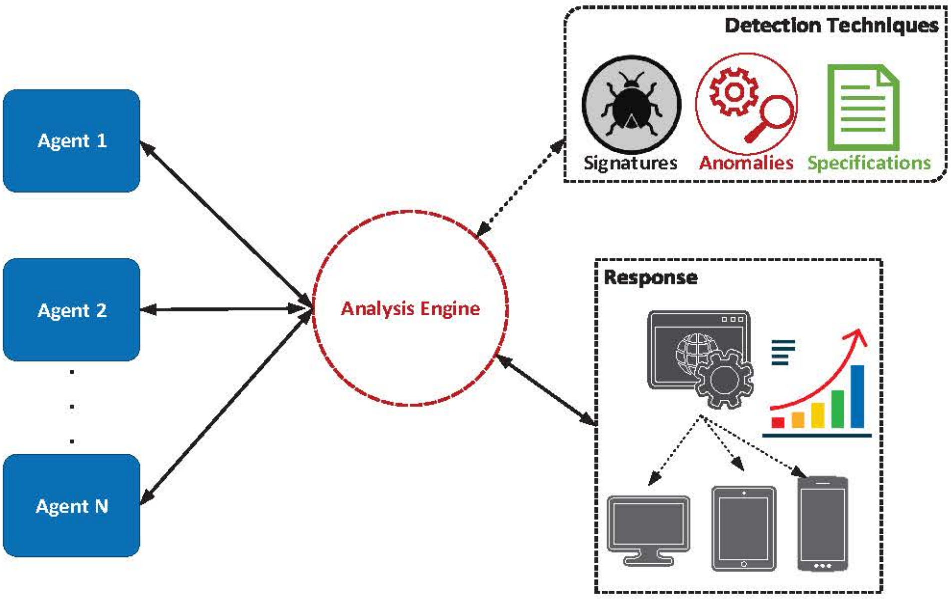


FIGURE 3.1: Typical IDPS Architecture [148]

3.3 Intrusion Detection Techniques

The function of the Analysis Engine relies on the assumption that the behaviour of an intruder differs from that of a normal/legitimate user, and this kind of difference can be quantified based on various methods. However, there is an overlap between the two behaviour profiles. Therefore, a loose interpretation of the intruder's behaviour will result in the detection of more cyberattackers but, at the same time, will lead to more False Positives (FP). On the other hand, a stricter interpretation of the intruder's behaviour will lead to more False Negatives (FN). Fig. 3.2 illustrates this overlap between the behaviour profile of an intruder and a normal/legitimate user. Consequently, it is evident that there is a practical element of compromise with respect to detecting intrusions and anomalies. Based on the methods used by the Analysis Engine, an IDPS can be classified into three categories: (a) signature-based detection, (b) anomaly-based detection and (c) specification-based detection. Each method is further analysed in the following subsections.

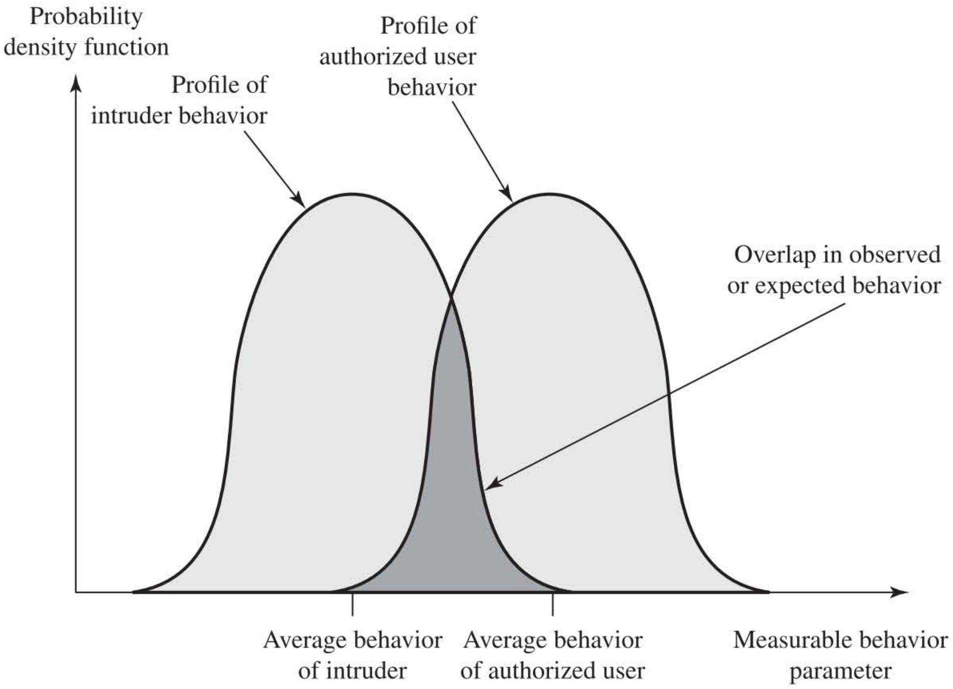

Probability density function
Profile of intruder behavior
Profile of authorized user behavior
Overlap in observed or expected behavior
Average behavior of intruder
Average behavior of authorized user
Measurable behavior parameter

FIGURE 3.2: Behaviour Profiles of an Normal/Legitimate User and an Intruder [179]

3.3.1 Signature & Specification-based Detection

The signature-based techniques (also known as misuse detection) adopt a set of known malicious patterns or attack rules (i.e., signatures) compared with the characteristics of the current behaviour, thus identifying the presence of an attacker or not. This kind of detection is characterised by a high detection accuracy and a relatively low computation and time cost; however, it cannot detect unknown anomalies and zero-day attacks. On the other hand, specification-based techniques use a set of rules (i.e., specifications) that determine the normal behaviour. Consequently, any action violating the specifications is considered an anomaly. The specification-based techniques include the aforementioned advantages and can detect unknown anomalies. However, they cannot discriminate the attack type. Based on the syntax of the patterns/rules, the IDPS is considered a signature-based IDPS or specification-based IDPS. Snort [32], Suricata [203] and Bro [195] are popular NIDPS of this category. Similarly, OSSEC [186] is a HIDPS of this category.

3.3.2 Anomaly-based Detection

The anomaly-based IDPS uses a model, which can recognise normal and malicious behaviour patterns based on statistical and AI methods. In particular, ML and DL methods are usually adopted for the implementation of this model, following a training process based on the data of the various agents. The training process can take place at distinct times or in a continuous manner, thus updating the model with the presence of new attacks and malware. Such methods can detect unknown anomalies and zero-day attacks, but they are characterised by a significant number of false alarms. Moreover, a drawback of this category is the necessary presence of an intrusion detection dataset, including malicious data samples. Due to the sensitive nature of this data (especially in the energy sector), such datasets are rarely available [139]. In addition, each environment is specified by unique characteristics and requirements, thus complicating the adoption of publicly available datasets. Despite the fact that there are many ML/DL methods, all of them follow the phases below.

- **Pre-processing Phase**: Given a pre-existing dataset (either labelled or not), in this phase, each data point/instance is pre-processed appropriately based on their features and the hyperparameters of the ML/DL methods, which will be used in the following phase. Usually, pre-processing methods like normalisation, min-max scaling, standardisation, max abs scaling and robust scaling are used [47].

- **Training Phase**: In this phase, the model responsible for the prediction/detection is trained and generated based on the pre-processed data of the previous phase. As mentioned, there are various ML/DL models for this purpose. In general, they can be classified into four main categories: (a) supervised detection [37], (b) unsupervised/outlier detection [22], (c) semi-supervised/novelty detection [197] and (d) RL [17]. The first category relies on a labelled dataset, thus including a particular label like "*Anomaly*", "*Normal*", or "*Unauthorised Activity*" for each data point/instance. Characteristics ML/DL methods for this category are Decision Trees [114], Random Forest [154] and Neural Networks [54]. On the other side, the second category can use clustering techniques [166] and unlabelled datasets based on the assumption that most of the data points/instances are normal. However, the training data can include outliers. Stochastic Outlier Selection (SOS), Isolation Forest [60] and Local Outlier Factor (LOF) [9] are indicative methods of this category. Next, in the this category, the training dataset does not include outliers, and the goal is to identify whether a new observation is an outlier or not. OneClassSVM [107] is a typical example of this category. In the last category, the goal is to create an agent interacting with the environment in order to identify the most efficient policy in terms of particular states, actions and rewards.

- **Inference**: After the training procedure, the model is ready to be used by the Analysis Engine described above. Based on the decision of the model, the relevant alert can be triggered or not by the Response Module.

Next, various ML/DL methods of the second phase are briefly described. These methods are used in Chapter 5 in a comparative study for detecting various cyberattacks against industrial protocols.

Logistic Regression: Logistic Regression [39] is a supervised ML method that is used to model and identify the probability of a particular event. It can be adopted when the data samples are linearly separable, and the prediction outcome is binary. However, it can also be used with multiclass classification problems based on the extended version of logistic Regression called multinomial logistic Regression. In summary, a model using Logistic Regression works as follows. First, a linear representation z is calculated and given to a sigmoid function $\sigma = \frac{1}{1+e^{-z}}$. Next, σ is responsible for predicting the outcome y. To evaluate the efficiency of the model, a loss function is used, calculating the error between y and the actual y'. Although the Mean Squared Error (MSE) ($MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$) is one of the most well-known loss functions [159], the outcome of Logistic Regression is between 0 and 1; therefore, MSE is not an appropriate loss function for a model relying on Logistic Regression. To this end, the Cross-Entropy Loss Function ($H(P^*|P) = -\sum_i P^*(i)logP(i)$) or Log Loss ($\frac{1}{N}\sum_{i=1}^{N} -(y_i * log(\tilde{y_i}) + (1-y_i) * log(1-\tilde{y_i}))$ can be used. In contrast to Linear Regression, Logistic Regression is suitable when there are outliers or the predicted value exceeds the range between 0 and 1.

Naive Bayes: Naive Bayes [76] is a probabilistic ML-supervised method, which relies on the Bayes' theorem $P(Y|X) = \frac{P(X|Y)P(Y)}{PX}$, assuming that the features of vector X are independent with equal importance. This means that changing the value of a feature does not affect the other features. Moreover, it is noteworthy that all the features contribute equally to the successful prediction. Consequently, based on n number of features, the Naive Bayes model can be represented as $P(Y = k|X_1, X_2, ..., X_n) \propto P(Y)\prod_{i=1}^{n}(X_i|Y)$. Supposing that values of the various features are continuous, then their distribution should be considered. Consequently, there can be various Naive Bayes classifiers based on the distribution $P(X_i|Y)$. A characteristic example is the Gaussian Naive Bayes, which relies on the assumption that the values of the various features follow the Gaussian distribution.

Support Vector Machine (SVM): SVM [31] is an ML-supervised method that can be used for classification and regression problems. However, it is usually adopted for classification problems. The goal of SVM is to create a hyperplane capable of distinguishing the various classes. The form of the hyperplane relies on the corresponding dimensionality space. For instance, in two-dimensional space, the hyperplane is represented by a single line. In a similar manner, in a three-dimensional space, the form of a hyperplane is a plane. The dimensionality of the hyperplane relies on the number of features used for the training procedure. The data points near the hyperplane in each case are called support vectors. These data points are responsible for the position of the hyperplane. The distance between the support vectors and the hyperplane is called margin. During the training procedure, one of the goals is to identify the appropriate hyperplane, which maximises the margin between the support vectors and the hyperplane. Given a linear problem, mathematically, SVM is summarised by the following equation: $min\frac{1}{2}||W||^2$, $s.t.y_i(W.X_i + b) \geq 1, \forall X_i$. For non-linear problems, various kernel functions can be used. They refer to functions that convert low-dimensional input space into a higher-dimensional space. There are various kernel functions, such as Radial Basis Function (RBF), polynomial kernel and sigmoid kernel.

K-Nearest Neighbour (KNN): KNN [38] is a lazy ML method for both classification and regression problems. It can be used as a supervised or unsupervised method. In particular, given a new data point, KNN calculates the distance between the new data point and the existing ones. Next, the distances are sorted in ascending order. Then, the first K distances are chosen. Finally, the mode or mean of the distances are calculated depending on whether the type of the problem is classification or regression, respectively. Regarding the calculation of the distance, various distance metrics can be used, such as Minkowski, Manhattan, Euclidean, Cosine and Jaccard. On the other hand, K is a hyperparameter which is chosen mainly based on the nature and characteristics of each domain. There are several algorithms implementing the KNN method, such as *kd_tree*, *ball˙tree*, *auto* and *brute*.

Decision Tree: A decision tree is another ML-supervised method for both classification and regression problems [147]. Focusing mainly on the classification problem, a decision tree can be represented as a set of `if-else` statements, categorising the various instances into particular classes based on the various features. As indicated by their name, a decision tree consists of internal nodes and leaves. On the one hand, the internal nodes represent the *if-else* statements responsible for the decision about the classification problem in terms of splitting the overall data space into smaller data spaces given the available features. This decision can rely on various criteria, such as Entropy defined by $E(S) = -p_{(+)}logp_{(+)} - p_{(-)}logp_{(-)}$ where p_+ denotes the probability for the positive class, while p_i indicates the probability for the negative class. Finally, S implies a subset of the training dataset. The Entropy indicates the degree of uncertainty related to a node. The lower the Entropy, the higher the purity of this node. Although the Entropy can identify the uncertainty of a node, it cannot provide the Entropy of the parent node. In particular, the Entropy cannot identify whether the Entropy of the parent nodes has been decreased or not. For this purpose, Information Gain: $IG = E(Y) - E(Y|X)$ is used. IG can measure the reduction of uncertainty based on the various features and play an important role as a deciding factor regarding which nodes will act as internal ones or leaves. Based on the aforementioned remarks, many algorithms can generate decision trees based on a labelled dataset, such as the Classification and Regression Tree (CART), Iterative Dichotomiser 3 (ID3), J48, Chi-square Automatic Interaction Detector (CHAID), C4.5 and Quick, Unbiased, Efficient, Statistical Tree (QUEST). In this thesis, CART is utilised. In particular, CART separates the dataset based on a single feature x and a relevant threshold t. To this end, CART searches for the best pair between x and t, which will provide the purest subsets. Next, for each subset, the same method is used according to a hyperparameter called `max depth` which defines when the splitting process will stop, thus avoiding overfitting issues.

Random Forest: Random Forest is an ensemble ML-supervised method, combining multiple ML models using the bagging method [154]. This means that different training subsets are generated randomly from the initial training dataset with replacement. It is worth mentioning that there is a high possibility that each data subset will not include unique data samples. Next, the various ML models are trained individually based on each subset. This step is known as row sampling or bootstrap. Then, the final outcome relies on the majority voting of the various ML models. This process is known as aggregation. The ML models usually refer to decision trees that were described earlier.

AdaBoost: AdaBoost [162] is another ML-supervised ensemble method which adopts the boosting method. This means that the prediction efficacy is improved by converting a number of weak learners into strong learners. In general, the rationale behind the boosting methods is that a first ML/DL model is generated based on an existing dataset; next, the second model intends to correct the error of the first model. This process is repeated until the error is minimised. AdaBoost usually adopts decision trees with one level. These trees are also known as decision stumps. First, an equal weight value is assigned to all the data points of the dataset. Therefore based on N data points, $w(x_i, y_i) = \frac{1}{N}, i = 1, 2, ...n$. Next, for each feature, a decision stump is calculated by calculating the Gini Index (another criterion like entropy). The decision stump with the lowest Gini Index is selected. Next, the "Importance" or "Influence" of this classifier is calculated by $a = \frac{1}{2}log\frac{}{1-TotalError}TotalError$. $TotalError$ indicates the sum of the misclassified data points. The "*Importance*" or "*Influence*" of the classifier is also known as the "*Amount of Say*". After computing the "*Amount of Say*", the weight values will be updated for each data point according to $w = w * e^{\pm a}$. The positive a is used when the data point was classified correctly, while the negative a is used when the data was misclassified. Consequently, the mistaken classifications will lead to higher weight values, while the correct classifications will result in lower weights. Next, the data points with the higher weight values are chosen repetitively to compose the updated dataset that will be used for the new training process.

Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) LDA is a linear ML-supervised method used for dimensionality reduction and classification problems [188]. In particular, the goal of LDA is to transform the data from a dimensional space D to a dimensional space D' where $D > D'$, thus maximising the variability between the classes and minimising the variability within the classes. Next, the transformed data can be used to construct a discriminant, taking full advantage of Bayes' theorem. Next, a linear score $\delta_k(x) = x^T \sum^{-1} m_k^T \sum^{-1} m_k + logpi_k$ [21] is computed, thus classifying a data point to the class with the highest score. In order to use LDA with non-linear problems, kernel functions can be used. The goal is to project the input data into a new high-dimensional space where the inner products can be calculated by the kernel functions. It is noteworthy that LDA relies on the assumption that the various data follow the Gaussian distribution while the classes are characterised by specific means and equal variance/covariance. On the other side, QDA [187] is an extension of LDA without using the assumption that the classes should have an equal covariance. In other words, the covariance matrices can differ for each class. Consequently, LDA is appropriate for small datasets, while QDA can be used for large datasets.

Principal Component Analysis (PCA): PCA is an unsupervised ML method which is commonly adopted for dimensionality reduction tasks [157]. However, it can also be used for detecting outliers. In general, the goal of PCA is to reduce the dimensionality of a d-dimensional dataset by projecting it into a k-dimensional subspace where $k < d$, retaining in parallel most of the information. To this end, the first step is to standardise the initial dataset. Next, the eigenvectors and eigenvalues are received from the covariance matrix or the correlation matrix. Otherwise, singular vector decomposition can also be used. The next step is to sort the eigenvalues in descending order and choose the k eigenvectors that correspond to the k largest eigenvalue, where k is the number of dimensions of the new feature subspace. Next, the projection matrix W is constructed from the selected k eigenvectors. Finally, the k-dimensional feature subspace Y is generated by appropriately transforming the original dataset X through W. It is worth mentioning that PCA does not remove some features from the initial dataset X but creates some new different features called principal components. Regarding the outlier detection process through PCA, the goal is to reconstruct the original data X from the principal components. The reconstructed data points/instances will not be the same as the original ones; however, they should be similar. By comparing the original data points/instances and the reconstructed ones, the reconstruction error is calculated. Consequently, based on the reconstruction error and a relevant threshold value, the outliers can be detected. Alternatively, the outlier scores can be computed based on the sum of the projected distance of the original data and the eigenvectors.

Isolation Forest: Isolation Forest is an ensemble unsupervised ML method which is capable of recognising anomalies/outliers based on the assumption that the anomalies/outliers are few and different [60]. In particular, Isolation Forest utilises binary decision trees that, in turn, handle a subset of data samples according to randomly selected features. Such trees are usually named Isolation Trees (iTrees). The data samples travelling deeper into the iTrees have a low probability of being anomalous. On the other hand, the data samples that end up shortly based on the structure of the iTree denote potential anomalies/outliers. First, some randomly selected samples are assigned to the iTree. Based on a random feature and the relevant threshold, the structure of the iTree is formed by creating left and right nodes. In particular, if the value of this feature is smaller than the threshold, then this data point is placed in the left node. Otherwise, the data point is placed in the right node. This process is conducted recursively until every data point is completely classified or differently a maximum depth is defined and reached. Consequently, a set of iTrees is created, and the training procedure of Isolation Forest is completed. During the inference mode, the new data point is passed through each of the previously-trained iTrees. Next, an anomaly score is calculated in an aggregate manner based on the depth achieved in each iTree. Practically, according to the *contamination parameter* given during the training procedure, an anomaly score of -1 indicates the potential anomalies/outliers, while 1 denotes the normal points.

Local Outlier Factor (LOF): LOF is an unsupervised ML method which calculates the local density of a data point/instance based on its neighbours [9]. The outliers/anomalies are characterised by a lower density than the relevant neighbours. In particular, LOF is defined as the ratio of the mean Local Reachability Distance (LRD) of the K neighbours of a data point/instance A to the LRD of A, that is $LOF_k(A) = \frac{\sum_{X_{j \in N_k(A)}} LRD_k(X_j)}{|N_K(A)|} \times \frac{1}{LRD_k(A)}$ where $N_k(A)$ denotes the K-neighbours of A. Moreover, $LRD_k(A)$ is defined as $LRD_k(A) = \frac{1}{\sum_{X_{j \in N_k(A)}} \frac{RD(A,X_j)}{||N_k(A)||}}$ where RD implies the Reachability Distance (RD) which is the maximum k-distance of X_j and the distance between X_j and X_i. Consequently, if a data point/instance is not an outlier/anomaly, LOF is approximately equal to the LRD of this data point/instance. On the other side, LOF will be greater.

Minimum Covariance Determinant (MCD): MCD [68] is an unsupervised ML method used to estimate the multivariate location and scatter with a high degree of confidence. Although it was introduced in 1984, it was widely adopted when Rousseeuw and Van Driessen introduced the FAST-MCD method. Due to its resistance to outlier observations, MCD is also highly useful for detecting outliers. In particular, the goal of MCD is to find the h observations with the lowest determinant in the covariance matrix. Thus, the MCD estimates of location and scatter are calculated by using the average and the covariance matrix of these h data points. Thus, given a data space S with n data points and p features, $MCD = (X_j^*, S_j^*)$ where $X_J^* = \frac{1}{h} \sum_{i \in J} x_i$ and $S_J = \frac{1}{h} \sum_{i \in J} (x_i - X_j^*)(x_i - X_j^*)^t$. Finally, J denotes a set of h points so that $|S_J^*| \leq |S_K^*| \forall$ sets K s.t. $\#|K| = h$ where $\#|\omega|$ denotes the number of elements in set ω. h can be considered as the minimum number of data points that will not be outliers. Given a particular cluster, the points that are outliers are not involved in the calculations related to the location and scatter.

Angle-Based Outlier Detection (ABOD): ABOD [91] is a geometric unsupervised ML method used for detecting outliers given a multivariate feature space. Unlike distance-based methods, ABOD can work well with a high-dimensional feature space. In particular, ABOD relies on the angle formed by the various features. The variance of the angle seems to vary with respect to the normal data points/instances and the outliers/anomalies. In particular, the variance for the normal data points/instances is greater compared to the outliers/anomalies. First, the angle of each data point/instance is formed and stored in a relevant list. Next, the variance of this list is calculated. Finally, the variance values that are smaller than a specific threshold indicate a potential outlier/anomaly.

Deep Neural Networks (DNNs) DNNs refer to DL methods that can be supervised, unsupervised or semi-supervised [131]. In this thesis, custom Multi-Layer Perceptron (MLP) [184], CNN [84], autoencoders [219] and Generative Adversarial Networks (GANs) [48] are utilised. More information about these DNNs is given in Chapter 5. In general, regarding classification and intrusion/anomaly detection problems, a DNN is composed of several neurons that can cooperate with each other in order to predict the potnetial class. Regarding MLP, also known as Artificial Neural Networks (ANNs), first, the various weights are initialised. Next, for each data point/instance in the dataset, the relevant neuron is created in the input layer. Subsequently, the neurons are activated, taking full advantage of an activation function like the sigmoid function. The outcomes of the neurons are propagated from left to right (forward-propagation), thus predicting the class y. Next, y is compared to the actual y', thus measuring the relevant error. Various loss functions can be used for this purpose, such as for example the MSE. Next, the error is back-propagated to the neurons, updating the relevant weights. The goal is to minimise the value of the relevant cost function. Two popular methods for this purpose are gradient descent and stochastic gradient descent. It is noteworthy that the previous steps are repeated for each data point/instance of the dataset. When this process is completed for all the data points/instances of the datasets, this leads to an epoch. On the other side, CNNs are mainly utilised for classifying images. Finally, Autoencoders and GANs are composed of two networks, namely (a) encoder and (b) decoder - (a) generator and (b) discriminator, respectively, that are mainly used to generate new data points. However, they can also be utilised for detecting anomalies/outliers. More information about CNNs is given in [84].

3.4 Honeypots

Honeypots are non-valuable assets that intend to mimic the behaviour of the actual assets, thus safeguarding them and gathering important information about the malicious activities of the cyberattackers [121]. It is worth mentioning that a honeypot is considered as a security hole, including intentionally security weaknesses and vulnerabilities in order to trap the potential cyberattackers. Fig. 3.3 illustrates how a honeypot can be categorised based on four main criteria: (a) Objectives and User Requirements, (b) Level of Interaction, (c) Physicality and (d) Operation Type. First, according to the goal and the user requirements, a honeypot may be divided into two types: (a) production honeypots and (b) research honeypots. The production honeypots are deployed in the production network in an attempt to conceal the actual assets from malicious insiders. On the other hand, research honeypots are accessible to public networks such as the Internet, enticing potential cyberattackers and gathering valuable information about their behaviour. It is worth noting that any engagement with a honeypot is seen as suspect because legitimate users have no incentive to communicate with it. Furthermore, honeypots may be classed as (a) Low-Interaction Honeypots (LIH), (b) Medium-Interaction Honeypots (MIH), and (c) High-Interaction Honeypots (HIH). LIH can imitate some network services in terms of communication protocols without totally simulating the network behaviour of the real assets. MIH can better mimic the network behaviour of the actual assets by broadcasting network packets as the genuine asset. Finally, HIH is a full replica of the genuine asset, including all hardware and software features. Next, a honeypot can be categorised as physical or virtual depending on how it was implemented and used. For instance, a physical honeypot can be an unused hardware entity/device, while a virtual honeypot can rely on software specially designed to play the role of a honeypot. Usually, the physical honeypots are characterised as MIH or HIH, while the virtual ones are LIH or HIH. Finally, depending on the operation type, an Information Technology (IT) asset may be characterised as a server or client. Thus, accordingly, a honeypot can play the relevant role (i.e., server or client) or can support both of them. In the last case, the honeypot is characterised as a hybrid. A detailed survey on honeypots is provided by M. Nawrocki et al. in [121].

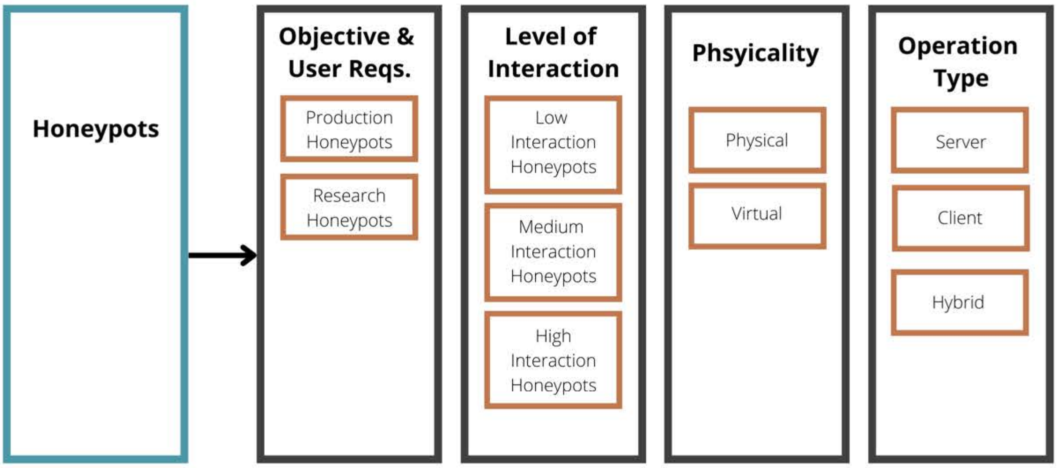


FIGURE 3.3: Honeypot Classification Criteria and Types

3.5 Intrusion Prevention Techniques

After the detection phase, mitigation and prevention actions can follow by the response module. A characteristic example is the activation of some firewall rules responsible for isolating the cyberattacker. Another example is the use of honeypots in order to mislead and trap future malicious activities. Finally, the response module can take full advantage of the SDN [204] technology in order to mitigate the cyberattacks or anomalies in near real-time. In this thesis, both honeypots and SDN are used by the proposed SIEM system in order to mitigate the various cyberattacks. In particular, as illustrated in Fig. 3.4, the SDN architectural design consists of four planes, namely (a) Data Plane, (b) Control Plane, (c) Application Plane and (d) Management Plane. The Data Plane refers to the entities/devices that are connected to hardware or software SDN switches. Next, the Control Plane is characterised by the presence of the SDN Controller (SDN-C), which is responsible for managing the network elements of the Data Plane through the southbound Application Programming Interface (API). To this end, various southbound protocols can be used, such as OpenFlow, NETCONF, OpFlex and the Simple Network

Management Protocol (SNMP). The Application Plane includes applications (such as the response module) that guide and communicate with the SDN-C in order to apply efficient policies with respect to the entities of the data plane. For this purpose, northbound APIs are utilised regarding the communication between the applications and the SDN-C, such as REpresentational State Transfer (REST). Finally, the Management Plane is a cross-layer block responsible for the deployment, configuration and management of the various entities/devices and components of the other planes.

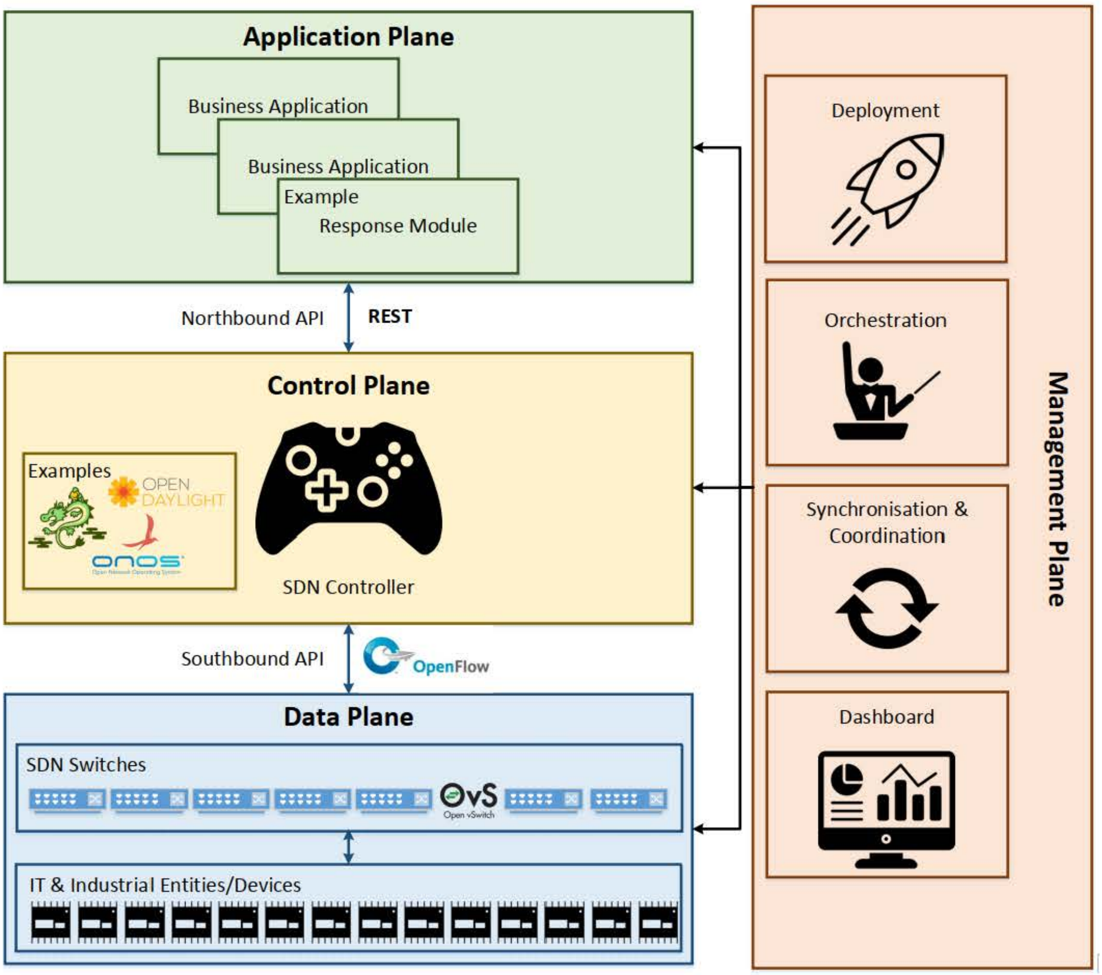


FIGURE 3.4: SDN Architectural Model

3.6 Security Information and Event Management Systems

The SIEM system is in charge of organising and coordinating the monitoring, detection, and mitigation processes of a smart ecosystem like the SG. In particular, a SIEM has the ability to aggregate, normalise, correlate, and visualise a wide range of security events, allowing it to detect and analyse potential security breaches [51]. A security event refers to a message or a security log that has been normalised in a specific format and is related to the security status of the monitored infrastructure. The correlation of security events can result in the generation of security alerts that refer to a group of linked security events. Security and AI association rules (such as the Apriori and Eclat ML methods) can be utilised to accomplish this. As a result, a SIEM acts as an umbrella for multiple security tools, commonly referred to as SIEM sensors, aggregating, normalising, correlating, and visualising their results. Characteristic examples of SIEM sensors are IDPS, firewalls and honeypots. Popular SIEM systems are AT&T Cybersecurity AlienVault Unified Security Management, Splunk Enterprise Security [65], IBM QRadar and McAfee Enterprise Security Manager. On the other side, a SOC [42] consists of a group of security experts that use various security tools (including SIEMs) and appropriately handle the various security events and alerts.

3.7 Chapter Summary

Given the great IoT threat landscape discussed in the previous chapter, it is evident that the presence of IDPS is necessary. This chapter provides a detailed overview of IDPS. In particular, first, the objectives and requirements of IDPS are enumerated and described. Next, according to D. Denning et al. in [41], the typical architectural model of the IDPS is discussed, explaining the role of each component and the various categories, such as HIDPS, NIDPS and Hybrid IDPS. Subsequently, the intrusion detection techniques (i.e., signature/specification-based detection and anomaly-based detection)are summarised, paying special attention to AI methods for intrusion and anomaly detection. Then the role of honeypots is provided, and intrusion prevention techniques are discussed, taking into consideration novel technologies, such as SDN. Finally, the role of SIEM systems is presented, organising the security and detection procedures in an aggregative manner.

Chapter 4

Review of Intrusion Detection and Prevention Systems for Smart Grid

After providing an overview and the requirements of the intrusion detection and prevention techniques, this chapter focuses on investigating and analysing a wide range of IDPS for the SG architectural components. Appendix D summarises this analysis. In particular, the various IDPS studied in this chapter can focus on the entire SG paradigm or the individual elements, such as AMI, SCADA, substations and synchrophasor systems. For each IDPS, the architectural design and implementation details are provided. Based on the detection category (i.e., signature/specification-based detection, anomaly-based detection), a special emphasis is given to the signature/specification rules and the statistical analysis/AI models, respectively. Moreover, technical details about the industrial protocols and the various cyberattacks and anomalies are provided, while the evaluation results for each IDPS are discussed. Finally, this chapter discusses honeypot-related works and SDN-enabled IDPS. According to this analysis, the strengths and limitations of the existing works are discussed, thus guiding the design and implementation of the proposed SDN-enabled SIEM, discussed in the following chapter.

4.1 Signature/Specification-based IDPS

This section focuses on signature & specification-based IDPS for the SG. In particular, 20 IDPS of this category are discussed, paying special attention to industrial protocols used in SCADA/ICS, substations and synchrophasor systems, such as Modbus/TCP, DNP3, IEC 60870-5-104, IEC 61850 and IEEE C37.118. Each IDPS is described in detail in a separate paragraph.

In [120], T.H. Morris et al. provide a set of signature rules for the Modbus protocol. In particular, Modbus is an industrial protocol following the master-slave paradigm. It was initially designed by Gould Modicon (now Schneider Electric) in 1979 for the SCADA communications between MTU (master) and logic controllers (slave). However, Modbus is now widely used in IIoT applications. The authors define 50 signature rules for both Modbus/RTU and Modbus/TCP based on their specifications. Snort is used to test the various rules.

In [106], H. Li et al. pay special attention to the DNP3 protocol, providing a set of Snort signature rules. DNP3 is another industrial protocol which was initially designed for the SCADA communications. However, it is now used in multiple IIoT applications. It was standardised by IEC TC-57 and deployed by IEEE Electric Power Engineering Association. Similarly to Modbus, DNP3 is characterised by severe security issues, thus allowing the cyberattackers to violate the DNP3 communications. In this work, the authors provide an intrusion detection template. Next, it is used to define the various signature rules. It is worth mentioning that this template can also be utilised for other industrial protocols, such as Profinet, Modbus and IEC 60870-5-104.

A specification-based IDS for AMI was also created in [24]. The proposed IDS focuses on American National Standards Institute (ANSI) C12.22 communications, while its architecture is composed of four main components. The first component, named dissector, is responsible for capturing the network traffic data. Next, the parser undertakes to parse and analyse the network traffic patterns. Then, the third component applies the various specifications defined by the normal characteristics of AMI. Finally, the last component undertakes to monitor the status of each entity. The security specifications are defined based on a threat model combining (a) meter reading attacks and (b) service switch attacks. In particular, the various specifications are categorised into three main classes: (a) device-based, (b) network-based and (c) application-based. According to the experimental results, the authors use Table TstBench and various virtual machines in order to emulate ANSI C12.22 and the AMI components, respectively. True Positive Rate (TPR) and True Negative Rate (TNR) reach 100% while 0.3% of CPU and 10MB of Random Access Memory (RAM) are used.

In [112], X. Liu et al. provide a specification-based IDS, which focuses on the smart meter communications. First, based on a Coloured Petri Net (CPN), the authors introduce an information model about the modules composing a smart meter. On this basis, a threat model is also defined, including two main attack classes: (a) data attacks and (b) command attacks. Finally, an IDS targetting false data injection attacks is proposed. The architectural design of the proposed IDS includes three modules: (a) Secret Information, (b) Event Log and (c) Spying Domain. The first module is a private data structure that can be accessed through legitimate actions. It is also used to encrypt the Event Log, which is used to store the activities related to the smart meters. Finally, the spying domain is composed of random storage areas, comprising the hash code of the Secret Information. A security alert is generated when a malicious user tries to access the storage units of the Event Log and the Spying Domain, respectively. The effectiveness of the proposed IDS is demonstrated by evaluation diagrams showing the TPR.

In [119], the authors introduce a specification-based IDS that consists of separate IDS modules for the AMI architectural components, including smart meters, data collectors and AMI headends. A set of normal behaviour rules is specified for each of the above components. The IDS unit focusing on the AMI headend has the ability to monitor and control the data collectors. In a similar manner, the IDS unit responsible for monitoring the data collector can also monitor the various smart meters. In general, the proposed IDS can address two attack kinds: (a) reckless attacks and (b) random attacks. According to the evaluation results, TPR reaches 100%. On the other hand, False Positive Rate (FPR) is less than 0.2% and 6%, respectively.

In [77], P. Jokar and V. Leung provide an IDPS for ZigBee-based HANs. In general, the proposed IDPS relies on security specifications using the characteristics of the physical and data link layers, while its architecture is composed of several IDPS units and a centralised one that are responsible for monitoring the security status of the various HAN and further analysing the security events, respectively. The security specifications are defined based on six main features: (a) datagram of IEEE 802.15.4 and Smart Energy Profile 2.0 (SEP 2.0), (b) Received Signal Strength (RSS), (c) traffic rate, (d) sequence number, (e) node availability and (f) Packet Error Rate (PER). When a security event is generated, an appropriate prevention action is also activated through an RL agent, which relies on Q-Learning. According to the evaluation results, the authors first demonstrate the detection efficiency against six attacks: (a) radio jamming attacks, (b) stenography attacks, (c) replay attacks, (d) back-off manipulation attacks, (e) DoS against Guaranteed Time Slot (GTS) requests and (f) DoS against data transmission during the Contention Free Period (CFP). The Receiver Operating Characteristic (ROC) diagrams show detection effectiveness.

A specification-based IDS for AMI was created by M. Attia et al. in [18]. In general, the proposed IDS relies on temporal and spatial detection methods, focusing on blackhole and time delay attacks. While a blackhole is a DoS attack discussed earlier, the goal of the time delay attack is to introduce an extra time when the various network packets are transmitted. The security specifications are defined based on the number of the various packets transmitted and the delay time between them. To this end, the mean value and the standard deviation of the Gaussian distribution are computed. The authors demonstrate the efficiency of the proposed method by comparing it with three other methods, namely: (a) spatial-based method, (b) temporal-based method and (c) SVM classifier. While the SVM classifier achieves the best TPR, the proposed IDS achieves the best FPR.

In [212], Y. Yang et al. provide a specification-based IDS for IEC 60870-5-104 ICS/SCADA systems. The functionality of their IDS is based on a Detection State Machine (DSM), which in general relies on the Finite State Machines (FSM). In particular, the IEC 60870-5-104 commands are specified via the correlations of FSM. In contrast to the traditional FSM-based systems, the proposed solution applies a set of alarms that are capable of distinguishing the protocol malfunctions. To evaluate their work, the authors use the Internet Traffic and Content Analysis (ITACA) software. Based on the evaluation results, TPR and FPR reach 100% and 0%, respectively.

In [210], Y. Yang et al. provide signature and specification rules for the IEC 60870-5-104 SCADA/ICS systems, using the syntax of Snort. After investigating the security issues of IEC 60870-5-104, the authors highlight attack signatures and specification rules for the following attacks: (a) unauthorised read commands, (b) unauthorised reset commands, (c) unauthorised remote control and adjustment commands, (d) spontaneous packets storm, e) unauthorised interrogation commands, (f) buffer overflows, (g) unauthorised broadcast requests and (h) IEC 60870-5-104 port scanning. As expected, according to the evaluation results, there are no false alarms based on the above signature and specification rules.

In [45], Z.Feng et al. focus their attention on the security of Profinet. Profinet is another industrial protocol used in SCADA/ICS environments. It was implemented by Profibus & Profinet International and standardised by IEC 61158 and IEC 61784. In particular, the authors provide a set of signature and specification rules, using the syntax of Profinet. Like the other industrial protocols, Profinet suffers from severe security issues since it does not include authentication and authorisation mechanisms, thus allowing MITM attacks. In this paper, the authors enhance Snort by decoding the Profinet attributes and providing appropriate signatures and specification rules for detecting MITM, DoS and reconnaissance attacks. According to the evaluation analysis, the traffic traces of D. Zhang et al. in [218] were used, while also DoS scenarios were emulated. The various signature and specification rules can successfully detect the Profinet-related attacks.

In [79], B. Kang et al. provide an IDS for substation environments using IEC 61850. In particular, the authors focus on the MMS standard of IEC 61850. The proposed IDS rely on signature rules, paying special attention to active power limitation attacks. The authors implement a stateful analysis plugin which can be incorporated into Suricata. This plugin includes three main function units: (a) the application layer protocol decoder, (b) the rule match engine and (c) the state manager. The first unit decodes the application layer packets and extracts their attributes. The second unit adopts content and state inspection rules in order to identify particular attack patterns. The content inspection rules investigate particular conditions for each MMS packet, while the state inspection rules check the presence of specific flags that should characterise the protected entities. Finally, the state manager updates the status of the protected entities. Based on the evaluation result, the proposed IDS can effectively recognise active power limitation attacks.

In [99], the authors provide a specification-based IDS for a substation environment in South Korea. Their IDS focuses on various protocols, such as the GOOSE and MMS protocols defined in IEC 61850, SNMP, Network Time Protocol (NTP) and ARP. In particular, the proposed IDS investigates general network traffic characteristics, such as the number of bits per second (bps), the number of packets per second (pps) and the number of connections per second (cps). Based on the above features, specification rules were defined, using statistical analysis techniques. Regarding the evaluation procedure, a real dataset from their environment was used, including network traces from various attacks, such as DoS attacks, port scanning attacks, GOOSE-related security violations, MMS-related security violations, SNMP-related attacks, ARP attacks and NTP attacks. Based on the evaluation results, $Precision = 100\%$, $FPR = 0\%$, $FNR = 1.1\%$ and $TPR = 98.9\%$.

In [213], Y. Yang et al. introduce a specification-based IDS, protecting substation environments using the IEC 61850 protocols: MMS, GOOSE and Sampled Measure Value (SMV). In particular, the architecture of the proposed IDPS consists of five modules: (a) configuration module, (b) network traffic capture module, (c) process core module, (d) rule module and (e) result module. The first one is responsible for investigating the characteristics of the substation, thus defining specific threshold values. The second module undertakes to capture and isolate the network traffic data of MMS, GOOSE and SMV. Next, the process core module relies on ITACA in order to decode and process the attributes of the above protocols. Then, the rule module applies the specification rules. Finally, the results module notifies the security administrator about potential security violations. In this paper, the specification rules can be classified into four categories: (a) access-control detection, (b) protocol whitelisting detection, (c) model-based detection and (d) multi-parameter detection. The category defines the legitimate MAC and Internet Protocol (IP) addresses. It also defines the normal TCP ports, thereby forming a whitelist. The specification rules of the second category intend to detect malicious packets that are not related to IEC 61850. Next, specification rules related to GOOSE, MMS and SMV are defined. Finally, the last category refers to specification rules focusing on the physical characteristics of the substation environment. Regarding the evaluation process, a dataset from an actual substation in China was utilised. According to the authors, the proposed IDS can detect various cyberattacks, including MITM, DoS and packet injection attacks.

In [78], M. Kabir-Querrec et al. provide a specification-based IDS which focuses on substation environments using GOOSE (defined in IEC 61850). In particular, the architecture of the proposed IDS relies on the data object model defined in IEC 61850, introducing a new intrusion detection function. This data object model is composed of many Logical Nodes (LNs) that define logical functions and can communicate with each other through the Piece of Information for COMmunication (PICOM) protocol. Despite the fact that IEC 61850 incorporates a security function called named Generic Security Application (GSAL), the authors provide a new function which defines and checks the specification rules. In general, to define a new function within IEC 61850, the following steps have to be accomplished: (a) a formal description of the function is needed, b) the function has to be decomposed into LNs and (c) the interaction with the other functions has to be determined. Therefore, the authors created a new LN called CYSN, which is responsible for capturing the GOOSE messages and sending them to two LNs that undertake to check the specification rules. In particular, the first LN called CYComChkSingle undertakes to verify the structure and parameters of each message. On the other hand, the second LN called CYComChkMany verifies the consistency of the messages based on a specific time slot.

In [132], U. Premaratne et al. provide a hybrid IDS for IEC 61850 substation environments. The proposed IDS combines signature and specification rules focusing on traffic analysis attacks, DoS attacks and password cracking. The authors emulated the previous cyberattacks, in order to identify the appropriate patterns and design the corresponding signature and specification rules, using the syntax of Snort. In particular, to execute these attacks, they used the ping command, THC Hydra and Seringe.

Nevertheless, although the authors claim that their IDS is devoted to IEC 61850 substation environments, it is worth mentioning that it cannot counter cyberattacks against the IEC 61850 protocols, such as GOOSE and MMS.

J. Hong et al. in [63] provide a specification-based IDS which focuses on multicast GOOSE and SMV messages. In particular, the authors describe in detail two specification rules used to detect GOOSE and SMV cyberattacks, respectively. Regarding the GOOSE cyberattacks, their IDS can detect relevant replay attacks, DoS attacks, attacks generating malicious GOOSE data, malicious activities changing the GOOSE control data and finally, actions modifying the time information. On the other hand, regarding the SMV attacks, the proposed IDS can detect relevant DoS attacks and malicious actions that modify or generate SMV data. The architecture of the proposed IDS is composed of four modules: (a) packet filtering module, (b) packet parser module, (c) specification-based IDS module and (d) HMI module. The first module is responsible for capturing only GOOSE and SMV packets. Next, the second module undertakes to extract from the GOOSE and SMV packets the corresponding attributes. Next, the specification-based IDS module defines and checks the specification rules. Finally, HMI informs the system operator or the security administrator about potential cyberattacks and anomalies. The authors evaluate the detection performance of their IDS under real conditions by implementing a Cyber Physical System (CPS) testbed. Based on the experimental results, FPR can reach 1.61×10^{-4}.

In [209], Yi. Yang et al. provide a specification-based IDS, which also focuses on smart substations using IEC 61850. The architecture of the proposed IDS is composed of five modules: (a) configuration module, (b) network traffic capturing module, (c) IDS process core, (d) rule module and (e) result module. The first module refers to the configuration files used to define the specification rules. The second module is responsible for capturing the IEC 61850 packets. Next, the IEC 61850 packets are processed, extracting their attributes. Then, the fourth module undertakes to compare the characteristics of the IEC 61850 packets with a predefined set of specification rules. Finally, the last module notifies the security administrator about the presence of potential attacks or anomalies. The specification rules are categorised into four classes: (a) access control detection rules, (b) protocol-based detection rules, (c) anomaly behaviour detection rules and (d) multi-parameter detection rules. The first category is responsible for allowing only the network traffic coming from legitimate MAC and IP addresses. The following specification rules allow only the packets of the IEC 61850 protocols (GOOSE, MMS, SMV). Next, based on the attributes of the IEC 61850 protocols, appropriate rules are defined, identifying the normal behaviour of the substation environment. Finally, the last category refers to specification rules related to physical attributes.

S. Pan et al. in [128] present a hybrid IDS for the synchrophasor systems, combining signature-based and specification-based rules. In particular, their work relies on the common-path mining approach and Snort. They investigate an architecture of three bus two-line transmission system consisting of a real-time digital simulator, four relays, four PMUs, a PDC, an energy management system using OpenPDC and a personal computer that executes Snort. The input data from the previous devices are compared with common paths. A common path is a sequence of system states that may be a specification of the normal behaviour or a signature of a cyberattack. Based on these characteristics, the particular IDS can classify an activity as (a) system disturbance, (b) normal operation and (c) cyberattack. The training process of the common-path mining algorithm includes the creation of a dataset which includes 25 scenarios of 10000 simulation instances. These scenarios are classified into three categories, namely (a) single-line to-ground faults, (b) normal operations and (c) cyberattacks. According to the evaluation results, the accuracy of the proposed IDPS is calculated at 90.4%.

In [85], R. Khan et al. introduce a hybrid IDS, which is mainly based on specification-based and signature-based rules for synchrophasor systems using the IEEE C37.118 protocol. The architecture of the proposed system consists of separate Host Intrusion Detection Systems (HIDS) and Network Intrusion Detection Systems (NIDS) called agents and sensors, respectively. The agents monitor the operation of PMUs or PDCs, while the sensors monitor the overall network traffic. Also, there is a management server, which aggregates and correlates all information coming from the individual agents or sensors. In addition, a database server is responsible for recording any detection alert or warning. The agents and sensors consist of six components: (a) Packet Capture (PCAP) filters, (b) IEEE C37.118 decoder, (c) analyzer/detector, (d) state manager, (e) events manager and (f) console. The PCAP filters are responsible for capturing the IEEE C37.118 packets. The IEEE C37.118 decoder analyses the previous packets and extracts the appropriate information. The analyser/detector adopts a set of rules in order to detect abnormal behaviours. This set is composed of four category rules: (a) signature-based rules, (b) range-based rules, (c) threshold-based rules and (d) stateful behaviour-based rules. According to the authors, the specific set of rules is able to detect a plethora of cyberattacks, such as ARP poisoning attacks, replay attacks, port scanning attacks, DoS attacks, GPS spoofing attacks, command injection attacks and physical attacks. Subsequently, the analyser/detector communicates with the state manager, which stores possible alerts or warnings in the database server. Next, the event manager communicates with the management server, whose operation was discussed previously. Finally, the console is a command line or a GUI environment with which the user can configure the operations of the previous components, such as the detection rules.

In [211], Y. Yang et al. provide a specification-based IDPS capable of protecting synchrophasor systems using also the IEEE C37.118 protocol. More specifically, their IDPS consists of three kinds of rules, including (a) access control rules, (b) protocol-based rules and (c) behaviour-based rules. The access control rules define a whitelist with the legitimate source and destination MAC and IP addresses as well as the corresponding ports at the transport layer. Next, the protocol-based rules adopt also a whitelist which in turn defines the application layer protocols allowed for the interaction between the synchrophasor components. In this case, this list will enable only the IEEE C37.118 traffic. Finally, the last category identifies behaviour rules based on the attributes of the IEEE C37.118 packets by utilising a deep packet inspection process. All the rules are described sufficiently in the paper. Regarding the evaluation process, the authors tested their IDPS in a real testbed by executing reconnaissance, MITM and DoS cyberattacks. According to the experimental results, the FPR of the proposed IDPS is calculated at 0

4.2 Anomaly-based IDPS

This section focuses on anomaly-based IDPS. In particular, 16 IDPS of this category are investigated and analysed. For each work, technical details about the statistical analysis and AI methods and models are provided. Moreover, the dataset utilised in each work is discussed. Similarly, each work is detailed in a separate paragraph.

An IDS for the entire SG ecosystem is proposed by A. Patel et al. in [129]. The functionality of this IDS relies on three main components: (a) an Ontology Knowledge Base (OKB), (b) an SVM classifier and (c) a fuzzy risk analyser. The detection of the potential attacks relies on SVM trained for more than 30 hours. For this purpose, the KDD dataset [2] was utilised, including also some cyberattack traces from the authors. The dataset includes a wide range of attack traces, such as duplicate insertion, payload mutation, brute force attacks, shellcode mutation, command insertion, packet splitting and DoS. Through the fuzzy analyser, the risk value of each SG entity can be calculated. Finally, OKB is used to identify the threat actors. With respect to the evaluation of the proposed IDS, the Area Under Curve (AUC) reaches 0.99451.

In [222], Y. Zhang et al. present SGDIDS, a decentralised IDS for the entire SG ecosystem. SGDIDS relies on Artificial Immune System (AIS) methods, including different IDS modules for each HAN, NAN and WAN. In particular, each IDS module uses the AIS methods [208], taking full advantage of the hierarchical nature of HAN, NAN and WAN. This means that a NAN-related IDS takes into consideration the detection outcomes of the relevant HAN IDS. In a similar, the WAN IDS considers the security events and alerts originating from the NAN IDS. The detection capability of each IDS is based on the CLONALG and AIRS2Parallel methods. For the training procedure, WEKA [58] and the NSL KDD [155] dataset are used. NSL-KDD includes various attacks, such as User to Root (U2R) attacks, Remote to Local (R2L) and DoS. The detection accuracy of CLONALG and AIRS2Parallel reach 99.7% and 98.7%, respectively.

In [44], M. Faisal et al. introduce a new IDS for AMI, evaluating a wide variety of ML methods through the Massive Online Analysis (MOA) tool. The proposed IDS is composed of three IDS, each of which is in charge of monitoring smart meters, data collectors and AMI headends, respectively. In particular, each IDS consists of the following components: (a) the data acceptor module, (b) the pre-processing unit, (c) the stream mining module and (d) the decision-making unit. Both KDD CUP 1999 [2] and NSL-KDD [155] were utilised for the training and evaluation procedures. These datasets include various attack traces, such as DoS, U2R and R2L. The authors investigate and evaluate with each other several ML methods in terms of the detection accuracy, the size of the classifier in Kilobytes, the classifier's processing time, FPR, the False Negative Rate (FNR) and the usage rate of the Random Access Memory. The ML methods under evaluation are: (a) Single Classifier Drift, (b) Bagging using Adaptive-Size Hoeffding Tree, (c) Bagging using ADWIN, (d) Limited Attribute Classifier, (e) Leveraging Bagging, (f) Active Classifier and (g) Accuracy Updated Ensemble. Based on the evaluation results, it seems that the Active Classifier and Single Classifier Drift are appropriate for the IDS monitoring the smart meters. On the other hand, Leveraging Bagging can be used to detect attacks against the data collectors. Finally, the Active Classifier is an efficient solution for detecting attacks against the AMI headend.

In [199] R. Vijayanand et al. describe an anomaly-based intrusion detection system (IDS) that monitors the AMI communications. The proposed IDS is incorporated into the data collector and relies on a Multi-SVM classifier. Multi-SVM is composed of several SVM that can recognise a wide range of cyberattacks. For this purpose, the ADFA-LD dataset is used, while the mutual information technique is used to identify the most useful features. In particular, the following features from the ADFA-LD dataset are used: (a) Source bytes, (b) Destination time to leave (ttl), (c) Source mean, (d) Destination mean and (e) Ct_state_ttl. According to the authors, the proposed IDS can detect the following attacks: (a) exploits, (b) DoS attacks, (c) fuzzers, (d) backdoors, (e) worms and (f) generic attacks. For each attack, an SVM classifier was generated, utilising a different kernel function. For instance, regarding the DoS and backdoors, the polynomial kernel was used. On the other side, the Gaussian kernel was used for the normal instances and the generic attacks. Matlab was utilised for the training procedure. According to the experimental results, the accuracy of the proposed IDS reaches 0.9, while TPR and TNR reach 89.2% and 93.4%, respectively.

The authors in [109] present an IDS for AMI, taking advantage of the Online Sequence Extreme Learning Machine (OS-ELM), which is a unique feed-forward neural network model based on online sequence learning. The authors follow a methodology consisting of three phases: (a) data pre-processing phase, (b) initialisation phase and (c) online sequence learning phase. During the first phase, the data is processed appropriately based on the Gain Ratio Evaluation feature selection method. Next, the various parameters for the training phase are initialised. Finally, the training process follows. Regarding the training procedure, the dataset of the CER Metering Project [109] was utilised. However, it is worth mentioning that this dataset does not include network traffic characteristics in order to recognise network-related attacks properly. According to the evaluation results, the accuracy of the proposed IDS reaches 97.23% while the FPR and FNR reach 5.897 and 3.614, respectively.

In [34], P-Y. Chen et al. present an anomaly-based IDS against false data injection attacks. In particular, the proposed IDS relies on the spatiotemporal evaluation, regulating the logical connections between the AMI state estimations. The functionality of this IDS consists of two main phases. During the first phase, a set of state estimations is defined based on temporal consistencies and spatial correlations. Next, a voting process follows, discriminating each state estimation as (a) good, (b) unknown or (c) anomaly. According to the experimental results, two false data injection attackers were modelled and emulated. The goal of the first attack was to increase the transmission cost, while the second attack aims to result in a power outage. The FPR is calculated at 0.43.

In [12], the authors show a flow-based IDS for AMI, taking full advantage of clustering techniques. In particular, the proposed IDS consists of several IDS units that monitor the data collectors and the AMI headends. First, the network flow data between the data collectors and the various smart meters are monitored, thus detecting relevant intrusions and anomalies. Next, in a similar manner, the network flow data between the data collectors and the AMI headends are analysed. In general, the detection process relies on mini-batch k-means with a sliding window. Regarding the training procedure, a custom dataset was created based on features related to the TCP/IP network flows. Furthermore, PCA was used in order to identify the most informative points, thus reducing the dimensionality of the dataset. According to the experimental results, three attack scenarios were performed: (a) TCP SYN attack, (b) port scanning and (c) a combination of the previous ones. The best detection performance in terms of FPR and the silhouette score is achieved when $k = 4$.

In [26], N. Boumkheld et al. provide an anomaly-based IDS which can successfully detect blackhole attacks against AMI NAN. In particular, a blackhole refers to a kind of DoS attack, corrupting all the legitimate packets. Based on the NS2, an AMI network was constructed, including two malicious nodes, one data collector and 100 smart meters. The communication of the various entities relies mainly on the AODV protocol. The proposed IDS is considered a separate node, which communicates only with the data collector. The detection procedure relies on the Naive Bayes classifier, which was implemented with WEKA. Three main features are used: (a) the number of the route requests packets, (b) the number of the dropped packets and (c) the number of the route reply packets. According to the evaluation results, the accuracy of the proposed IDS reaches 99%, while TPR, Precision and AUC reach 100%, 66% and 100%, respectively.

In [196], the authors introduce an anomaly-based IDS, which focuses on protecting the AMI. The architecture of the proposed IDS consists of various IDS modules that are distributed in the various HANs, NANs and WANs. If a cyberattack is detected by an IDS, the corresponding security event is generated. Moreover, a centralised IDS module undertakes to gather and further analyse the security events raised by the various IDS modules. The detection process uses the ISCX2012 dataset, including and evaluating various ML methods, such as J48, JRip, BayesNet, SVM and MLP. This dataset includes various cyberattacks, such as Lan-to-Lan (L2L) attacks, botnets, DoS and Secure Shell (SSH) bruteforce attacks. According to the evaluation results, the best detection performance is achieved by j48, where $Precision = 99.70\%$, while $TPR = 99.60\%$.

In [53], V. Gulisano and M. Almgren provide a two-tier IDS that monitors and controls the network activities of AMI. The goal of the proposed IDS is to detect potential malicious activities with respect to the network traffic data exchanged between the smart meters and the data collectors. In particular, the data processing and detection procedures rely on a data streaming technique which analyses the communication traffic through acyclic directed graphs. The architecture of the proposed IDS relies on two main components, namely (a) Device Modeller and (b) Pattern Matcher. The first component is responsible for monitoring the network traffic data and detecting potential intrusions based on a Bayesian network. To this end, three main features are used: (a) the number of requests, (b) the identifier of the smart meters and (c) time information. Next, the second component receives the various security events and aims to correlate and identify potential alert patterns with the help of a cybersecurity analyst. Based on the evaluation results, energy exfiltration attack scenarios were emulated while $TRP = 91\%$.

In [62], E. Hodo et al. present an anomaly-based IDS for a SCADA system which utilises the IEC 60870-5-104 protocol. In 1995, IEC released IEC-60870-5-101, which includes essential telecontrol messages between a logic controller and a controlling server. Six years later, IEC 60870-5-104 was released, combining the application messages of IEC-101 with TCP/IP. However, IEC 60870-5-104 is characterised by several security issues since it does not include any authentication and authorisation mechanisms. The authors create their own dataset, which includes passive ARP poisoning attacks, DoS attacks and replay attacks that replace the legitimate packets with malicious ones. Based on this dataset and WEKA, they evaluated multiple ML algorithms, such as Naive Bayes, J48, Random Forest, OneR, RandomTree and DecisionTable. According to the evaluation analysis, J48 and DecisionTable achieved the best accuracy score.

In [50], N. Goldenberg and A. Wool present an anomaly-based IDS which is devoted to the Modbus/TCP communications. The functionality of this IDS relies on a Moore Deterministic Finite Automaton (DFA), which in turn is based on the high periodicity of the Modbus/TCP packets. In particular, the proposed DFA monitors the requests and replies between MTU and each logic controller, thus identifying the normal and anomalous states. In particular, DFA is composed of (a) a set of states, (b) an alphabet (which is a set of symbols), (c) the transition function and (d) the first state. A state denotes how normal the Modbus/TCP traffic is. It can take four values: (a) Normal, (b) Retransmission, (c) Miss and (d) Unknown. The symbols and the transition function define the states for each communication. The symbols are separated into two main classes: (a) known symbols and (b) unknown symbols. The first category includes those symbols that were observed during the learning phase, resulting in a known state (Normal, Retransmission, Miss), while the second category implies those symbols that lead to the Unknown state. To evaluate their work, the authors generated two real datasets using Wireshark, Pcapy and Impacket. Based on the experimental results, their solution does not present any false alarms.

In [14], S. D. Anton et al. provide a comparison of four ML methods for detecting anomalies given a Modbus/TCP intrusion detection dataset. In particular, the dataset of A. Lemay and J. M. Fernandez [104] was used, including three sub-datasets, namely DS1, DS2 and DS3. DS1 consists of 3319 Modbus/TCP packets, including 75 malicious cases. Next, DS2 includes 11166 198143 packets with ten malicious cases. Finally, DS3 includes 365906 packets with 2016 malicious cases. From the above datasets, specific features were used for the training procedure. The various features refer only to the TCP/IP attributes of the Modbus/TCP packets. Four ML methods are evaluated, namely (a) SVM, (b)

Random Forest, (c) KNN and (d) k-means. According to the authors, the accuracy of SVM is equal to 100%, 100% and 99.99% for each of the previous sub-datasets, respectively. Similarly, the detection accuracy of Ransom forest is 100%, 99.99% and 99.99%. Next, the accuracy of KNN is equal to 99.7%, 99.9% and 99.9%. Finally, the accuracy of k-means is equal to 98.1%, 55.62% and 63.36%.

In [201], P.H. Wang et al. implement an anomaly-based IDS, using a clustering method and honeypot data. The proposed IDS focuses on detecting intrusions against Modbus/TCP, using the traces of Conpot. Conpot is a honeypot that can emulate a wide range of industrial protocols. Each request to Conpot is considered a cyberattack. Next, the authors combine a similarity evaluation method of the Modbus/TCP requests with a hierarchical clustering technique to extract Sequential Attack Patterns (SAPs). After this process, the IDS can classify the new Modbus/TCP requests as an existing SAP or unexpected SAP. Finally, the authors provide a visualisation method, illustrating the various SAPs as flow graphs. According to the evaluation results, the proposed IDS can successfully detect reconnaissance and DoS attacks. For each of the previous attacks, TPR is equal to 90% and 95.12%, respectively, while FPR is equal to 0%.

In [108], S.-C. Li et al. implement an anomaly-based IDS for the Modbus/TCP protocol based on classification ML models. In particular, the authors adopt a J48 decision tree and three neural networks implemented with WEKA. To train the above models, they created a dataset by constructing a real testbed consisting of a programmable logic controller, an MTU, a cyberattacker unit, and a cyberdefender unit. This dataset includes (a) DoS attacks, (b) command injection attacks, (c) response injection attacks and (d) reconnaissance attacks. To create their dataset, the authors used Wireshark and a Programming Hypertext Preprocessor (PHP) script responsible for converting the Packet Description Markup Language (PDML) format of Wireshark into Comma-Separated Values (CSV) format. The training process relies on 39 features; however, they are not given in the paper. Based on the evaluation results, the accuracy of j48 is equal to 99.8361%, while the detection accuracy of the neural networks reaches 97.4185%, 97.4603% and 97.3876%, respectively.

In [215], H. Yoo and T. Shon provide an anomaly-based IDS for substations utilising the IEC 61850 standard. In particular, the proposed IDS focuses on MMS and GOOSE, taking full advantage of an one-class SVM classification model. The proposed IDS consists of four main functions: (a) data capturing and preprocessing, (b) outlier processing, c) one-class SVM training and (d) anomaly detection. The first function refers to capturing and preprocessing the MSS and GOOSE packets. Thus, three datasets are generated. The first dataset includes the attributes of the MMS and GOOSE packets. The second dataset refers to the network flows formed by the MMS and GOOSE communications. Finally, the third dataset includes traffic characteristics, such as pps and bps. The second function is responsible for cleaning the training datasets from outliers. For this process, the Expectation Maximisation (EM) and LOF were used. Finally, the next functions refer to the training and testing procedure of the one-class SVM classification model. To this end, data from a real substation was used. Based on the evaluation results, FPR ranges between 1% and 6%.

4.3 SDN-enabled IDPS

This section is devoted to describing SDN-enabled IDPS, i.e., IDPS taking full advantage of the SDN technology in order to mitigate and prevent potential intrusions and anomalies. In particular, five relevant works are analysed, investigating how SDN can be used to mitigate the various cyberattacks in an efficient manner. In a similar manner, each work is analysed in a separate paragraph.

H. Lin et al. in [110] provide Integer Linear Programming (ILP) models and a greedy heuristic algorithm in order to increase the observability of a power system operation network. In particular, considering that a PDC is under attack and taking full advantage of the SDN technology, the authors investigate how to re-route the measurements originating from the PMUs to a redundant PDC. Thus, they formulate a self-healing process capable of maintaining and preserving the PMUs' measurements. The ILP models aim to minimise the self-healing process's overhead, taking into account the constraints of the (a) network topology, (b) computing resources and (c) the power system observability. On the other side, the proposed greedy heuristic algorithm calculates the paths of PMUs one by one instead of determining a comprehensive optimum path. The evaluation results in IEEE 30 bus and IEEE 118-bus systems validate the proposed approach.

In [152], M. Rehmani et al. provide a path failure learning method capable of addressing DoS attacks targeting the network links in a smart electrical grid environment with multiple network paths. Three attack types are examined: (a) probabilistic attacks, (b) random attacks and (c) intelligent attacks. For each packet, the proposed method aims to choose a reliable path. In particular, the path selection is transformed into a Multi-Armed Bandit (MAB) problem, which is solved with the e-Greedy method. If the packet is transmitted successfully, then the corresponding path is rewarded. The algorithm chooses the path with the maximum reward mean, while there is a probability e to choose another path, thus satisfying the exploration phase. To evaluate their method, the authors use the Ryu controller and Mininet. Based on the evaluation results, it seems that the algorithm selects commonly the optimal path.

In [130], J. A. Perez-Diaz et al. present an SDN-based architecture for detecting and mitigating low-rate DDoS attacks against Hypertext Transfer Protocol (HTTP). The proposed architecture consists of two main components: (a) IPS and (b) IDS. On the one hand, IPS is responsible for gathering the network flows and mitigating them based on the detection outcome of IDS. In particular, IPS is composed of three modules: (a) Flow Management Module, (b) Suspicious Attackers Management and (c) Mitigation Management Module. The Flow Management Module gathers the HTTP flows from the SDN switches. These flows will be further processed to detect a potential low-rate DDoS attack. HTTP flow statistics are generated through Flowtbag and transmitted to IDS. The Suspicious Attackers Management module handles a blacklist of potential cyberattackers. Finally, the Mitigation Management module follows a mitigation strategy and generates appropriate rules to mitigate the malicious flows. These rules are transmitted to SDN-C. In this work, the Open Network Operating System (ONOS) is utilised as SDN-C. On the other hand, IDS comprises three modules: (a) Identification API, (b) ML Model Selection and (c) Identification. The Identification API manages the communication with the Flow Management Module of IPS. The ML Model Selection Module represents a set of pre-trained ML models. Finally, the Identification module selects one of the pre-trained ML models to analyse the HTTP flow each time. To evaluate their work, the authors use Mininet, SlowHTTPTest and the 2017 CIC DoS dataset. The experimental results confirm the efficiency of the proposed method.

In [206], T. Xing et al. present an SDN-based IPS called SDNIPS. The SDNIPS architecture consists of four modules: (a) Snort agent, (b) SDNIPS daemon, (c) alert interpreter and (d) rules generator. The Snort agent is responsible for detecting the potential cyberattacks by applying the respective signature rules. Next, the SDNIPS daemon undertakes to transform the detection results into a (JavaScript Object Notation) JavaScript Object Notation (JSON) format, which is transmitted to the SDN controller. The alert interpreter processes the JSON files, thus extracting the appropriate information, such as the IP addresses. Finally, the rule generator produces the OpenFlow entries introduced into the Open Virtual Switch (OVS) flow tables. The authors evaluate their IPS with a typical IPS relying on iptables. The evaluation criterion is whether both IPS can generate alerts under tremendous network traffic conditions. To this end, two DoS attacks are emulated. The proposed IPS exceeds the performance of the typical IPS using iptables.

In [116], P. Manso et al. provide an SDN-based IDPS, which combines the Ryu SDN controller and Snort in order to mitigate DoS attacks. The architectural model consists of three virtual machines representing (a) the internal network simulated by Mininet, (b) the SDN-based IDPS and (c) online services. It is noteworthy that the second virtual machine (i.e., that hosting the SDN-based IDPS) hosts both Ryu and Snort. First, Snort receives the overall network traffic through a port mirroring capability provided by OVS of the first virtual machine (i.e., Mininet). If Snort detects a potential cyberattack, it informs Ryu based on a Uniplexed Information and Computing System (UNIX) domain socket. Next, Ryu transmits the appropriate OpenFlow commands to OVS of the first virtual machine (i.e., Mininet), thus isolating the malicious nodes. The authors evaluate their IDPS with three DDoS scenarios, measuring (a) DDoS mitigation time, (b) average Round Trip Time and (c) packet loss. The experimental results demonstrate the efficiency of the proposed IDPS.

4.4 Honeypots and Honeynets

Both academia and industry have implemented several honeypots. In particular, Deception Toolkit (DTK) [10] was the first honeypot released in 1997, emulating known vulnerabilities of UNIX. HoneyBOT [74] is a LIH for Windows systems, simulating relevant vulnerabilities. Similarly, KFSensor [93] is a commercial honeypot for Windows OS. HoneyD [193] is probably the most known honeypot capable of emulating at the same time multiple hosts. Tiny Honeypot [121] is a server-based honeypot, which listens to all TCP ports, logging all interaction activities. Dionaea [169] is written in Python and emulates the MQ Telemetry Transport (MQTT) protocol. Jackpot [121] is related to Simple Mail Transfer Protocol (SMTP) and aims to combat email spam. Cowrie [174] is a LIH emulating SSH. Conpot [49] is an industrial honeypot emulating multiple relevant protocols like Modbus and IEC 60870-5-104. In addition, an overview of Wireless Honeypots (WHs) along history is discussed in [175], where they are defined as nodes that offer wireless access whose value is being probed, attacked, or compromised, letting the attackers to interact with them. In more detail, the main goal of WHs is to gather information about the attacks performed on wireless networks and the associated technologies, focusing on the attacks that exploit the wireless technologies' weaknesses, which are mainly due to the use of unguided transmission medium [6]. The main principles of the WHs can be used in several types of networks, including cellular, Local Area Networks (LANs), sensor networks and Unmanned Aerial Vehicles (UAVs)-based networks [52].

Many supporting tools have also been developed in order to analyse the data retrieved from honeypots or to extend their functionalities [89]. In particular, Bait-n-Switch [189] aims to redirect all malicious traffic to a honeypot. Next, Honeynet Security Console (HSC) [115] analyses, correlates and visualises honeypot logs. Honeysnap [167] processes PCAP files that were collected by server-based honeypots. GSOC-Honeyweb [121] is devoted to the management of client-based honeypots via a user-friendly environment. Moreover, TraCINg [121] aggregates data from multiple honeypots and correlates this information in order to discover possible worms.

It is noteworthy that many honeypots projects have been organised in order to exploit at the maximum level the benefits of honeypots and discover potential zero-day attacks. In particular, the Honeynet Project was started in 1999 to explore and investigate zero-day cyberattacks. Furthermore, the Leurre.com project [102] deployed multiple LIHs in more than 30 counties, aiming at collecting quantitative data related to cyberthreats and vulnerabilities. Accordingly, NoAHProject coordinated by FORTH deployed an HIH called Argos [137] to enhance the protection of ISPs and investigate potential zero-day attacks. The mw-collect Alliance project collected information about various malware by deploying multiple Nepenthes sensors [137]. Moreover, Telekom-Fruhwarnsystem [137] was started in 2013 to collect various datasets related to honeypot activities. Finally, H2020 SPEAR [142] and H2020 SDN-microSENSE [136] implemented various industrial honeypots for the smart electrical grid.

4.5 Summary and Discussion

Undoubtedly the previous works provide valuable methods and systems. In particular, 20 signature/specification-based IDPS are discussed, while 16 IDPS use anomaly-based techniques. Moreover, five IDPS take full advantage of the SDN technology in order to mitigate and prevent the various attacks. Moreover, three IDPS of the above analysis monitor the entire SG ecosystem, while 13 IDPS focus on AMI. Next, ten

IDPS are devoted to the protection of the SCADA systems, while eight and three IDPS focus on substations and synchrophasors, respectively. As already discussed, each detection category is characterised by corresponding advantages and disadvantages. The signature-based IDPS usually achieve high detection performance; however, they cannot recognise unknown attacks and anomalies. On the other side, anomaly-based IDPS can detect zero-day attacks and unknown anomalies, but they are characterised by a high number of false alarms. Finally, the specification-based IDPS combine the benefits of the previous categories; however, they cannot discriminate the attack or anomaly type. Also, the generation of signature and specification rules is a time-consuming process, taking into account the different characteristics of each environment. Consequently, it seems that a hybrid approach, combining the previous methods, is the most appropriate solution.

In addition, it is worth mentioning that the previous works do not present information about the detection time; however, the detection latency is an important measure, taking into account the sensitive nature of the IIoT environments. Furthermore, the various IDPS should consider the limited computing resources of the IoT entities. Moreover, it is noteworthy that most of the IDPS examined in this chapter are not quite scalable, considering that they do not use data from multiple sources. Most of them focus on network traffic data without considering heterogeneous operational data and values. Additionally, despite the fact that many works focus on industrial protocols, like Modbus/TCP, DNP3 and IEC 61850, they do not investigate and analyse their attributes at the application layer. Also, the current works do not adopt visualisation methods in order to recognise better potential cyberattacks/anomalies and reduce the number of false alarms. Apart from the SDN-based IDPS, the other ones do not also include sufficient mitigation and self-healing mechanisms. Moreover, although SDN can lead to the automated mitigation of malicious activities, the presence of false alarms can result in more disastrous consequences, given that the continuous operation of the IIoT environments is necessary. Therefore, a wrong decision can lead the SDN controller to stop a normal and legitimate operation with the corresponding negative effects. Finally, most of the current works do not compute quantitatively the severity of the various cyberattacks and anomalies against the industrial protocols.

In general, despite the importance of the current IDPS, they do not fully satisfy the requirements defined in the previous chapter. For this purpose, cross-layer mechanisms focusing on situational awareness are necessary. According to Endsley [43], situational awareness consists of three layers. The first layer focuses on the perception of information, identifying the charcateristics and the elements composing the target system. Next, the second layer refers to the comprehension of information. For this purpose, storing and interpretation mechanisms are necessary. Finally, the projection level includes predictive and prescriptive algorithms that intend to interpret relevant events. According to [148], B. McGuinness and L. Foy introduce an extra layer called Resolution. This layer aims to identify the appropriate methods and practices that optimise a current situation. Therefore, based on the aforementioned remarks, it is evident that the current detection mechanisms should follow a hybrid approach, taking into account both the cyber and physical attributes of the target system. Secondly, the proper and continuous interpretation of this information is necessary. For this purpose, the analysis of the application-layer protocols used by the IoT systems is critical. Finally, although there can be multiple countermeasures, such as SDN and firewall rules, the identification of the appropriate mitigation and prevention strategy is necessary, taking into account the special characteristics of each situation.

4.6 Chapter Summary

This chapter provides a comprehensive literature review about IDPS protecting the energy sector. In particular, based on the detection categories and the IDPS requirements described previously, this chapter describes and analyses relevant works in this research area. For each detection category, various works are discussed in detail, while Appendix D summarises this analysis. Moreover, a particular emphasis is given to honeypots and honeynets, describing relevant implementations, complementary honeypot tools and honeypot-related projects. Next, the important role of SDN and SDN-enabled IDPS is highlighted, discussing relevant works. Finally, based on this analysis, the strengths and limitations of the current solutions are identified, thus guiding the implementation of the proposed SDN-enabled SIEM detailed in the following chapter.

Chapter 5

Detection and Mitigation of Cyberattacks and Anomalies against Smart Grid

Based on the previous literature review about the IDPS systems in IIoT/SG environments, this chapter aims to summarise and present in detail the technical achievements that took place during this PhD program. In particular, they focus on intrusion detection and prevention mechanisms that are related to IIoT/SG, taking full advantage of novel technologies, such as SDN and AI. It is worth mentioning that in the context of this thesis, the various technical achievements, such as AI-powered IDPS, threat models, mitigation strategies and honeypot deployment mechanisms, are presented as a unified SDN-enabled SIEM solution. Therefore, all the achievements implemented and published during this PhD program are incorporated conceptually into an SDN-enabled SIEM system, where the various architectural elements collaborate with each other in order to detect and mitigate potential intrusions and anomalies in a timely manner. The following sections describe in detail the proposed SDN-enabled SIEM in terms of the corresponding architectural components, their mechanisms and how they interact with the IIoT/SG entities/devices.

5.1 Architecture of the Proposed SDN-enabled SIEM

According to the SDN paradigm, Fig. 5.1 illustrates the architectural design of the proposed SDN-enabled SIEM. The primary goal is to detect, normalise, correlate and mitigate cybersecurity incidents against IIoT/SG environments, taking full advantage of SDN, honeypots and AI. In particular, three AI-powered IDPS were implemented, generating relevant security events, while the Normalisation, Correlation and Mitigation Engine (NCME) undertakes to normalise and correlate them, thus composing security alerts. In addition, NCME guides appropriately the SDN-C and includes sophisticated honeypot deployment mechanisms in order to mitigate the malicious network flows and increase the resilience of the underlying IIoT/SG infrastructure, respectively.

First, the Network Flow-based IDPS (NF-IDPS) focuses on detecting cyberattacks and anomalies against application-layer industrial communication protocols, such as Modbus/TCP, DNP3, IEC 60870-5-104, IEC 61850 (GOOSE), HTTP and SSH. For each of the previous protocols, appropriate ML/DL intrusion and anomaly detection models were implemented, utilising custom and publicly available datasets. Next, the H-IDPS is responsible for detecting potential anomalies based on operational electricity data from IIoT/SG environments. Next, V-IDPS focuses on detecting malicious Modbus/TCP network flows, taking full advantage of binary visual representations and AI.

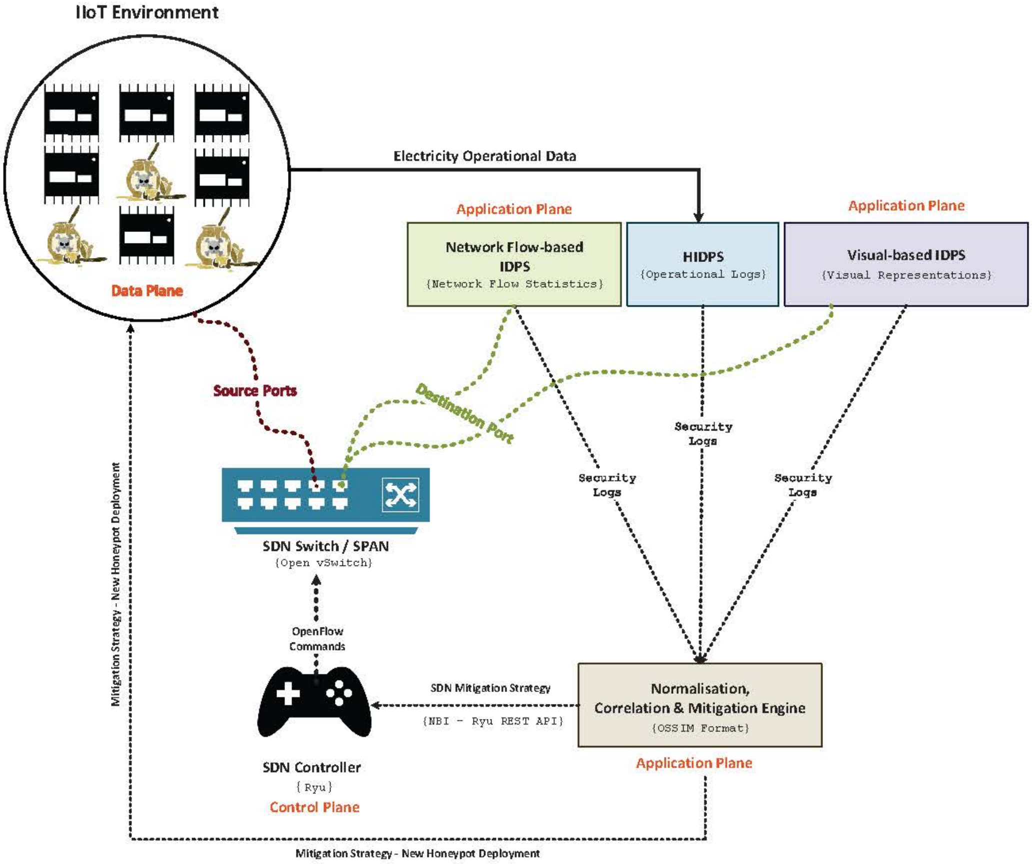


Figure 5.1: Architecture of the Proposed SDN-enabled SIEM

Next, NCME undertakes to normalise and correlate the security events from the previous IDPS. For this process, the AlienVault Open Source SIEM (OSSIM) format was adopted, while security rules are used to correlate the security events with each other. Moreover, NCME incorporates an RL-based mechanism in order to guide appropriately the SDN-C about dropping the malicious network flows. Finally, NCME includes sophisticated honeypot deployment mechanisms that rely on a security game between two players: the attacker and the defender.

More technical details about the aforementioned components are given in the following sections.

5.2 NF-IDPS: Network Flow-based Intrusion Detection and Prevention System

As illustrated in Fig. 5.2, the Network Flow-based Intrusion Detection and Prevention System (NF-IDPS) is located in the application plane (according to the SDN architectural paradigm) and consists of four modules: (a) Network Traffic Capturing Module (NTCM), (b) Network Flow Extraction Module (NFEM), (c) Intrusion Detection Engine (IDE) and (d) Notification Module (NM). First, the NTCM can monitor and capture the network traffic data (i.e., pcap/pcapng files) of the overall SDN network through a Switched Port Analyser (SPAN). To this end, tcpdump is utilised. Next, the NFEM receives the network traffic data from the NTCP and generates the corresponding network flow statistics. In particular, two kinds of flow statistics are produced. The first one refers to bidirectional TCP/IP flow statistics of the network packets. These statistics are generated through CICFlowMeter. On the other side, the second kind refers to bidirectional flow statistics related to application-layer industrial protocols, such as DNP3 and IEC 60870-5-104. For this purpose, custom Python decoders were implemented. Both cases are characterised by a time limit, which affects (a) the flows number, (b) the flows statistics and, therefore, (c) the detection performance. This limit can be defined experimentally depending on the network characteristics of each environment. According to P. Radoglou-Grammatikis et al. in [144] show that 120 seconds is an appropriate value for detecting anomalies related to IEC 60870-5-104 flows.

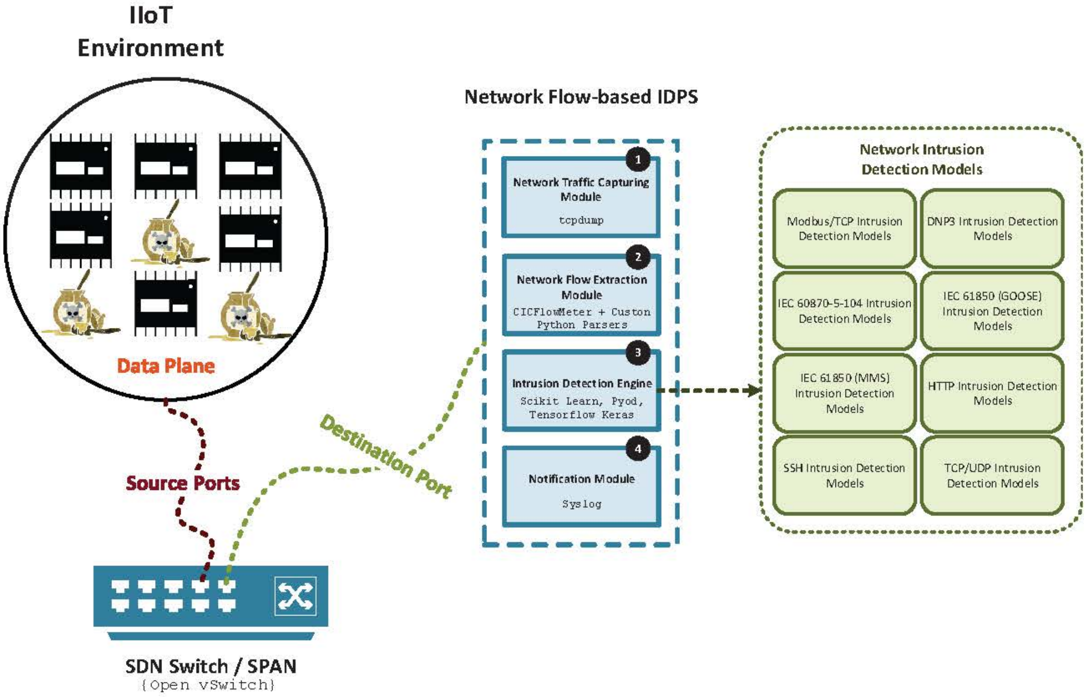

IIoT Environment
Data Plane
Source Ports
Destination Port
SDN Switch / SPAN
(Open vSwitch)
Network Flow-based IDPS
1
Network Traffic Capturing Module
tcpdump
2
Network Flow Extraction Module
CICFlowMeter + Custon Python Parsers
3
Intrusion Detection Engine
Scikit Learn, Pyod, Tensorflow Keras
4
Notification Module
Syslog
Network Intrusion Detection Models
Modbus/TCP Intrusion Detection Models
DNP3 Intrusion Detection Models
IEC 60870-5-104 Intrusion Detection Models
IEC 61850 (GOOSE) Intrusion Detection Models
IEC 61850 (MMS) Intrusion Detection Models
HTTP Intrusion Detection Models
SSH Intrusion Detection Models
TCP/UDP Intrusion Detection Models

FIGURE 5.2: Architecture of the Network Flow-based Intrusion Detection and Prevention System

Next, the IDE is responsible for detecting the presence of a potential intrusion or anomaly, including

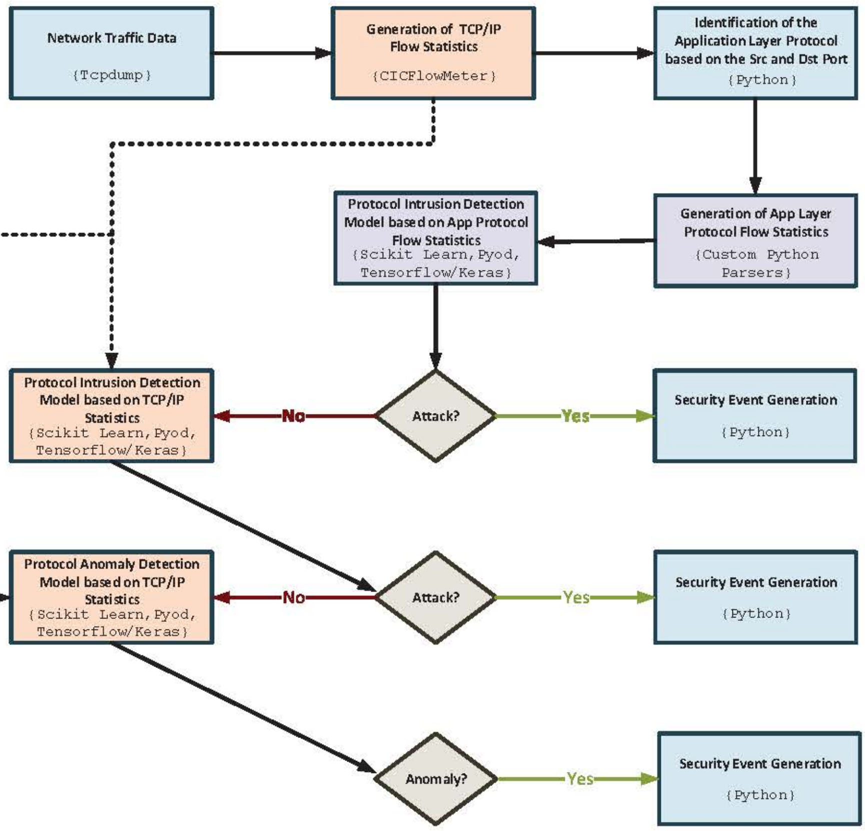

Network Traffic Data
{Tcpdump}
Generation of TCP/IP Flow Statistics
{CICFlowMeter}
Identification of the Application Layer Protocol based on the Src and Dst Port
{Python}
Protocol Intrusion Detection Model based on App Protocol Flow Statistics
{Scikit Learn, Pyod, Tensorflow/Keras}
Generation of App Layer Protocol Flow Statistics
{Custom Python Parsers}
Protocol Intrusion Detection Model based on TCP/IP Statistics
{Scikit Learn, Pyod, Tensorflow/Keras}
No
Attack?
Yes
Security Event Generation
{Python}
Protocol Anomaly Detection Model based on TCP/IP Statistics
{Scikit Learn, Pyod, Tensorflow/Keras}
No
Attack?
Yes
Security Event Generation
{Python}
Anomaly?
Yes
Security Event Generation
{Python}

FIGURE 5.3: NF-IDPS Operation Flowchart

a wide range of intrusion/anomaly detection models. In this thesis, intrusion refers to a particular cyberattack type, while anomaly denotes that the behaviour of the target system is not usual without distinguishing the attack type. In particular, IDE focuses on detecting cyberattacks and anomalies against the IIoT application-layer protocols, such as Modbus/TCP, DNP3, IEC 60870-5-104, IEC 61850 (MMS), HTTP and SSH. For each protocol, various detection models are utilised according to the operation flowchart illustrated in Fig. 5.3. These models are classified into three main categories: (a) Application-Layer Intrusion Detection Models, (b) TCP/IP Intrusion Detection Models and (c) TCP/IP Anomaly Detection Models. First, the ML/DL models of the first category are adopted, utilising flow statistics related to the attributes of the application-layer protocols. Consequently, based on the TCP/User Datagram Protocol (UDP) source (src) and destination (dst) ports, first, the application-layer protocol is identified, and the corresponding Application-Layer Intrusion Detection Model is applied, detecting the presence of potential cyberattacks. For instance, if the source port is equal to 2404, then the IEC 60870-5-104 Intrusion Detection Model is used. Depending on the detection outcomes, the respective security event(s) are generated or differently the second category is activated. The ML/DL models of the second category use flow statistics from the transport and network layers of the TCP/IP stack. They can also discriminate a particular cyberattack type, generating the corresponding security event(s). Otherwise, the last category is activated, trying to identify a potential anomaly and producing the corresponding security event(s). Similarly, in this case, flow statistics from the transport and network layers are used by the ML/DL models. It is worth mentioning that in each category, the various intrusions and anomalies are always associated with the corresponding application-layer protocols. In particular, IDE includes multiple intrusion and anomaly detection models for the following application-layer protocols: (a) Modbus/TCP, (b) DNP3, (c) IEC 60870-5-104, (d) IEC 61850 (MMS), (e) HTTP and (f) SSH. More technical details for the intrusion and anomaly detection models of the above protocols are given in the following subsections. Finally, the NM undertakes to send the security events(s) to NCME.

5.2.1 Modbus/TCP Intrusion and Anomaly Detection Models

The Modbus/TCP intrusion and anomaly detection models in this thesis rely on a threat model combining (a) Spoofing, Tampering, Repudiation, Information Disclosure, DoS, and Elevation of Privilege (STRIDE)-per-element, (b) ADT, (c) CVSS and (d) OWASP Risk Rating (OWASP-RR) methodology. The main goal is to identify the various Modbus/TCP cyberattacks and prioritise their severity, considering their probability and impact as isolated and combined cases against the essential cybersecurity principles: Confidentiality, Integrity and Availability (CIA). First, STRIDE-per-element is adopted in order to define the primary cyberattack super-classes reflecting the target behind the various Modbus/TCP cyberattacks. Next, ADT is used to map and combine those Modbus/TCP cyberattacks with the STRIDE-per-element super-classes. Subsequently, both CVSS and OWASP-RR are utilised for calculating the severity of each Modbus/TCP threat. Finally, the logical relationships among the nodes of the ADT are used to estimate the severity of the STRIDE-per-element classes. Consequently, the proposed Modbus/TCP threat model combines the benefits of each methodology, thus determining the severity of the individual Modbus/TCP threats and their super-class. The following paragraphs provide a brief description for each of the aforementioned methodologies, while subsequently, the Modbus/TCP-related ADT and the respective CVSS and OWASP-RR scores are analysed.

First, STRIDE is an acronym that stands for Spoofing, Tampering, Repudiation, Information Disclosure, DoS, and Elevation of Privilege. It was developed by L. Kohnfelder and P. Garg and was adopted by Microsoft in 2008 [173]. In the context of this thesis, the variant called STRIDE-per-element [173] is used to identify the Modbus/TCP threats supported by existing penetration testing tools. In particular, five penetration testing tools are investigated: Smod, Metasploit, Nmap, mbtget and ModScan. Hence, 14 cyberattacks (Table 5.1) are identified. These cyberattacks are considered as Data Flow elements. Thus, from the initial STRIDE attack families, only three families are taken into account: (a) Tampering, (b) Information Disclosure and (c) DoS.

Subsequently, ADT is used to structure and visualise the Modbus/TCP threats. In particular, an ADT consists of two opponent nodes: (a) attacking nodes and (b) defending nodes [90]. The first category expresses the goal and the malicious activities that a cyberattacker may perform to violate the security of the target system. On the other side, the defending nodes indicate the countermeasures that the defender can adopt in order to mitigate or even prevent the cyberattacks. Each node can be expanded with one or more children of the same type, thus defining refinements that indicate sub-goals and actions. In addition, each node can have children of the opposite type, denoting threats or countermeasures, respectively. The refined nodes can be divided into two types (a) conjunctive and (b) disjunctive. In the first case, a conjunctively refined node carries out its goal if all of its children accomplish necessarily their goals. In contrast, the goal of a disjunctively refined node is achieved if at least one of its children carries out its goal. Therefore, the conjunctive and disjunctive refinements are represented by the AND and OR logical operators, respectively.

Finally, CVSS and the OWASP-RR methodology are used to evaluate the severity of each Modbus/TCP threat quantitatively and qualitatively. Both of them can be used independently and rely on different methodologies. In particular, CVSS is an open vulnerability assessment framework, which quantifies the severity of each vulnerability or attack between 0 and 10. CVSS consists of three metric groups, namely (a) Base Group, (b) Temporal Group and (c) Environmental Group. The Base Group reflects the intrinsic features of the vulnerability/attack. These features cannot be affected over time or modified by compensating factors. The Temporal Group focuses on vulnerabilities/attacks that evolve or change over time, evaluating their exploitability as well as the availability of the respective security controls. Finally, the Environmental Group enables an organisation to adjust appropriately the values of the Base Group, taking into account its own security requirements. In [105], E. Li et al. explain how the CVSS score is computed, respectively. On the other side, the calculation of the OWASP-RR score is more straightforward than CVSS. Actually, the OWASP-RR score [158] is calculated by Equation 5.1. Both Likelihood and Impact depend on additional factors. In particular, Likelihood expresses the possibility of the occurrence of each identified threat, and it is computed by averaging the values of the Threat Agent Factor and the Vulnerability Factor. The Threat Factor is calculated by summing the values of four factors: (a) Skill Level, (b) Motive, (c) Opportunity and (d) Size. Similarly, the Vulnerability Factor is computed by adding four factors: (a) Ease of Discovery, (b) Ease of Exploit, (c) Awareness and (d) Intrusion Detection. Accordingly, Impact represents the consequences if that threat eventuates, and it is determined by averaging the values of the Technical Impact Factor and the Business Impact Factor. In a similar manner, the Threat Impact Factor is calculated by summing the values of four factors: (a) Loss of Confidentiality, (b) Loss of Integrity, (c) Loss of Availability and (d) Loss of Accountability. On the other hand, the Business Impact Factor is also calculated by summing the values of four factors: (a) Financial Damage, (b) Reputation Damage, (c) Non-compliance and (d) Privacy Violation. The values of the aforementioned factors range between $0-9$. E. Rios et al. in [158] provide more insights regarding how OWASP-RR score is computed.

Fig. 5.4 illustrates the proposed ADT. The STRIDE elements represent the refined nodes, while 14 cyberattacks supported by the aforementioned Modbus/TCP-related penetration testing tools denote the non-refined nodes. Therefore, Tampering is related to the integrity principle and is composed of two disjunctive refinements: (a) modbus/function/writeSingleCoils and (b) modbus/function/writeSingleRegister. Similarly, DoS refers to the availability requirement and consists of six disjunctive refinements: (a) modbus/dos/writeSingleCoils, (b) modbus/dos/writeSingleRegister, (c) modbus/function/readCoils (DoS), (d) modbus/function/readCoils (DoS), (e) modbus/function/readInputRegister (DoS) and (f) modbus/function/readDiscreteInput (DoS). Finally, Information Disclosure corresponds to the confidentiality principle and comprises six disjunctive refinements, namely (a) modbus/function/readCoils, (b) modbus/scanner/getfunc, (c) modbus/scanner/uid, (d) modbus/function/readInputRegister, (e) modbus/function/readHoldingRegister and (f) modbus/function/readDiscreteInput. The aforementioned cyberattacks take full advantage of the fact that Modbus/TCP does not include any authentication and authorisation mechanism, thus allowing a cyberattacker to use the Modbus/TCP function codes for malicious purposes. Table 5.1 provides a description for each non-refined node, including the CVSS and OWASP-RR textual representations. The textual representations reflect the values of the CVSS and OWASP-RR criteria that lead to the respective quantitative scores. The names of the non-refined nodes originate from the corresponding modules of the aforementioned penetration testing tools. For each non-refined node, CVSS and OWASP-RR are applied individually, calculating the corresponding severity scores. Next, these scores are propagated to the upper nodes based on Equation 5.2 and Equation 5.3. In particular, Equation 5.2 is applied when the refined node comprises conjunctive refinements since the parent's goal is achieved whether all children accomplish their goal. Therefore, the severity score of a conjunctively refined node is equal to the product of the childrens' severity scores. The product indicates the probability behind the severity score of each child [105]. In contrast, Equation 5.3 is utilised when the refined node includes disjunctive refinements since the respective goal is achieved whether a child will accomplish its goal. Consequently, the severity score of the disjunctively refined node equates with the maximum severity score of the various children.

Based on these computations, both CVSS and OWASP-RR estimate the severity of each Modbus/TCP threat as "high". Fig. 5.4 presents the quantitative scores. Finally, the proposed ADT includes a countermeasure called Intrusion Detection and Mitigation. This countermeasure comprises two conjunctive refinements: (a) Intrusion Detection and (b) SDN-based mitigation. The first one is responsible for the timely detection of the Modbus/TCP threats and includes two disjunctive refinements. On the other side, SDN-based mitigation refers to the mitigation of the Modbus/TCP threats, taking full advantage of the SDN technology.

$$OWASP - RR_{Risk} = Likelihood \times Impact \tag{5.1}$$

$$\begin{aligned} &CVSS(\text{or OWASP-RR})_{RefinedNode} = \\ &\prod_{i=1}^{n} CVSS(\text{or OWASP-RR})_{Refinement_i} \end{aligned} \tag{5.2}$$

$$\begin{aligned} CVSS(\text{or OWASP-RR})_{RefinedNode} = \max\{ \\ (CVSS(\text{or OWASP-RR})_{Refinement_1}) \\ ,(CVSS(\text{or OWASP-RR})_{Refinement_2}) \\ ,...,(CVSS(\text{or OWASP-RR})_{Refinement_n})\} \end{aligned} \tag{5.3}$$

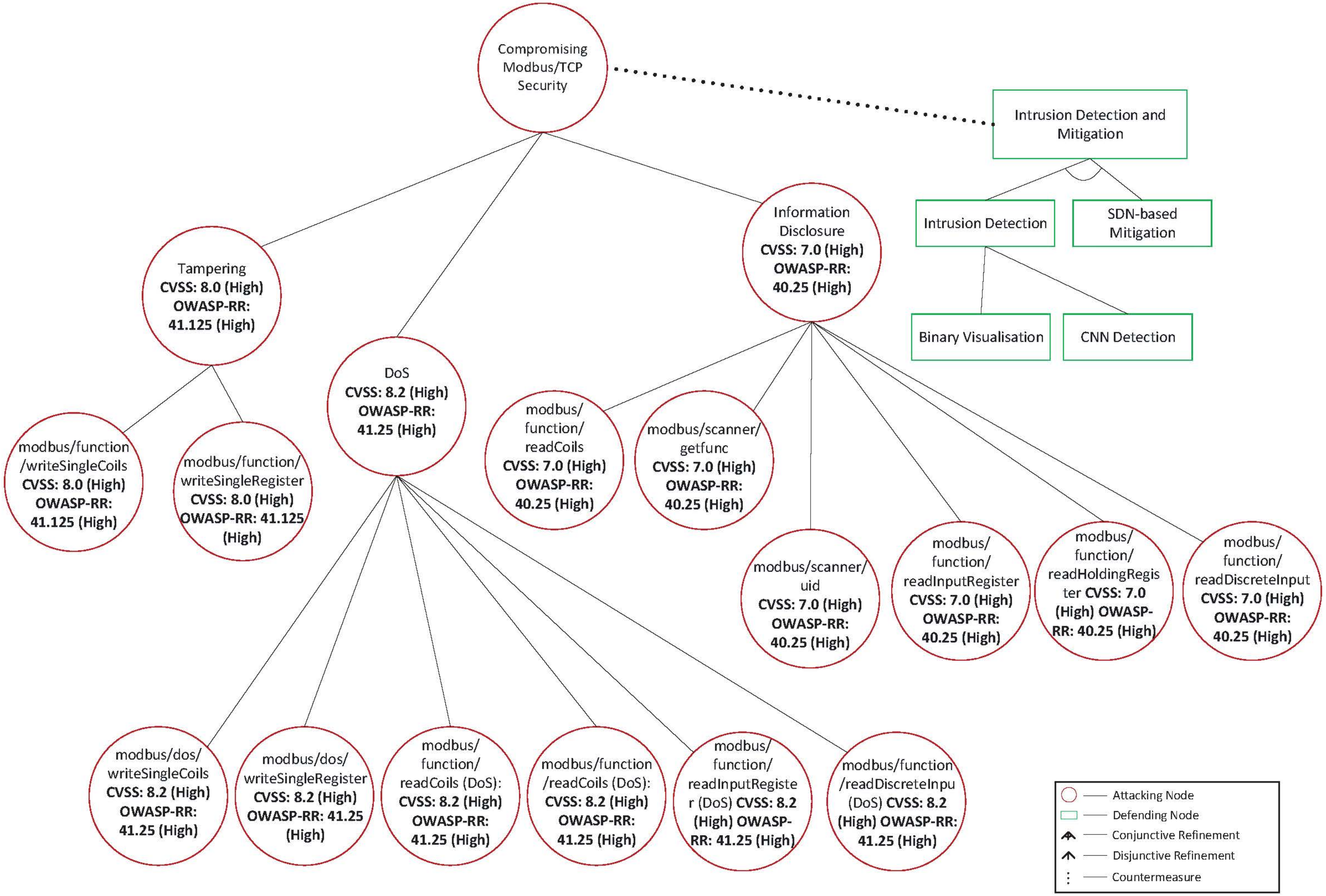

Compromising Modbus/TCP Security
Intrusion Detection and Mitigation
Intrusion Detection
SDN-based Mitigation
Binary Visualisation
CNN Detection
Tampering CVSS: 8.0 (High) OWASP-RR: 41.125 (High)
DoS CVSS: 8.2 (High) OWASP-RR: 41.25 (High)
Information Disclosure CVSS: 7.0 (High) OWASP-RR: 40.25 (High)
modbus/function /writeSingleCoils CVSS: 8.0 (High) OWASP-RR: 41.125 (High)
modbus/function/ writeSingleRegister CVSS: 8.0 (High) OWASP-RR: 41.125 (High)
modbus/ function/ readCoils CVSS: 7.0 (High) OWASP-RR: 40.25 (High)
modbus/scanner/ getfunc CVSS: 7.0 (High) OWASP-RR: 40.25 (High)
modbus/scanner/ uid CVSS: 7.0 (High) OWASP-RR: 40.25 (High)
modbus/ function/ readInputRegister CVSS: 7.0 (High) OWASP-RR: 40.25 (High)
modbus/ function/ readHoldingRegister CVSS: 7.0 (High) OWASP-RR: 40.25 (High)
modbus/ function/ readDiscreteInput CVSS: 7.0 (High) OWASP-RR: 40.25 (High)
modbus/dos/ writeSingleCoils CVSS: 8.2 (High) OWASP-RR: 41.25 (High)
modbus/dos/ writeSingleRegister CVSS: 8.2 (High) OWASP-RR: 41.25 (High)
modbus/ function/ readCoils (DoS): CVSS: 8.2 (High) OWASP-RR: 41.25 (High)
modbus/function /readCoils (DoS): CVSS: 8.2 (High) OWASP-RR: 41.25 (High)
modbus/ function/ readInputRegister (DoS) CVSS: 8.2 (High) OWASP-RR: 41.25 (High)
modbus/function /readDiscreteInpu (DoS) CVSS: 8.2 (High) OWASP-RR: 41.25 (High)
Attacking Node
Defending Node
Conjunctive Refinement
Disjunctive Refinement
Countermeasure

Figure 5.4: Modbus/TCP Threat Model

TABLE 5.1: Non-refined Modbus/TCP threats with CVSS and OWASP-RR representations

Modbus/TCP Threat	Description	CVSS Represenatation	OWASP-RR Representation	CVSS Score	OWASP Score
modbus/function/ writeSingleCoils	It changes the value of a single coil via function code 05	AV:N/AC:L/PR:L/UI:R/S:C/C:N/I:H/A:N/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:C/MC:N/MI:H/MA:N	SL:1/M:9/O:4/S:4/ED:9/EE:9/A:7/ID:4/LC:0/LI:8/LAV:3/LAC:8/FD:8/RD:7/NC:7/PV:6	8.0	41.125
modbus/function/ writeSingleRegister	It changes the value of a single register via function code 06	AV:N/AC:L/PR:L/UI:R/S:C/C:N/I:H/A:N/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:C/MC:N/MI:H/MA:N	SL:1/M:9/O:4/S:4/ED:9/EE:9/A:7/ID:4/LC:0/LI:8/LAV:3/LAC:8/FD:8/RD:7/NC:7/PV:6	8.0	41.125
modbus/function/ readCoils	It reads the value of a single coil via function code 01	AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:L/A:N/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:U/MC:H/MI:L/MA:N	SL:1/M:9/O:4/S:4/ED:9/EE:9/A:7/ID:3/LC:9/LI:0/LAV:0/LAC:6/FD:7/RD:8/NC:7/PV:6	7.0	40.25
modbus/scanner/ getfunc	It lists all function codes of the target system	AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:L/A:N/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:U/MC:H/MI:L/MA:N	SL:1/M:9/O:4/S:4/ED:9/EE:9/A:7/ID:3/LC:9/LI:0/LAV:0/LAC:6/FD:7/RD:8/NC:7/PV:6	7.0	40.25
modbus/function/ readHoldingRegister	It reads the content of a holding register via a function code 03	AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:L/A:N/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:U/MC:H/MI:L/MA:N	SL:1/M:9/O:4/S:4/ED:9/EE:9/A:7/ID:3/LC:9/LI:0/LAV:0/LAC:6/FD:7/RD:8/NC:7/PV:6	7.0	40.25
modbus/scanner/uid	It enumerates the user IDs of the target system	AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:L/A:N/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:U/MC:H/MI:L/MA:N	SL:1/M:9/O:4/S:4/ED:9/EE:9/A:7/ID:3/LC:9/LI:0/LAV:0/LAC:6/FD:7/RD:8/NC:7/PV:6	7.0	40.25
modbus/function/ readInputRegister	It reads the content of an Input Register via function code 04	AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:L/A:N/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:U/MC:H/MI:L/MA:N	SL:1/M:9/O:4/S:4/ED:9/EE:9/A:7/ID:3/LC:9/LI:0/LAV:0/LAC:6/FD:7/RD:8/NC:7/PV:6	7.0	40.25
modbus/function/ readDiscreteInput	It reads the content of a discrete input via function code 02	AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:L/A:N/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:U/MC:H/MI:L/MA:N	SL:1/M:9/O:4/S:4/ED:9/EE:9/A:7/ID:3/LC:9/LI:0/LAV:0/LAC:6/FD:7/RD:8/NC:7/PV:6	7.0	40.25
modbus/dos/ writeSingleCoils	It floods the target system with packets with function code 05	AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:H/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:C/MC:L/MI:L/MA:H	SL:1/M:9/O:4/S:4/ED:9/EE:8/A:6/ID:3/LC:2/LI:1/LAV:8/LAC:8/FD:8/RD:8/NC:8/PV:6	8.2	41.25
modbus/dos/ writeSingleRegister	It floods the target system with packets with function code 06	AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:H/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:C/MC:L/MI:L/MA:H	SL:1/M:9/O:4/S:4/ED:9/EE:8/A:6/ID:3/LC:2/LI:1/LAV:8/LAC:8/FD:8/RD:8/NC:8/PV:6	8.2	41.25
modbus/function/ readCoils (DoS)	It floods the target system with packets with function code 01	AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:H/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:C/MC:L/MI:L/MA:H	SL:1/M:9/O:4/S:4/ED:9/EE:8/A:6/ID:3/LC:2/LI:1/LAV:8/LAC:8/FD:8/RD:8/NC:8/PV:6	8.2	41.25
modbus/function/ readHoldingRegister (DoS)	It floods the target system with packets with function code 03	AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:H/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:C/MC:L/MI:L/MA:H	SL:1/M:9/O:4/S:4/ED:9/EE:8/A:6/ID:3/LC:2/LI:1/LAV:8/LAC:8/FD:8/RD:8/NC:8/PV:6	8.2	41.25
modbus/function/ readInputRegister (DoS)	It floods the target system with packets with function code 04	AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:H/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:C/MC:L/MI:L/MA:H	SL:1/M:9/O:4/S:4/ED:9/EE:8/A:6/ID:3/LC:2/LI:1/LAV:8/LAC:8/FD:8/RD:8/NC:8/PV:6	8.2	41.25
modbus/function/ readDiscreteInput (DoS)	It floods the target system with packets with function code 02	AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:H/E:F/RL:T/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MUI:R/MS:C/MC:L/MI:L/MA:H	SL:1/M:9/O:4/S:4/ED:9/EE:8/A:6/ID:3/LC:2/LI:1/LAV:8/LAC:8/FD:8/RD:8/NC:8/PV:6	8.2	41.25

Based on the aforementioned remarks, two Modbus/TCP-related intrusion and anomaly detection models are used: namely (a) Modbus/TCP Intrusion Detection Model and (b) Modbus/TCP Anomaly Detection Model. The first model uses a decision tree classifier, which is capable of discriminating the above Modbus/TCP cyberattacks. On the other side, the second model uses a custom autoencoder [138], which can recognise relevant anomalies based on the operation flowchart depicted in Fig. 5.3. Both models rely on TCP/IP flow statistics provided in Appendix E. Fig. 5.5 shows the architecture of the proposed autoencoder. It is composed of six fully connected layers and maps input data $x \in X = R^n$ to an output $x' \in X$. In particular, it consists of an encoder $f: X \implies Z$ and a decoder $g: Z \implies X$, which together result in the output $x' = g(f(x))$. The low-dimensional latent representation of x is obtained from the encoder and is defined as $z = f(x) \in Z = R^m(m << n)$. As a result of this dimensionality reduction, the proposed autoencoder avoids becoming an identity function, and the training process aims to minimise the reconstruction error $L(x, x')$, which is typically the Euclidean distance in space X. Since the proposed autoencoder is trained, anomalies are detected by measuring the reconstruction error $L(x, x')$ and comparing it with a threshold T, classifying all operational data samples y with $L(y, g(f(y))) > T$ as anomalies. The selected threshold T is estimated heuristically based on the reconstruction error L of all normal training data samples. In practice, the threshold T, in order to be more robust, is selected to be a large percentile of the reconstruction error $T = p0.9(L(x, x')|x \in X)$ or if a validation dataset is available, it is selected to maximise the performance for the validation data. It is noteworthy that the training dataset should only consist of normal observations, and therefore it is expected to be reconstructed well. Regarding the training procedure, a labelled Modbus/TCP intrusion detection dataset was implemented, using the aforementioned penetration testing tools (i.e., (a) Smod, (b) Metasploit, (c) Nmap, (d) mbtget and (e) ModScan) and both actual and emulated Modbus/TCP industrial devices. This dataset will be published soon in IEEE Dataport and Zenodo.

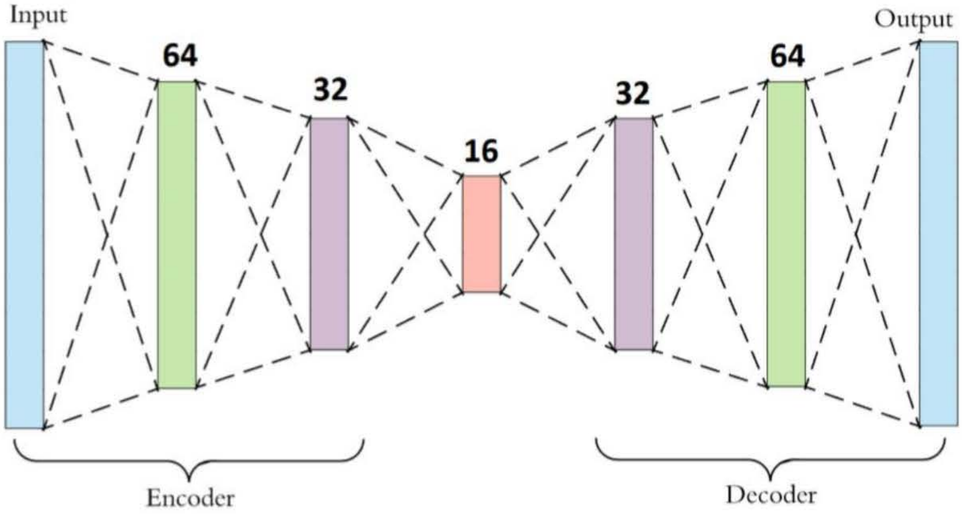


FIGURE 5.5: Autoencoder for Anomaly Detection

5.2.2 DNP3 Intrusion Detection Models

DNP3 is a reliable protocol applied largely in CIs in the US. In particular, DNP3 is adopted to transfer messages between master devices and outstations. It supports several topologies, including (a) point-to-point, where an outstation and one master communicate with each other, (b) multiple-drop, where several masters and outstations interact with each other and (c) hierarchical interface, where an entity can operate with both roles. DNP3 includes three layers: (a) link layer, (b) transport layer and (c) application layer. The link-layer offers addressing services, multiplexing, data fragmentation, error checking and link control. On the other side, the transport layer is used as in the case of the Open Systems Interconnection (OSI) model, and it is represented with one byte utilised for fragmenting the

DNP3 packets. Finally, the application layer defines a set of functional commands used for managing and controlling the IIoT entities. Apart from the DNP3 serial line communication, DNP3 can be used over TCP/IP, where in this case, the aforementioned DNP3 layers are incorporated into the application layer of TCP/IP.

Three intrusion and anomaly detection models were implemented for the DNP3 protocol, namely: (a) DNP3 Intrusion Detection Model, (b) DNP3 TCP/IP Intrusion Detection Model and (c) DNP3 TCP/IP Anomaly Detection Model. The first one uses DNP3 flow statistics of Appendix F, while the other ones rely on TCP/IP flow statistics in Appendix E. Regarding the detection process, after a comparative study, a decision tree classifier is adopted for the first two models, while the third model adopts ABOD for detecting DNP3-related anomalies. For the training procedure, a custom DNP3 intrusion detection dataset was generated in the context of this thesis. This dataset will be available soon in IEEE Dataport and Zenodo. Both models of the aforementioned models can detect the following DNP3 attacks.

- **DNP3 Enumerate**: This reconnaissance attack aims to discover which DNP3 services and functional codes are used by the target system.
- **DNP3 Info**: This attack constitutes another reconnaissance attempt, collecting various DNP3 diagnostic information.
- **DNP3 Disable Unsolicited Messages Attack**: This attack targets an outstation device, establishing a connection with it while acting as a master station. The false master then transmits a packet with the DNP3 Function Code 21, which requests to disable all the unsolicited messages on the target.
- **DNP3 Cold Restart Message Attack**: In a similar manner to the previous attack, the attacker acts as the master station and sends a DNP3 packet that includes the Cold Restart function code. When the target receives this message, it initiates a complete restart and sends a reply with the time window available before the restart.
- **DNP3 Warm Restart Message Attack**: This attack is quite similar to the Cold Restart Message, but aims to trigger a partial restart, re-initiating a DNP3 service on the target outstation.
- **Stop Application**: This attack is related to the Function Code 18 (Stop Application) and requires from the slave to stop its function so that the slave cannot receive messages from the master.
- **Data Initialisation**: This cyberattack is related to Function Code 15 (Initialize Data). It is an unauthorised attack, which demands from the slave to re-initialise possible configurations in their initial values, thus changing potential values defined by legitimate masters.
- **Replay Attack**: This cyberattack replays DNP3 packets coming from a legitimate DNP3 master or slave.

5.2.3 IEC 60870-5-104 Intrusion Detection Models

IEC 60870-5-104 is a communication protocol provided by the IEC 60870-5 standard for monitoring and controlling automated processes in energy applications by utilising the transport capabilities offered by TCP/IP. In particular, it utilises, by default, the TCP port 2404. Fig. 5.6 illustrates the payload of this protocol which is named Application Protocol Data Unit (APDU). APDU consists of two parts, namely (a) Application Protocol Control Information (APCI) and (b) Application Service Data Unit (ASDU). APCI includes the start character (68h), the length of APDU and four Control Fields (CFs). On the other side, ASDU is an optional part which is determined by the format of APDU. In particular, APDU can take three formats: (a) I-format, (b) S-format and (c) U- format. The I-format is used to execute numbered information transfers and always includes ASDU. The S-format is used to perform numbered supervisory functions and comprises only APCI. Finally, the U-format is responsible for performing unnumbered control functions, and it also includes only APCI. The format of APDU is determined by CF1 and specifically by its two last bits. If the two last bits of CF1 are equal to 00, then the I-format is used. Accordingly, if the last bits of CF1 are equal to 01, then the S-format is applied. Finally, if the aforementioned bits are 11, the U-format is used. Concerning the ASDU, it includes the following fields: (a) Type Identification, (b) Structure Qualifier (SQ), (c) Number of Objects or Elements, (d) T, (e) P/N, (f) Cause of Transmission (CoT), (g) Originator Address (ORG), (h) ASDU Address, or Common Address of ASDU (CoA), (i) Information Object Address (IOA), (j) Information Elements and (k) Time Tag. The Type Identification determines the type of information objects. All information objects of an ASDU must have the same type. SQ specifies how the information objects and elements are structured. The Number of Objects or Elements field denotes the number of information objects or elements depending on the value of SQ. Accordingly, T defines those ASDUs which are dedicated for testing. P/N determines the positive or negative confirmation of an activation command. CoT directs ASDU to specific tasks and simultaneously interprets the data received by the destination side. ORG is an optional field and undertakes to explicitly define the identity of the controlling station (i.e., MTU). CoA defines the address of MTU or RTUs at the application layer. IOA determines the address of an information object. Information Elements provide and transmit specific information and finally, Time Tag provides time information.

IEC 60870-5-104 relies on the TCP/IP, which itself includes multiple security issues. Moreover, IEC 60870-5-104 does not include any authentication and authorisation mechanism, thus enabling potential MITM and unauthorised access attacks. In particular, the intrusion detection models related to this protocol rely on a threat model combining ADT and CVSS. Fig. 5.7 depicts the ADT of the proposed IEC 60870-5-104 threat analysis. The non-refined nodes of this threat analysis are considered as IEC 60870-5-104 cyberattacks supported by existing attacking tools, such as the Metasploit framework (i.e., auxiliary/client/iec104/iec104), Qtester104, OpenMUC j60870, IEC-TestServer and custom Ettercap filters. Therefore, the non-refined nodes are (a) MITM, (b) Traffic Sniffing, (c) C_RD_NA_1, (d) C_CI_NA_1, (e) C_RP_NA_1, (f) C_SC_NA_1, (g) C_SE_NA_1, (h) M_SP_NA_1_DOS, (i) C_CI_NA_1_DOS,

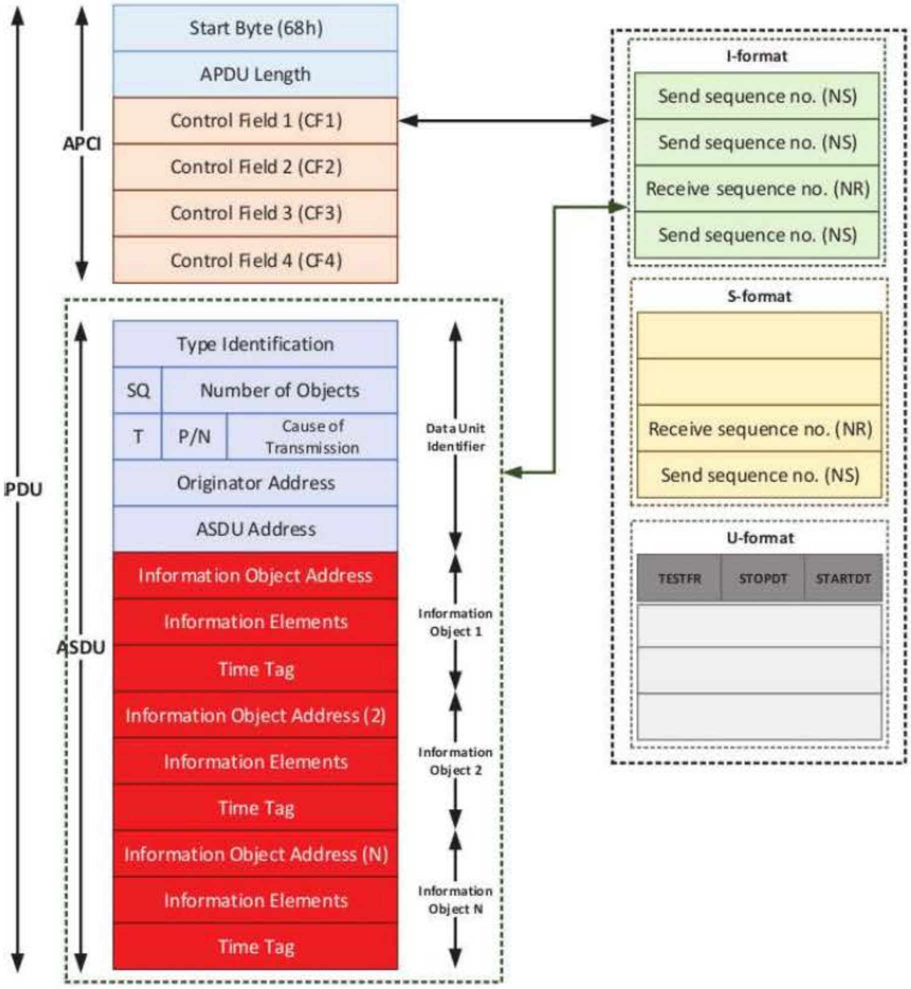


FIGURE 5.6: IEC 60870-5-104 Attributes

(j) C_SE_NA_1_DOS, (k) C_RD_NA_1_DOS and (l) C_RP_NA_1_DOS. The cyberattacks between (c) and (f) refer to unauthorised access cyberattacks related to the respective IEC 60870-5-104 commands. Similarly, the cyberattacks between (f) and (l) denote DoS cyberattacks corresponding to the IEC 60870-5-104 commands. Fig. 5.7 quantifies their severity based on CVSSv3.1. It should be noted that the Confidentiality Requirement (CR), the Integrity Requirement (IR) and the Availability Requirement (AR) of the Environmental Group are defined to "High" since the proposed threat model is adopted in a CI, so that the IEC 60870-5-104 communications should be secured as much as possible. The other CVSS values are determined based on the nature of each IEC 60870-5-104 command. Table 5.2 summarises the IEC 60870-5-104 cyberattacks, including their CVSS textual representations. Subsequently, the CVSS scores of the non-refined nodes are propagated upper by using the equation (5.2) and equation (5.3). Therefore, the CVSS scores of the refined nodes (i.e., (a) Compromising Confidentiality, (b) Compromising Integrity and (c) Compromising Availability) are calculated and illustrated by Fig. 5.7. Moreover, the proposed threat model considers two countermeasures called "Intrusion Detection" and "SDN-based Mitigation". The first node is responsible for the detection process, while the second undertakes to mitigate the intrusion through SDN.

Table 5.2: IEC 60870-5-104 Cyberattacks Description and CVSS Representation

IEC 60870-5-104 Cyberattack	Description	CVSS Representation
Man-In-the-Middle	During this attack, the cyberattacker is inserted between two endpoints, thus monitoring and controlling the network traffic exchanged.	AV:N/AC:L/PR:H/UI:R/S:C/C:H/I:L/A:L/E:H/RL:O/RC:C/MAV:N/MAC:L/MPR:H/MUI:R/MS:C/MC:H/MI:L/MA:L/CR:H/IR:H/AR:H
Capturing and Dropping IEC 60870-5-104 Packets	This attack is a refinement of the Man-In-The-Middle attack, where the cyberattacker can drop the IEC 60870-5-104 packets.	AV:N/AC:L/PR:H/UI:R/S:C/C:H/I:N/A:N/E:H/RL:O/RC:C/MAV:N/MAC:L/MPR:H/MUI:R/MS:C/MC:L/MI:N/MA:N/CR:H/IR:H/AR:H
Traffic Sniffing	Traffic Sniffing is a passive attack, where through the MITM the cyberattacker can monitor and capture the IEC 60870-5-104 packets.	AV:N/AC:L/PR:H/UI:R/S:C/C:H/I:N/A:N/E:H/RL:O/RC:C/MAV:N/MAC:L/MPR:H/MUI:R/MS:C/MC:L/MI:N/MA:N/CR:H/IR:H/AR:H
C_CI_NA_1	The C_CI_NA_1 is a Counter Interrogation command in the control direction. This cyberattack sends unauthorised IEC 60870-5-104 C_CI_NA_1 packets to the target system.	AV:N/AC:L/PR:H/UI:R/S:C/C:L/I:H/A:N/E:F/RL:T/RC:R/MAV:N/MAC:L/MPR:H/MUI:R/MS:C/MC:L/MI:H/MA:N/CR:H/IR:H/AR:H
C_SC_NA_1	The C_SC_NA_1 command is a single command. This cyberattack sends unauthorised C_SC_NA_1 60870-5-104 packets to the target system.	AV:N/AC:L/PR:H/UI:R/S:C/C:L/I:H/A:N/E:F/RL:T/RC:R/MAV:N/MAC:L/MPR:H/MUI:R/MS:C/MC:L/MI:H/MA:N/CR:H/IR:H/AR:H
C_SE_NA_1	The C_SE_NA_1 command is a set-point command with normalised values. This cyberattack sends unauthorised IEC 60870-5-104 C_SE_NA_1 packets to the target system.	AV:N/AC:L/PR:H/UI:R/S:C/C:L/I:H/A:N/E:F/RL:T/RC:R/MAV:N/MAC:L/MPR:H/MUI:R/MS:C/MC:L/MI:H/MA:N/CR:H/IR:H/AR:H
C_RD_NA_1	The C_RD_NA_1 command is a read command. This cyberattack sends unauthorised IEC 60870-5-104 C_RD_NA_1 packets to the target system.	AV:N/AC:L/PR:H/UI:R/S:C/C:L/I:H/A:N/E:F/RL:T/RC:R/MAV:N/MAC:L/MPR:H/MUI:R/MS:C/MC:L/MI:H/MA:N/CR:H/IR:H/AR:H
C_RP_NA_1	The C_RP_NA_1 command is a reset command. This cyberattack sends unauthorised IEC 60870-5-104 C_RP_NA_1 packets to the target system.	AV:N/AC:L/PR:H/UI:R/S:C/C:L/I:H/A:N/E:F/RL:T/RC:R/MAV:N/MAC:L/MPR:H/MUI:R/MS:C/MC:L/MI:H/MA:N/CR:H/IR:H/AR:H
M_SP_NA_1_DoS	This attack floods the target system with IEC 60870-5-104 M_SP_NA_1 packets.	AV:N/AC:H/PR:H/UI:R/S:C/C:N/N:L/A:H/E:F/RL:W/RC:R/MAV:N/MAC:H/MPR:H/MUI:R/MS:C/MC:N/MI:N/MA:H/CR:H/IR:H/AR:H
C_CI_NA_1_DoS	This attack floods the target system with IEC 60870-5-104 C_CI_NA_1 packets.	AV:N/AC:H/PR:H/UI:R/S:C/C:N/N:L/A:H/E:F/RL:W/RC:R/MAV:N/MAC:H/MPR:H/MUI:R/MS:C/MC:N/MI:N/MA:H/CR:H/IR:H/AR:H
C_SE_NA_1_DoS	This attack floods the target system with IEC 60870-5-104 C_SE_NA_1 packets.	AV:N/AC:H/PR:H/UI:R/S:C/C:N/N:L/A:H/E:F/RL:W/RC:R/MAV:N/MAC:H/MPR:H/MUI:R/MS:C/MC:N/MI:N/MA:H/CR:H/IR:H/AR:H
C_SC_NA_1_DoS	This attack floods the target system with IEC 60870-5-104 C_SC_NA_1 packets.	AV:N/AC:H/PR:H/UI:R/S:C/C:N/N:L/A:H/E:F/RL:W/RC:R/MAV:N/MAC:H/MPR:H/MUI:R/MS:C/MC:N/MI:N/MA:H/CR:H/IR:H/AR:H
C_RD_NA_1_DoS	This attack floods the target system with IEC 60870-5-104 C_RD_NA_1 packets.	AV:N/AC:H/PR:H/UI:R/S:C/C:N/N:L/A:H/E:F/RL:W/RC:R/MAV:N/MAC:H/MPR:H/MUI:R/MS:C/MC:N/MI:N/MA:H/CR:H/IR:H/AR:H
C_RP_NA_1_DoS	This attack floods the target system with IEC 60870-5-104 C_RP_NA_1 packets.	AV:N/AC:H/PR:H/UI:R/S:C/C:N/N:L/A:H/E:F/RL:W/RC:R/MAV:N/MAC:H/MPR:H/MUI:R/MS:C/MC:N/MI:N/MA:H/CR:H/IR:H/AR:H

Based on the aforementioned remarks, three intrusion and anomaly detection models were implemented, namely (a) IEC 60870-5-104 Intrusion Detection Model, (b) IEC 60870-5-104 TCP/IP Intrusion Detection Model and (c) IEC 60870-5-104 TCP/IP Anomaly Detection Model. According to the comparative study provided in the following chapter, the first one relies on a CART decision tree, using the IEC 60870-5-104 flow statistics of Appendix G. The second model adopts also a CART decision tree based on the TCP/IP flow statistics in Appendix E. Finally, the last model uses an isolation forest also with the TCP/IP flow statistics in Appendix E. The first two models can recognise the above IEC 60870-5-104 cyberattacks, while the last model can discriminate the presence of an anomaly. Regarding the training procedure of the previous ML intrusion and anomaly detection models, the IEC 60870-5-104 Intrusion Detection Dataset was utilised. This data was published in IEEE Dataport and Zenodo in the context of this PhD programme.

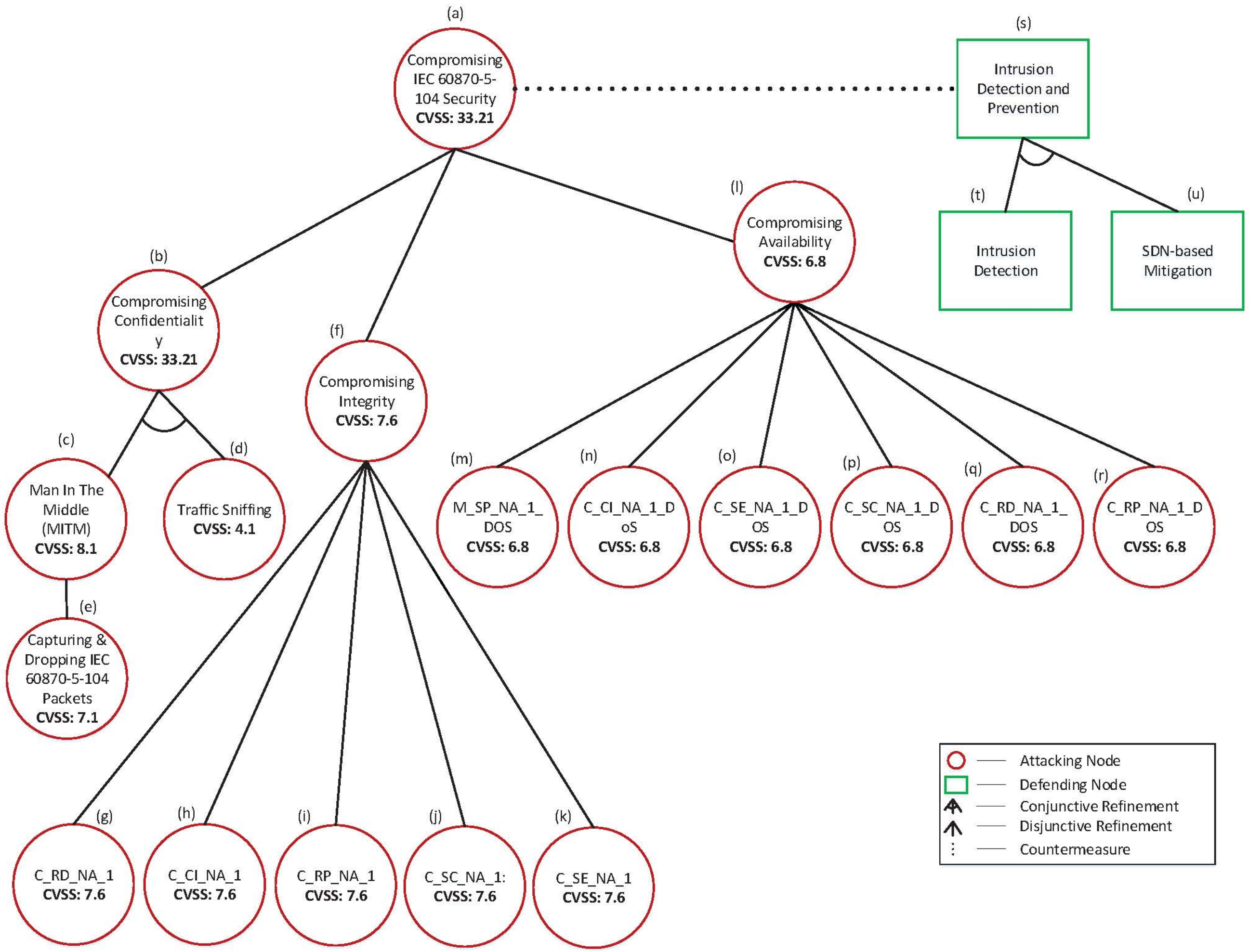


FIGURE 5.7: IEC 60870-5-104 ADT

5.2.4 IEC 61850 Intrusion Detection Models

IEC 61850 is an international communication standard for electrical substation environments, defining a hierarchical, object-oriented data representation model. In particular, each IIoT asset is characterised by a data model composed of naming, diagnostic and configuration information. The purpose of this data model is to facilitate the information exchange among the IIoT assets without referring to their functional and technical details. The IEC 61850 stack consists of four types of messages: (a) MMS, (b) Generic Substation State Events (GSSE), (c) GOOSE and (d) SMV. In this thesis, a special emphasis is given to GOOSE and MMS. In particular, the GOOSE Intrusion Detection Model was implemented, using Random Forest, the GOOSE flow statistics in Appendix L and the dataset provided P.P. Biswas et al. in [25]. In particular, the following GOOSE-related cyberattacks can be detected by the GOOSE Intrusion Detection Model.

- **GOOSE DoS**: This refers to a GOOSE-related DoS attack, which floods the target system with GOOSE messages, to block legitimate IEDs from accessing resources.
- **Data Manipulation**: This is an unauthorised access attack, which injects malicious GOOSE packets, aiming to impact the grid stability or to cover unauthorized changes.
- **Message Suppression**: This is an unauthorised access attack, which injects malicious GOOSE packets, aiming to impact the grid stability or to cover unauthorised changes.
- **Disturbance**: It refers to electricity-related disturbances and faults that might occur.

On the other side, the MMS TCP/IP Anomaly Detection Model was generated, using MCD, the TCP/IP flow statistics in Appendix E, and a custom MMS anomaly detection dataset, which was generated in the context of the H2020 SPEAR project [142].

5.2.5 HTTP Intrusion Detection Models

The NF-IDPS includes two models related to HTTP, namely: (a) HTTP TCP/IP Intrusion Detection Model and (b) HTTP TCP/IP Anomaly Detection Model. The first model uses a decision tree, which can detect the following HTTP-related cyberattacks.

- **DoS**: This DoS attack floods the target system with HTTP packets.
- **SQL-Injection**: This attack aims to exploit vulnerabilities of web applications in order to access unauthorised information.
- **Bruteforce-Web**: This attack attempts to access a password-protected web application by using multiple password combinations.

- **XSS**: XSS is a type of injection attack where malicious scripts are injected into web applications.

On the other side, the HTTP Anomaly Detection Model adopts LOF. The training procedure of both models relies on the TCP/IP flow statistics in Appendix E and the CSE-CIC-IDS 2018 dataset [51].

5.2.6 SSH Intrusion Detection Models

Finally, two SSH-related models are involved in the IDE of NF-IDPS, namely (a) SSH TCP/IP Intrusion Detection Model and (b) SSH TCP/IP Anomaly Detection Model. The first one uses Adaboost to recognise SSH bruteforce attacks, while the second model is based on MCD to detect anomalous SSH network flows. Both models take as input the TCP/IP network flow statistics in Appendix E, while the CSE-CIC-IDS 2018 dataset was utilised for the training procedure.

5.3 H-IDPS: Host-based Intrusion Detection Prevention and System

The proposed H-IDPS includes four detection models that rely on operational data related to the energy sector (i.e., time series electricity measurements). In particular, these data summarised in Appendices H-K are related to four IIoT/SG use cases from the H2020 SPEAR project [142]: (a) hydropower plant, (b) substation, (c) power plant and (d) smart home. All the detection models for the previous use cases use the smArt gRid Intrusion dEtection System (ARIES) GAN [140]. The aim of an adversarial network concerning the problem of the anomaly detection is to train an unsupervised network, which will be capable of recognising anomalies, using a dataset, which includes data of a single class. In particular, this data is used only for the training process and denotes a benign behaviour. Supposing two datasets: (a) a training dataset $D = \{X_1, ..., X_M\}$, which contains M normal occurrences and (b) a testing dataset $\hat{D} = \{(\hat{X}_1, y_1), ..., (\hat{X}_N, y_N)\}$ which includes N both normal and abnormal occurrences and $y_i \in [0, 1]$ denotes the label of each occurrence. It is worth noting that $M \gg N$.

The goal is to model appropriately D to understand the manifold representation and then to recognise anomalies in $\hat{D}$. In particular, the model f learns the normal data distribution and produces an anomaly score $A(x)$. A high value of $A(x)$ denotes a potential anomaly for the specific data point. More precisely, a threshold value T determines whether $A(x)$ indicates an anomaly or not if $A(x) > t$. T is defined experimentally, utilising a testing dataset.

Fig. 5.8 depicts ARIES GAN architecture, which is composed of two sub-networks: (a) generator and (b) discriminator. The generator receives the input $z = \{x(t), R\}$ representation that includes the real data $x(t)$ at the current time t and a noise vector R. The output x' is the reconstruction of the input data for the current time t and all the previous N instances. In particular, the generator G first reads the input z, where $z \in \Re^{w2}$, and forwards it to the encoder network E. Based on fully connected layers followed by

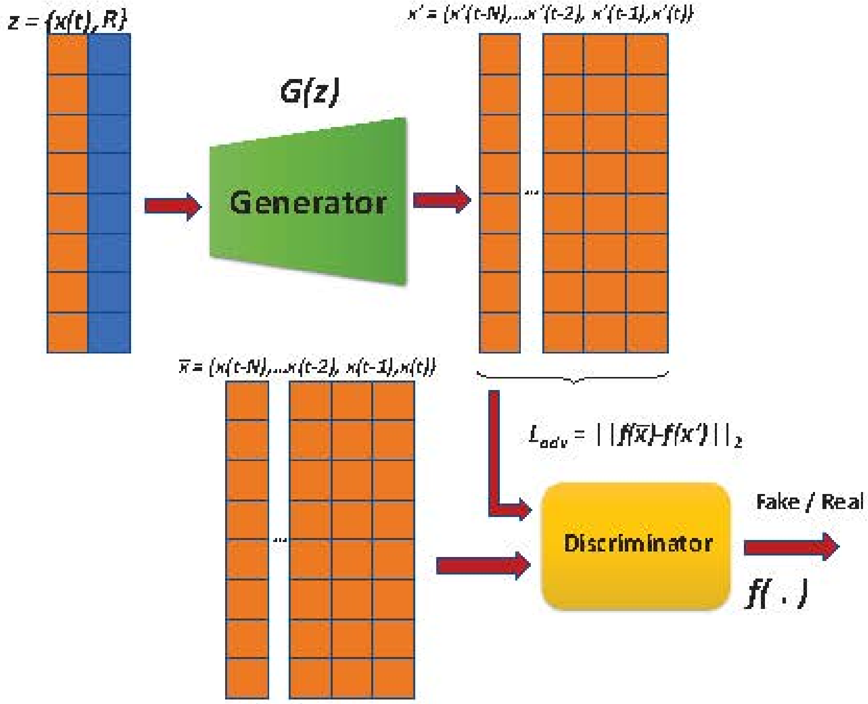

z = {x(t), R}
G(z)
Generator
x' = (x'(t-N),...x'(t-2), x'(t-1),x'(t))
...
x̄ = (x(t-N),...x(t-2), x(t-1),x(t))
...
L_adv = ||f(x̄)-f(x')||_2
Discriminator
Fake / Real
f(.)

FIGURE 5.8: ARIES GAN Architecture

a batch-norm and leaky $ReLU()$ activation function, G regresses z to x'. Consequently, the generator G produces the data x' via $x' = G(z)$, where $z = \{x(t), R\}$. On the other side, the discriminator intends to distinguish the input $\bar{x}$ and the output x' as real or fake, respectively. It consists of fully connected layers followed by batch-norm and leaky ReLU activation.

Supposing that abnormal data points are forward-passed into the network G; however, since the generator is modelled only with normal samples during the training, it fails to reconstruct the abnormalities in the previous N time instances. In this thesis, abnormal data does not occur in single-time instances. An output x' that has not taken into account anomalies can result in the encoder network E corresponding x' to a vector z', which also has not considered an anomalous feature representation, thereby creating a dissimilarity between z and z'. In such a case (i.e., when there is a dissimilarity within the latent vector space for an input signal $x(t)$), the model categorizes x as an anomaly.

Regarding the training process of this network, the loss function was selected, considering the feature matching loss as illustrated in Equation 5.4.

$$L_{adv} = \|f(\bar{x}) - f(x')\|_2 \tag{5.4}$$

In particular, f is a function, which outcomes an intermediate layer of the discriminator D based on a given input $\bar{x}$; feature matching calculates the L_2 distance between the feature representation of the real and the produced data points.

5.4 V-IDPS: Visual-based Intrusion Detection and Prevention System

The goal behind the proposed V-IDPS is also to detect and mitigate timely the Modbus/TCP cyberattacks discussed above. However, its functionality can also be utilised in a similar manner with other industrial protocols and cyberattacks. As illustrated in Fig. 5.9, the architecture of the proposed IDPS consists of five modules: (a) Network Traffic Monitoring and Capturing Module (NTMCM), (b) Network Flow Extraction and Clustering Module (NFECM), (c) Visual Representation Generation Module (VRGM), (d) IDE and (e) NM. The first module is responsible for monitoring and capturing the entire Modbus/TCP network traffic. To this end, SPAN and Tcpdump are utilised. NFECM receives the overall Modbus/TCP network traffic as an overall pcap file and discriminates the bidirectional Modbus/TCP network flows, generating the corresponding pcap files. A network flow is characterised by four elements: (a) source IP address, (b) destination IP address, (c) source TCP/UDP port and (d) destination TCP/UDP port. Thus, each pcap file generated by NFECM includes the Modbus/TCP packets of a specific Modbus/TCP network flow. For this purpose, PcapPlusPlus-PcapSplitter is used. Next, VRGM uses Binvis in order to convert each pcap file related to the Modbus/TCP network flows into visual representations. More details about this conversion are provided below. Subsequently, IDE adopts an Active ResNet50-based CNN, which receives the visual representations and classifies them into the aforementioned Modbus/TCP cyberattacks. Accordingly, more information about the operation of the proposed Active ResNet50-based CNN is given below. Finally, NM sends the security logs of V-IDPS to NCME.

The IDE combines two detection layers that work in a complementary manner. The first layer constitutes a binary visualisation mechanism that supports the security administrator to distinguish manually the Modbus/TCP threats. On the other side, the second layer applies an Active ResNet50-based CNN in order to classify the Modbus/TCP network flows automatically. Both layers work together for the accurate detection of the Modbus/TCP cyberattacks. In particular, the first layer constitutes a verification method through which the security administrator can oversee the detection results of the second layer. Moreover, it is worth mentioning that the first layer contributes to the re-training process of the Active ResNet50-based CNN. The following subsections provide more details for each detection layer, respectively.

5.4.1 Binary Visualisation

The proposed IDPS adopts Binvis in order to transform the pcap files reflecting the corresponding Modbus/TCP network flows into understandable visual representations (i.e., images) utilised by the security administrator to discriminate the aforementioned Modbus/TCP threats. Binvis relies on the Python library scurve, which transforms binary files into various curve representations. In particular, each byte of the pcap files is translated into a pixel, utilising the following colour scheme of scurve: (a) Black: 00, (b) White: FF, (c) Blue: printable characters and (d) Red: everything else. Thus, each

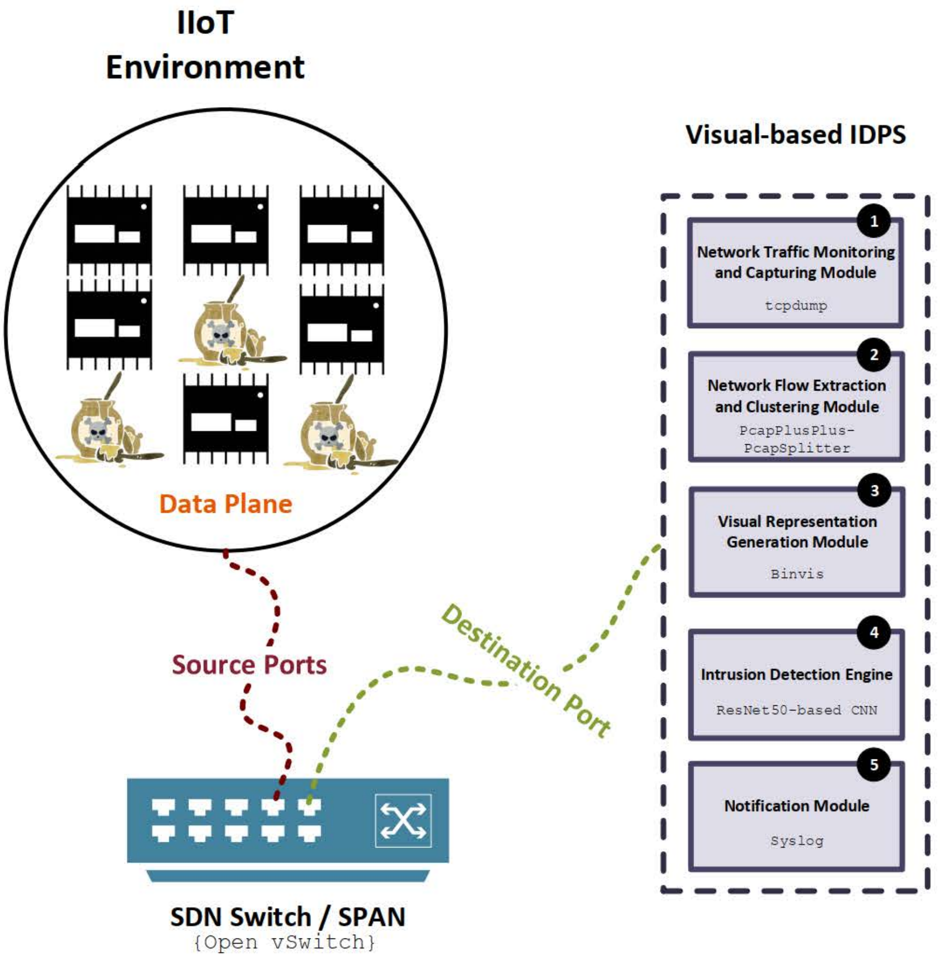

IIoT Environment
Data Plane
Source Ports
Destination Port
SDN Switch / SPAN
{Open vSwitch}
Visual-based IDPS
1
Network Traffic Monitoring and Capturing Module
tcpdump
2
Network Flow Extraction and Clustering Module
PcapPlusPlus-PcapSplitter
3
Visual Representation Generation Module
Binvis
4
Intrusion Detection Engine
ResNet50-based CNN
5
Notification Module
Syslog

FIGURE 5.9: Architecture of V-IDPS

pixel is placed on the two-dimensional visual representation, taking into account the locality of the binary elements. The binary elements being close in the pcap files should be placed as near as possible on the two-dimensional representation. To this end, Hilbert Curve is used to arrange the pixels in the image. The Hilbert Curve belongs to the family of the recursive Space-Filling Curves (SFCs) that divide a space into several segments, visiting the segments with a particular order. SFCs, also known as Peano curves, project the data from one-dimensional space into an n-dimensional space by preserving the properties of the original data. M. Wattenberg in [202] describes the relationship between the space-filling visualisation and the mathematics behind SFCs. In particular, four properties are preserved: (a) stability, (b) split neutrality, (c) order adjacency and (d) locality. The range of SFC covers the two-dimensional unit square and, in general, an n-dimensional unit hypercube; however, in this thesis, the two-dimensional space is used since the output of Binvis is a two-dimensional visual representation. Thus, a two-dimensional unit square refers to a visual representation of $n \times n$ pixels, and the Hilbert curve represents a continuous curve for each unit square (i.e., pixel of the image). Although G. Peano was the first who defined and discovered the first SFC, D. Hilbert was the one who identified a geometrical process that allows the generation of an entire class of SFCs. D. Hilbert defined that each t belonging to an interval $I = [0, 1]$ is determined by a sequence of nested closed intervals that are generated by a successive partitioning. This sequence corresponds to a sequence of nested closed squares whose diagonals shrink into a point, determining a unique point in $Q = [0, 1]^2$ which is the image $f_h(t)$ of t. $f_{h*}(I)$ is called Hilbert Curve [161]. H. Sagan in [161] provides a detailed analysis about various SFCs, including the Hilbert curve.

Fig 5.10 depicts how the Hilbert curve is utilised for transforming one-dimensional data (i.e., pcap binary file) into a two-dimensional visual representation. First, each byte of the binary pcap file is transformed into a particular colour based on the colour scheme of scurve. Then, the Hilbert curve is applied in order to map the one-dimensional data into a two-dimensional visual representation. Similarly, Fig 5.11 shows the Binvis visualisations for each pcap file corresponding to the malicious network flows of the aforementioned Modbus/TCP threats. Although the Binvis visualisations are similar to each other, a granular inspection can distinguish the differences, thus identifying the Modbus/TCP threats discussed previously.

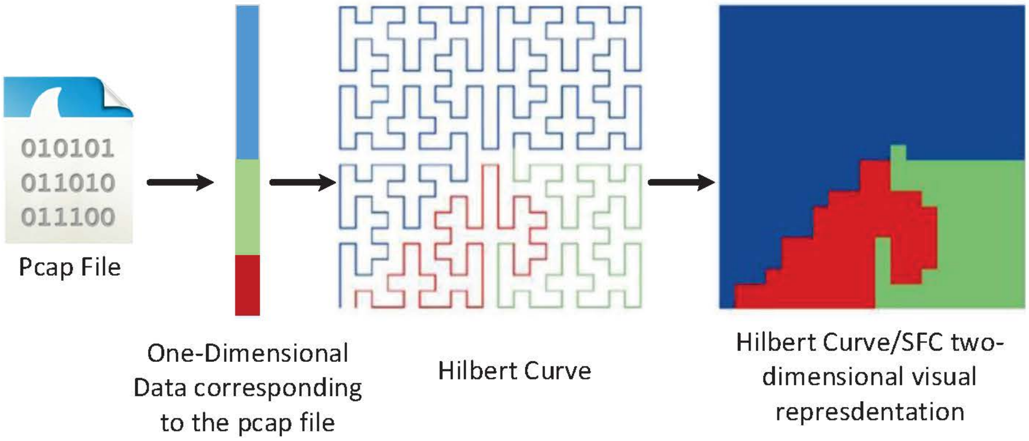
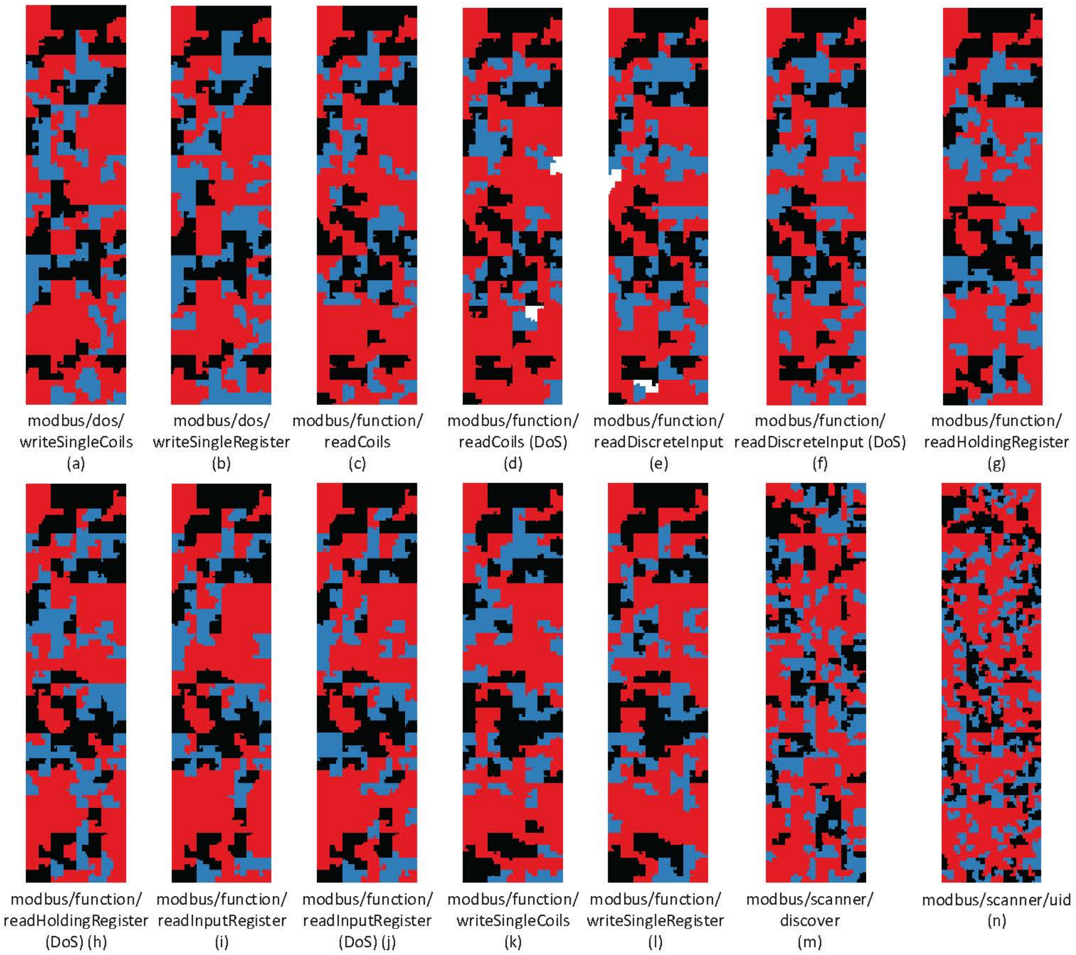

010101
011010
011100
Pcap File
One-Dimensional Data corresponding to the pcap file
Hilbert Curve
Hilbert Curve/SFC two-dimensional visual represdentation

FIGURE 5.10: Transformation of a binary pcap file into a Hilbert curve two-dimensional visual representation

5.4.2 Active ResNet50-based CNN Detection

Although the first detection layer provides an adequate manner for discriminating the Modbus/TCP threats, it constitutes a manual solution, not applicable for a large number of Modbus/TCP network flows. The binary visualisation can be utilised only as an additional detection mechanism verifying or correcting the outcomes of automatic means. The second layer of the proposed IDPS adopts a CNN, which combines Transfer Learning and Active Learning in order to classify the pcap visual representations of the Modbus/TCP network flows into the Modbus/TCP threats automatically. Both Transfer Learning and Active Learning are adopted when there are not available datasets or a sufficient amount of data, as in our case, since IIoT/SG environments like CIs cannot disclose and share their sensitive data. On the one side, Transfer Learning refers to when an ML/DL model pre-trained for another task is used to solve a problem from another domain [122]. This approach is applied widely to the CNN models. In particular, the new CNN uses some weights of a pre-trained CNN, which has been trained on



modbus/dos/ writeSingleCoils (a)

modbus/dos/ writeSingleRegister (b)

modbus/function/ readCoils (c)

modbus/function/ readCoils (DoS) (d)

modbus/function/ readDiscreteInput (e)

modbus/function/ readDiscreteInput (DoS) (f)

modbus/function/ readHoldingRegister (g)

modbus/function/ readHoldingRegister (DoS) (h)

modbus/function/ readInputRegister (i)

modbus/function/ readInputRegister (DoS) (j)

modbus/function/ writeSingleCoils (k)

modbus/function/ writeSingleRegister (l)

modbus/scanner/ discover (m)

modbus/scanner/uid (n)

FIGURE 5.11: Visual representation of the pcap files corresponding to the malicious network flows of the Modbus/TCP threats

a large-scale dataset like ImageNet [40]. Usually, from the pre-trained CNN, the final fully-connected layers are removed. Next, a concise training process follows to adjust the remaining parts of the new CNN corresponding to the fully connected layers. Multiple pre-trained CNNs have already demonstrated their efficiency, using the ImageNet dataset, which involves 1.2 million images. Characteristic examples are VGG16, VGG19, ResNet50, Xception, MobileNet, DenseNet121 and EfficientNetB0. Based on a comparative analysis described in the following chapter, the proposed IDPS uses ResNet50 [61].

More specifically, Fig. 5.12 shows the CNN architecture behind the second detection layer of the proposed IDPS. First, ResNet50 is utilised, and then a sequence of a `Flatten` layer and 5 Dense layers follow with 1024, 512, 256, 128 and 15 neurons, respectively. Apart from the last Dense layer, the remaining ones use the ReLu activation function given by Equation 5.5. The last `Dense` layer uses the Softmax function, given by Equation 5.6. ResNet50 is inspired by VGG19, utilising 34-layer plain network architecture in which shortcut connections are added, thus leading to the residual network illustrated by Fig. 5.12. The colour scheme denotes the number of the filters with respect to the convolutional layers. More information about ResNet50 is given in [61]. The training process uses the Categorical Cross-Entropy function (Equation 5.7) and the Adam optimiser.

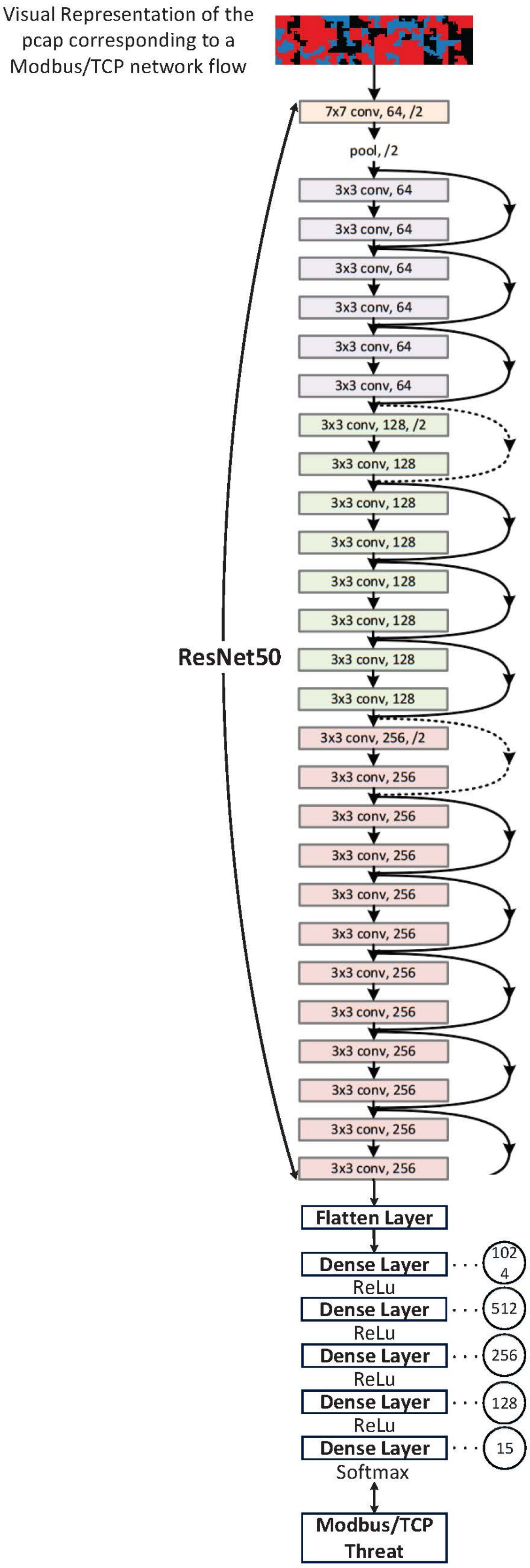


FIGURE 5.12: Active ResNet-based CNN architecture

$$f(x) = \begin{cases} 0, & \text{for } x \leq 0 \\ x, & \text{for } x \geq 0 \end{cases} \tag{5.5}$$

$$softmax(z)_i = \frac{e^{z_i}}{\sum_{j,n} e^{z_j}} \tag{5.6}$$

$$L_{cc}(r,p) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(r_{ij} \times log(p_{ij})) \tag{5.7}$$

Although the ResNet50-based CNN constitutes an initial and efficient model for detecting and classifying the Modbus/TCP threats, its performance relies on the available training data (i.e., pcap files reflecting malicious Modbus/TCP threats.) However, such data is rarely available. Even if there are some synthesised datasets, the Modbus/TCP threats and their consequences can differ from one IIoT/SG environment to another. Therefore, the proposed IDPS adopts an Active Learning approach, which makes IDE capable of re-training itself. Active Learning composes a functional framework, which allows the selection of the most informative data samples from an unlabelled dataset, thus creating or enhancing the training dataset, leading, in our case, to a more accurate multi-class classification model [94]. In Active Learning, the classifier is called $Hypothesis$. Unlike Passive Learning, which selects the data samples randomly, Active Learning follows particular criteria leading to represented and representative data samples providing more accurate results [153]. Usually, an external factor called $Oracle$ assesses and annotates the data samples selected by the Active Learning methods [170]. In our case, IDE and particularly the ResNet50 CNN represents the $Hypothesis$, while the system administrator plays the role of the $Oracle$, utilising the Binvis representations. Fig. 5.13 illustrates the Active Learning procedure behind the proposed IDPS. In the first step, the pooling-based sampling method is adopted in order to create a pool with the unlabelled data. Next, a query strategy is used to decide which data samples from the pool will be labelled by $Oracle$ and added to the new training dataset. With respect to the query strategy, the Uncertainty Sampling technique is used, based on the uncertainty of the $Hypothesis$. In other words, the Uncertainty Sampling selects those binary representations for which the Active ResNet50-based CNN is less confident. Subsequently, the $Hypothesis$ is fed with the unlabelled data selected in the previous step. Next, the $Hypothesis$ predicts the labels of this data. The prediction outcome of the ResNet50-based CNN can be assessed by the security administrator based on the binary visualisation of the first detection layer. Suppose the security administrator agrees with the decision of the ResNet50-based CNN. In that case, this data sample (i.e., the visual representation corresponding to the pcap file of the malicious Modbus/TCP network flow) is added to the new training dataset. Otherwise, $Oracle$ will correct the decision of $Hypothesis$, and the data sample is added to the new training dataset. Finally, the new training dataset is used to re-train the ResNet50-based CNN, thus converting it into an Active ResNet50-based CNN.

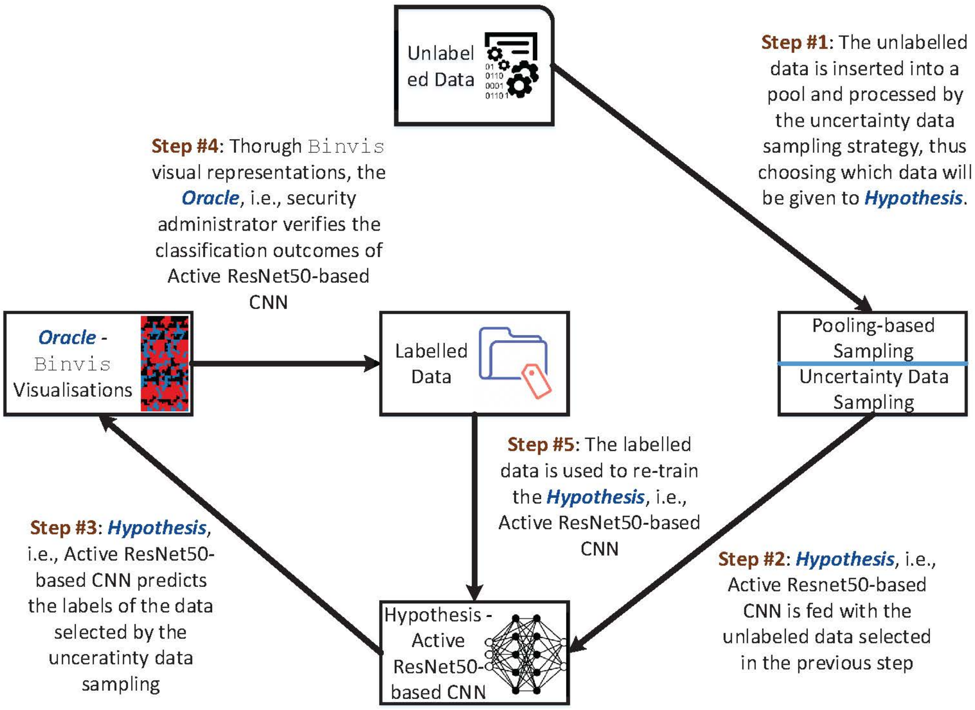

Unlabel ed Data
01 0110 0001 01101
Step #1: The unlabelled data is inserted into a pool and processed by the uncertainty data sampling strategy, thus choosing which data will be given to Hypothesis.
Step #4: Thorugh Binvis visual representations, the Oracle, i.e., security administrator verifies the classification outcomes of Active ResNet50-based CNN
Oracle - Binvis Visualisations
Labelled Data
Pooling-based Sampling
Uncertainty Data Sampling
Step #5: The labelled data is used to re-train the Hypothesis, i.e., Active ResNet50-based CNN
Step #3: Hypothesis, i.e., Active ResNet50-based CNN predicts the labels of the data selected by the unceratinty data sampling
Step #2: Hypothesis, i.e., Active Resnet50-based CNN is fed with the unlabeled data selected in the previous step
Hypothesis - Active ResNet50-based CNN

Figure 5.13: Proposed Active Learning Procedure

Suppose the visual representations corresponding to the Mobuds/TCP network flows from an IIoT/SG environment are generated continuously. Let x be an unlabelled visual representation from the input space X and y the respective label related to the Modbus/TCP threats discussed earlier, comprising also the normal state. Furthermore, U denotes a set of unlabelled visual representations within the pool, while L indicates the new training dataset, which will be used to re-train IDE. Therefore, on the one hand, the function $f(x) = y$ is the target function that discriminates and classifies the visual representations accurately without any functional error. On the other hand, the function $h(x) = y'$ represents the Active ResNet50-based CNN predicting the label of the visual representation. Consequently, the goal is to minimise the generalisation error defined by Equation 5.8.

$$E[l(h)] \int_{-\infty}^{\infty} l(h(x), f(x))\, dx \tag{5.8}$$

where l is the squared error function defined by Equation 5.9.

$$l(h(x), f(x)) = (h(x) - f(x))^2 \tag{5.9}$$

Therefore, the Active Learning problem lies in labelling correctly and selecting the appropriate visual representations from U, thus composing and enhancing a new training dataset L that will re-train the Active ResNet50 CNN ($Hypothesis$) and will optimise its detection efficiency. The labelling process is conducted by the $Hypothesis$ itself and is validated by the security administrator through the binary visualisation. To identify the suitable visual representations in U, Uncertainty Sampling is used. The $Hypothesis$' uncertainty can be calculated with various criteria: (a) entropy, (b) least confidence of prediction and (c) least margin. In this thesis, entropy defined by Equation 5.10 is used.

$$H = -\sum_{i=1}^{m} p_\theta(y_i|x)\log_2(p_\theta(y_i|x)) \tag{5.10}$$

where p_θ denotes the probability of class i for the visual representation x, while θ implies the parameters of the $Hypothesis$. Therefore, the entropy criterion chooses the visual representations x^* from U that fulfil the Equation 5.11. In this paper, δ is determined experimentally.

$$x^* = argmax(x) + H > \delta \tag{5.11}$$

Based on the above remarks, Algorithm 1 illustrates the Active Learning process of the Active ResNet50-based CNN. First, the $Hypothesis$ $h(X)$ is trained with an initial dataset L comprising a few data samples. To this end, a Modbus/TCP intrusion detection dataset was constructed by emulating the aforementioned Modbus/TCP threats. Next, U is filled in continuously with new visual representations. While the size of U is greater than 0, $h(x)$ classifies each visual representation within U. The security administrator verifies this process through the visual representations. As depicted in Fig. 5.11, although the visual representations of the Modbus/TCP threats present common characteristics, they constitute an adequate manner for discriminating the Modbus/TCP threats manually. Next, the uncertainty of $h(x)$ is calculated. If the entropy criterion is satisfied, then the corresponding visual representation of U is moved in L. Next, when the size of L reaches a new threshold t, the re-training process is applied.

Algorithm 1: Active ResNet50-based CNN: Pooling-based Sampling and Uncertainty Sampling Strategy

```
Data: U, L, h
Result: Re-train h
Train h;
while size(U) $> 0$ do
    if uncertainty(h(U(i))) $> \delta$ then
        h predicts y(i);
        The security administrator verifies the prediction of h;
        Add U(i) and y(i) in L;
        Re-train h
    end
    if size(L) $==$ $t$ then
        Re-train h;
        Clear U;
    end
end
```

5.5 NCME: Normalisation, Correlation and Mitigation Engine

The role of NCME is twofold, first, to receive, normalise and correlate the security events from the previous IDPS and second, to indicate potential mitigation actions. First, the normalisation process relies on the AlienVault OSSIM format, which is provided by (Table 5.3). Next, the correlation capacity of NCME is based on correlation rules that focus on the Modbus/TCP-related security events. However, similar rules can be used for other industrial communication protocols. This kind of correlation aims to identify relationships between the Modbus/TCP security events, composing alerts that reflect multi-step attack scenarios against Modbus/TCP. The correlation rules are constructed by combining the information of the security events and additional fields, such as time information (e.g., a sequence of events appearing in a specific period of time) or the number of continuous security events. Event Processing Language (EPL) statements are utilised for the syntax of these correlation rules. The following table summarises these rules. On the other side, the mitigation actions lie in two main categories: (a) SDN-based mitigation and (b) deployment of a suitable number of honeypots for enhancing the resilience of the IIoT/SG infrastructure in terms of hiding and protecting the actual IIoT/SG entities/devices. More information for each category is provided in the following subsections.

Table 5.3: OSSIM Security Event Format

Security Event Field Name	**Security Event Field Description**
Date	Date and time of the security event.
Sensor	The sensor, which processed the security event.
Device IP	The IP address of the sensor, which processed the security event.
Event Type ID	Identifier assigned by the component, which generates the security event.
Unique Event ID	Unique identifier assigned by the component, which generates the security event.
Protocol	Protocol related to the security event.
Category	Event taxonomy for the security event.
Subcategory	Subcategory of the security event taxonomy type listed under Category.
Data Source Name	Name of the external application or device that produced the security event.
Data Source ID	Identifier related to the external application or device which generated the security event.
Product Type	Product type related to the security event.
Additional Info	URL including more details about the security event.
Priority	It reflects the significance of the security event in the range between 0-5.

Reliability	It reflects the detection reliability in the range between 0-10.
Risk	Risk calculation relies on the formula: _Asset Value * Event Reliability * Event Priority / 25_
OTX Indicators	Number of indicators related to an Open Threat Intelligence (OTX) IP reputation or OTX pulse event.
Source/Destination ID	Identifier of the source/destination related to the security event.
Source/Destination IP	IP addresses of source/destination, respectively related to security event.
Source/Destination Hostname	Hostname of source/destination.
Source/Destination MAC Address	Media Access Control (MAC) of source/destination.
Source/Destination Port	Port of source/destination.
Source/Destination Latest Update	The last time when the component, which generated the security event updated the source/destination properties.
Source/Destination Username and Domain	Username and domain related to source/destination.
Source/Destination Asset Value	Asset value of source/destination. It reflects the significance of source/destination.
Source/Destination Location	If the origin of source/destination is known, it reflects the host country.
Source/Destination Context	If the asset belongs to a user-defined group of entities, AlienVault OSSIM shows the relevant contexts.
Source/Destination Asset Groups	When the source/destination belongs to one or more asset groups, this field lists the asset group name or names.
Source/Destination Networks	When the source/destination belongs to one or more networks, this field lists the networks.
Source/Destination Logged Users	A list of users and their information related to source/destination.
Source/Destination OTX IP Reputation	(Yes or No) Whether or not the OTX IP Reputation identifies the IP address as suspicious.
Source/Destination Service	List of services or applications related to the source/destination ports.
Service Port	Port utilised by the service or application.
Service Protocol	Protocol utilised by the service or application.
Raw Log	Raw log details of the security event.
Filename	Name of a file related to the security event.
Username	Usernames related to the security event.
Password	Passwords related to the security event.
Userdata 1-9	User-generated log fields.

Rule Detection	AlienVault OSSIM NIDS rule used to detect the security event.

TABLE 5.4: Security Correlation Rules for Modbus/TCP

No	**Description**
Rule #1	If there are X or more consecutive events denoting a modbus/function/ readInputRegister (DoS) attack, then an alert called 'modbus/function/ readInputRegister (DoS)' is raised. X is defined by the user.
Rule #2	If there are X or more consecutive events denoting a modbus/dos /writeSingleRegister attack, then an alert called 'modbus/dos /writeSingleRegister' is raised. X is defined by the user.
Rule #3	If there are X or more consecutive events denoting a modbus/function/ readDiscreteInputs (DoS) attack, then an alert called 'modbus/function/readDiscreteInputs (DoS)' is raised. X is defined by the user.
Rule #4	If there are X or more consecutive events denoting a modbus/ function/readHoldingRegister (DoS) attack, then an alert called 'modbus/function/readHoldingRegister (DoS)' is raised. X is defined by the user.
Rule #5	If there are X or more consecutive events denoting a modbus /function/readCoils (DoS) attack, then an alert called 'modbus/function/readCoils (DoS))' is raised. X is defined by the user.
Rule #6	If there are X or more consecutive events denoting a modbus/dos /writeSingleCoils attack, then an alert called 'modbus/dos /writeSingleCoils' is raised. X is defined by the user.
Rule #7	If there are X events denoting a modbus/scanner/uid attack and right after X events denoting a modbus/scanner/getfunc, then an alert called 'Modbus Reconnaissance'. X is defined by the user.
Rule #8	If there are X or more consecutive events denoting a modbus/ scanner/getfunc attack, then an alert called 'Modbus Reconnaissance' is raised. X is defined by the user.
Rule #9	If there are X or more consecutive events denoting a modbus/scanner /uid attack, then an alert called 'Modbus Reconnaissance' is raised. X is defined by the user.

Rule #10	If there are X events denoting a modbus/scanner/uid attack and right after X events denoting a modbus/function/writeSingleCoils, then an alert called ‘modbus/function/writeSingleCoils’ is raised. X is defined by the user.
Rule #11	If there are X events denoting a modbus/scanner/getfunc attack and right after X events denoting a modbus/function/writeSingleCoils, then an alert called ‘modbus/function/writeSingleCoils’ is raised. X is defined by the user.
Rule #12	If there are X or more consecutive events denoting a modbus/function /writeSingleCoils, then an alert called ‘modbus/function/ writeSingleCoils’ is raised. X is defined by the user.
Rule #13	If there are X events denoting a modbus/scanner/uid attack and right after X events denoting a modbus/function/readInputRegister, then an alert called ‘modbus/function/readInputRegister’ is raised. X is defined by the user.
Rule #14	If there are X events denoting a modbus/scanner/getfunc attack and right after X events denoting a modbus/function/readInputRegister, then an alert called ‘modbus/function/readInputRegister’ is raised. X is defined by the user.
Rule #15	If there are X or more consecutive events denoting a modbus/function /readInputRegister, then an alert called ‘modbus/function/ readInputRegister’ is raised. X is defined by the user.
Rule #16	If there are X events denoting a modbus/scanner/uid attack and right after X events denoting a modbus/function/writeSingleRegister, then an alert called ‘modbus/function/writeSingleRegister’ is raised. X is defined by the user.
Rule #17	If there are X events denoting a modbus/scanner/getfunc attack and right after X events denoting a modbus/function/writeSingleRegister, then an alert called ‘modbus/function/writeSingleRegister’ is raised. X is defined by the user.
Rule #18	If there are X or more consecutive events denoting a modbus/function /writeSingleRegister, then an alert called ‘modbus/function /writeSingleRegister’ is raised. X is defined by the user.
Rule #19	If there are X events denoting a modbus/scanner/uid attack and right after X events denoting a modbus/function/readDiscreteInput, then an alert called ‘modbus/function/readDiscreteInput’ is raised. X is defined by the user.

Rule #20	If there are X events denoting a modbus/scanner/getfunc attack and right after X events denoting a modbus/function/readDiscreteInput, then an alert called 'modbus/function/readDiscreteInput' is raised. X is defined by the user.
Rule #21	If there are X or more consecutive events denoting a modbus/function /readDiscreteInput, then an alert called 'modbus/function /readDiscreteInput' is raised. X is defined by the user.
Rule #22	If there are X events denoting a modbus/scanner/uid attack and right after X events denoting a modbus/function/readHoldingRegister, then an alert called 'modbus/function/readHoldingRegister' is raised. X is defined by the user.
Rule #23	If there are X events denoting a modbus/scanner/getfunc attack and right after X events denoting a modbus/function/readHoldingRegister, then an alert called 'modbus/function/readHoldingRegister' is raised. X is defined by the user.
Rule #24	If there are X or more consecutive events denoting a modbus/function /readHoldingRegister, then an alert called 'modbus/function /readHoldingRegister' is raised. X is defined by the user.

5.5.1 SDN-based Mitigation

In this thesis, SDN plays the role of a mitigation mechanism that can drop or re-arrange the malicious Modbus/TCP network flows. In contrast to typical IPS and traditional firewall systems, SDN represents a more reliable mitigation mechanism with respect to a massive amount of alerts. In particular, T. Xing et al. in [206] study and compare the reliability of an SDN-based IDPS against a plethora of malicious packets related to a DoS attack. They demonstrate that the proposed SDN-based IDPS can handle more efficiently the malicious packets rather than a typical IPS using iptables. The latter could not handle and process all the relevant packets of the DoS attack. Finally, in this thesis, instead of corrupting the malicious network flows directly, it is further examined whether this action (i.e., dropping malicious Modbus/TCP network flows) could generate more destructive effects, taking into account the sensitive nature of an IIoT/SG environment.

If NCME takes a decision to drop automatically the malicious network flows, then NCME does not use OpenFlow directly, but it takes full advantage of the Ryu REST API in order to guide Ryu on how to insert the appropriate rules to the flow tables of OVS. In particular, two rules are added; thus, two REST requests are sent by NCME to Ryu. Next, the appropriate OpenFlow commands are transmitted automatically by Ryu in order to insert the new rules into the OVS flow tables. The REST requests include the following fields: table_id, actions, hard_timeout, idle_timeout, priority, dpid and match. The last field includes also seven extra sub-fields: in_port, eth_type, ip_proto, ipv4_src, tcp_src, ipv4_dst and tcp_dst. First, table_id expresses the identifier of the table where the new rules will be inserted. actions defines a set of instructions such as, for example, to allow, drop or forward the Modbus/TCP packets specified by the rule. hard_timeout denotes the maximum time before discarding. idle_timeout implies the idle time prior to discarding. Next, priority defines the priority of this rule, while dpid denotes the identifier of the corresponding SDN switch (i.e., OVS). Finally, match defines the criteria utilised for identifying the Modbus/TCP packets that will be managed by this rule. in_port indicates the input port of OVS. eth_type determines the Ethernet frame type according to Internet Assigned Numbers Authority (IANA). ip_proto defines the protocol attribute of IPv4 based on IANA. ipv4_src, ipv4_dst, tcp_src and tcp_dst are used to identify the network flows controlled by this rule. In particular, the first two attributes define the source IP address and the source TCP/UDP port, while the latest ones specify the destination IP address and the destination TCP/UDP port, respectively. The first REST request uses the ipv4_src and the tcp_src, while the second uses the ipv4_dst and the tcp_dst. Both ipv4_src and ipv4_dst refer to the same IP address. Similarly, tcp_src and tcp_dst are assigned to 502, which is the default TCP port for the Modbus/TCP protocol.

After normalising and correlating the security events, the mitigation phase follows, taking full advantage of the network programmability provided by SDN. In particular, NCME takes a decision on whether the assets (IIoT physical or virtual devices) related to the security alerts will be isolated or not by SDN-C. The continuous operation of the IIoT, such as SG, is critical since possible disturbances can lead to more devastating consequences, cascading effects or even fatal accidents. Therefore, the NCME cannot instruct arbitrarily the SDN-C to drop the potential malicious network flows. Such an irresponsible action by SDN-C could lead to a more severe impact than an actual cyberattack. For example, the impact of a reconnaissance cyberattack is less significant than a legitimate action targeting the availability of the relevant IIoT/SG assets. Moreover, the presence of a false positive alarm can result in the wrong decision. Although both CVSS and OWASP-RR can estimate the severity of the various threats, the decision about isolating the assets affected by the security alerts cannot exclusively rely on these scores since (a) the sensitive nature of IIoT/SG environments comprises extensive risks that are hard to estimate, (b) both CVSS and OWASP-RR do not consider the special peculiarities of an IIoT/SG environment and (c) they cannot calculate the actual cost, which can be different for each organisation.

Based on the aforementioned remarks, NCME utilises an RL methodology to mitigate or even prevent the various alerts. In particular, for each security alert, the response of NCME relies on three strategies: s_1: NCME will instruct SDN-C to isolate the assets affected by the security alerts, thus corrupting entirely the malicious network flows, s_2: NCME will instruct SDN-C to drop some of the malicious network flows with a probability p_c, thus trying to thwart the cyberattackers' plans and s_3: NCME will wait for the security administrator to decide. The probability p_c in s_2 can be associated with parameters of the IIoT/SG environment or the number of the security alerts. Each strategy is characterised by a respective cost that can be related to financial damages, monetary claims, reputation damage, privacy violation or, in general, unit costs. In this thesis, a general case of unit costs is used. Moreover, the assumption that the unit costs follow the normal distribution $N(\mu, \tau^-1)$ is used. The goal is to train NCME to decide for each security alert the appropriate strategy with the maximum expected reward, which corresponds to the minimum unit cost. The unit cost for each strategy is called $Return$ and symbolised by x_i.

$$
\begin{aligned}
p(\mu \mid X) &\propto p(X \mid \mu)p(\mu) \\
&= (\prod_{i=1}^{N} \sqrt{\frac{\tau}{2\pi}} e^{\frac{-\tau}{2}} (x_i - \mu)^2)(\sqrt{\frac{\lambda_0}{2\pi}} e^{\frac{-\lambda_0}{2}} (\mu - \mu_0)^2 \\
&= ([\sqrt{\frac{\tau}{2\pi}}]^N e^{-\frac{\tau}{2}\sum_{i=1}^{N}(x_i-\mu)^2})(\sqrt{\frac{\lambda_0}{2\pi}} e^{-\frac{\lambda_0}{2}(\mu-m_0)^2}) \\
&= ([\sqrt{\frac{\tau}{2\pi}}]^N e - \frac{\tau}{2}\sum_{i=1}^{N}(x_i - \mu)^2(\sqrt{\frac{\lambda_0}{2\pi}} e^{\frac{\lambda_0}{2}(\mu-m_0)^2}) \\
&\propto (e^{-\frac{\tau}{2}\sum_{i=1} N(x_i-\mu)^2})(e^{-\frac{\lambda_0}{2}(\mu-m_0)^2}) \\
&= e^{-\frac{\tau}{2}\sum_{i=1}^{N}(x_i-\mu)^2 - \frac{\lambda_0}{2}(\mu-m_0)^2} \\
&= e^{-\frac{\tau}{2}\sum_{i=1}^{N}(\mu^2-2\mu x_i+x_i^2) - \frac{\lambda_0}{2}(\mu^2-2\mu m_0+m_0^2)} \\
&= exp(-\frac{\tau}{2}(N\mu^2 - 2\mu\sum_{i=1}^{N} x_i) + \sum_{i=1}^{N} x_i^2) - \frac{\lambda_0}{2}(\mu^2 \\
&- 2\mu m_0 + m_0^2)) \propto exp(-\frac{\tau}{2}(N\mu^2 - 2\mu\sum_{i=1}^{N} x_i) \\
&- \frac{\lambda_0}{2}(m^2 - 2\mu m_0)) = exp(-\frac{\tau N + \lambda_0}{2}\mu^2 + \\
&(\tau\sum_{i=1}^{N} x_i + \lambda_0 m_0)\mu)
\end{aligned}
\tag{5.12}
$$

$$
\begin{aligned}
p(\mu|X) &= \sqrt{\frac{\lambda}{2\pi} exp(-\frac{\lambda}{2}(\mu - m)^2)} \\
&= \sqrt{\frac{\lambda}{2\pi} exp(-\frac{\lambda}{2}(\mu^2 - 2m\mu + m^2))} \\
&\propto exp(-\frac{\lambda}{2}(\mu^2 - 2m\mu)) \\
&= exp(-\frac{\lambda}{2}\mu^2 + m\lambda\mu)
\end{aligned}
\tag{5.13}
$$

$$
\begin{aligned}
\lambda &= \tau N + \lambda_0 \\
m &= \frac{1}{\tau N + \lambda_0}(\tau\sum_{i=1}^{N} x_i + \lambda_0 m_0)
\end{aligned}
\tag{5.14}
$$

The above decision problem can be considered as a MAB problem, where the NCME plays the role of the gambler and the strategies correspond to the slot machines. The MAB problem refers to a

Algorithm 2: SDN-based Mitigation - TS with Normal Distribution

```
Data: $S$, $\tau$, $m_0$, $\lambda_0$, $m$, $\lambda$, $x\_Matrix$, $sum\_x\_Matrix$, $\lambda\_Matrix$, $m\_Matrix$
Result: selectedStrategy
$securityEventCounter = 0$;
$\tau = 1$, $m_0 = 0$, $\lambda_0 = 1, m = 0$;
$x\_Matrix = [], sum\_x\_Matrix = []$, $\lambda\_Matrix = [], m\_Matrix = []$;
while True do
    Receive a security alert;
    securityAlertCounter = securityAlertCounter +1;
    selectedStrategy = 0;
    min = $\infty$;
    for $strategy \leftarrow 0$ to $S$ by $1$ do
        posteriorProbabilitySample = $N(0,1)\sqrt{\frac{1}{\tau}} + m\_Matrix[selectedStrategy]$;
        if posteriorProbabilitySample $<$ min then
            min = posteriorProbabilitySample;
            selectedStrategy = strategy;
        end
    end
    SDN controller executes selectedStrategy;
    $x\_Matrix[selectedStrategy] = N(0,1)\sqrt{\frac{1}{\tau}} + \mu$;
    $sum\_x\_Matrix[selectedStrategy] =$
      $sum\_x\_Matrix[selectedStrategy] + x\_Matrix[selectedStrategy]$;
    $\lambda\_Matrix[selectedStrategy] = \lambda\_Matrix[selectedStrategy] + \tau$;
    $m\_Matrix[selectedStrategy] = \tau \times sum\_x\_Matrix =$
      $[selectedStrategy]/\lambda\_Matrix[selectedStrategy]$;
end
```

gambler who tries to increase the profit, choosing each time that slot machine offering the maximum payout. Each time, the gambler can choose only one slot machine. Therefore, the gambler faces an exploration-exploitation dilemma, where exploration denotes the identification of the slot machine providing the maximum profit while exploitation maximises the gambler's profit. More details and a formal definition of the MAB problem is given by V. Kuleshov et al. in [92] and M. Katehakis et al. in [83]. Unlike the typical MAB problem, the goal is to minimise the possible cost related to the aforementioned mitigation strategies. Therefore, to solve this kind of MAB problem, the Thompson Sampling (TS) method is used. TS can balance the consecutive actions of an exploration-exploitation dilemma, where in our case exploration refers to discovering more information about the cost of the various strategies, while the exploitation focuses on mitigating the security alerts with the minimum cost. TS is a Bayesian method, which utilises the properties of the conjugate pairs to calculate the posterior probability $p(\mu|X)$. In particular, given $X = x_1, x_2, ..., x_n$, the likelihood $p(X|\mu,\tau) = \prod_{i=1}^{N} \sqrt{\frac{\tau}{2\pi} e^{-\frac{\tau}{2}(x_i-\mu)^2}}$, $x_i \sim N(\mu, \tau^{-1})$. Given τ and $\mu \sim N(m_0, \lambda_0^{-1})$, the prior probability $p(\mu)$ equals with $p(\mu) = \prod_{i=1}^{N} \sqrt{\frac{\lambda_0}{2\pi} e^{-\frac{\lambda_0}{2}(\mu-m_0)^2}}$, while the posterior probability $p(\mu|X)$ can be calculated by $p(\mu|X) \propto p(X|\mu)p(\mu)$ where $\mu|X \sim N(m, \lambda^{-1})$. The goal is to define the parameters m and λ of the posterior probability $p(\mu|X)$ as a function of the data X and the prior parameters m_0 and

λ_0. Thus, based on Equations 5.12-5.14, $\lambda = \tau N + \lambda_0$ and $m = \frac{1}{\tau N + \lambda_0}(\tau \sum_{i=1}^{N} x_i + \lambda_0 m_0)$. Suppose μ follows the standard normal distribution, (i.e., $m_0 = 0$ and $\lambda_0 = 1$), for each security event, TS takes a sample from the posterior probability for each strategy: $N(m, \lambda^{-1}) \rightarrow N(0,1)\sqrt{\frac{1}{\tau}} + m$, selecting the minimum value. Next m and λ are updated based on Equation 5.14. Algorithm 2 shows how the TS method is applied. The matrices: x_Matrix, sum_x_Matrix, λ_Matrix, and m_Matrix are used to store x_i, $\sum_{i=1}^{N} x_i$, λ and m for each strategy. N denotes the corresponding number of the latest security alert, while S indicates a set of the three strategies: s_1, s_2 and s_3 described earlier.

5.5.2 Honeypot Mitigation and Resilience

In this section, a sophisticated honeypot deployment mechanism is provided as a mitigation action in order to increase the resilience of the underlying IIoT/SG infrastructure based on the various security alerts composed by NCME. The proposed mechanism relies on a security game between two entities: (a) cyberattacker and (b) defender. In general, the security games introduce an analytical framework in order to formalise the relationship between malicious cyberattackers and defenders like security officers and security administrators. Their interactions are modelled using the rich mathematical basis provided by the game theory field. In particular, the main idea behind the security games is based on the allocation of the limited resources of the previous entities. If both had unlimited resources, then the solution would be trivial and meaningless. However, in reality, the players are characterised by particular constraints, and they have to act strategically in order to allocate the appropriate number of resources.

A security game is defined by four main entities, namely (a) the players (cyberattacker and defender), (b) the set of strategies for each player, (c) the outcome of each interaction and (d) the information structure. The terms cyberattacker and defender are used for the sake of simplicity, indicating a plethora of malicious cyberattackers and people protecting the underlying infrastructure. For each strategy, there is a specific outcome in terms of costs and benefits for each player. If the player can estimate these values, they can adapt their strategies suitably. The NE solution lies at the intersection of the best responses, where no player has any motive to deviate from the NE since it would result in a worse outcome. Therefore, practically, if the defender adopts NE, then it does not matter whether the cyberattacker is reasonable or not since any deviation from the NE will decrease the attacker's benefit and the cost to the defender.

In this thesis, a deployment mechanism is provided, calculating how many honeypots should be deployed in an IIoT/SG infrastructure, based on a honeypot security game if NE exists or differently using (a) a max-min analysis (solved based on the cvxpy Python library) or (b) an RL method, which relies on epsilon-greedy. The following subsections explain in detail (a) the honeypot security game, (b) the NE solution, (c) the max-min analysis when NE does not exist and (c) the RL method when NE does not also exist. In particular, the following mathematical analysis aims to prove the existence and the valid performance of NE, which is the basis for calculating how many honeypots should be deployed. Accordingly, if NE does not exist, the following sections describe and prove how the defender's strategy can be converted into a max-min convex optimisation problem or an RL problem and how they can be solved, respectively.

Table 5.5: Symbols and Notation

Symbol & Notation	Explanation
N_{max}	The maximum number of the real IIoT/SG assets and honeypots that can be simultaneously connected.
N	The number of the real IIoT/SG assets and honeypots that are connected.
$s_{a,i}$	The strategy of the attacker for the i-th host.
$s_{d,i}$	The strategy of the defender for the i-th host.
a_1	The benefit of the attacker for each attack against a real IIoT/SG asset.
a_2	The cost of the attacker for each attack against a honeypot.
a_3	The cost of the attacker for each attack against any machine (honeypot or not).
d_1	The benefit of the defender for each attack against a honeypot.
d_2	The cost of the defender for each attack against a real IIoT/SG asset.
d_3	The cost of the defender for each real IIoT/SG asset which is replaced by a honeypot.
d_4	The cost of the defender as N increases.
$U_A[t]$	The utility of the $Attacker$ at the time interval t.
$U_D[t]$	The utility of the $Defender$ at the time interval t.
θ	The ratio of N utilised by honeypots.
ϕ	Portion of the number of hosts (N) that are attacked in the t-th time interval.

5.5.2.1 Honeypot Security Game

Let N_{max} denote the maximum number of honeypots and real IIoT/SG entities/devices that can be hosted in a network by the defender. N_{max} is determined by the defender and depends on the available IPs and the available computing resources that can be provided. Also, let $N \leq N_{max}$ denote the total number of connected IIoT/SG entities/devices that can be either honeypots or real devices. Apart from the total number of connected IIoT/SG entities/devices, the defender can control which of them are used by real devices and honeypots. The portion of N that consists of honeypots is represented by θ. The attacker's set of strategies is whether or not to attack a host. For the t-th interval, let $s_{d,i}[t] \in \{1, -1\}$ be equal to 1 when the i-th host is used by a real device and equal to -1 when it is used by a honeypot. On the other hand, regarding the set of strategies of the attacker, let $s_{a,i}[t] \in \{1, 0\}$ be equal to 1 when the attacker attacks the j-th host and equal to 0 when the j-th host is not attacked. For the sake of clarity, the various symbols and notations are provided in Table 5.5.

According to the aforementioned description, the payoff of the attacker during t is given by Equation 5.15, which takes into account (a) the number of the actual IIoT/SG entities/devices under attack and (b) the number of honeypots under attack and (c) the total number of attacks.

$$U_A[t] = f(a_{i\in\{1,2,3\}}), \sum_{i=1}^{N} \frac{(1+s_{d,i})}{2} \times s_{a,i}, \sum_{i=1}^{N} \frac{1-s_{d,i}}{2} \times s_{a,i}, \sum_{i=1}^{N} s_{a,i} \tag{5.15}$$

a_1 , a_2 and a_3 in Equation 5.15, refer to non-negative weights about the benefit for each attack against an actual IIoT/SG entity/device, the damage for each attack against a honeypot and the damage for each attack against any asset (honeypot or actual IIoT/SG entity/device), respectively. Assuming that the aforementioned terms have a linear impact on the attacker's payoff, $U_{A[t]}$ could be written as:

$$U_{\mathrm{A}}[t] = a_1 \sum_{i=1}^{N} \frac{(1+s_{\mathrm{d},i})}{2} s_{\mathrm{a},i} - a_2 \sum_{i=1}^{N} \frac{1-s_{\mathrm{d},i}}{2} s_{\mathrm{a},i} - a_3 \sum_{i=1}^{N} s_{\mathrm{a},i}. \tag{5.16}$$

On the other side, the payoff of the defender in time t is given in Equation 5.17, taking into account the number of honeypots under attack, the number of the real IIoT/SG entities/devices under attack, the number of the actual IIoT/SG entities/devices and the total number of IIoT/SG entities/devices, including honeypots.

$$U_{\mathrm{D}}[t] = g\left(d_{i\in\{1,2,3,4\}}, \sum_{i=1}^{N} \frac{(1-s_{\mathrm{d},i})}{2} s_{\mathrm{a},i}, \sum_{i=1}^{N} \frac{(1+s_{\mathrm{d},i})}{2} s_{\mathrm{a},i}, \sum_{i=1}^{N} \frac{(1+s_{\mathrm{d},i})}{2}, N\right), \tag{5.17}$$

d_1, d_2, d_3, d_4 in Equation 5.18 refer to non-negative weights that correspond to the benefit for each attack against a honeypot, the cost for each attack against a real IIoT/SG entity/device, the cost for each real IIoT/SG entity/device replaced by a honeypot, and the cost of the defender as N increases. Assuming that the previous terms have a linear impact on the attacker's payoff, $U_D[t]$ could be written as:

$$U_{\mathrm{D}}[t] = d_1 \sum_{i=1}^{N} \frac{(1-s_{\mathrm{d},i})}{2} s_{\mathrm{a},i} - d_2 \sum_{i=1}^{N} \frac{(1+s_{\mathrm{d},i})}{2} s_{\mathrm{a},i} - d_3 \left(\sum_{i=1}^{N} \frac{(1+s_{\mathrm{d},i})}{2} - N_r\right)^2 - d_4 N \tag{5.18}$$

Considering that the attacker attacks ϕN IIoT/SG entities/devices, let $0 \leq \theta[t] \leq 1$ denote the ratio of the total number of N that are honeypots. It is also assumed that all IIoT/SG entities/devices have the same probability of being a honeypot or attacked. In this case, the payoff of the attacker can be expressed by Equation 5.19, as a function of ϕ and θ, i.e.,

$$U_{\mathrm{A}} = \tilde{f}(a_1, a_2, a_3, \phi, \theta, N). \tag{5.19}$$

The goal of the attacker is to maximise the relevant payoff. Thus, this can be written as a maximisation problem, as follows.

$$\begin{aligned} \max_{\phi} \quad & U_{\mathrm{A}} \\ \textbf{s.t.} \quad & \mathrm{C}_1 : 0 \leq \phi \leq 1 \end{aligned} \tag{5.20}$$

The expected payoff of the defender can also be written as a function of ϕ and θ, i.e.,

$$U_{\mathrm{D}} = \tilde{g}(d_1, d_2, d_3, d_4, \phi, \theta, N) \tag{5.21}$$

Similarly, the goal of the defender is to maximise the relevant payoff based on Equation 5.22

$$\begin{aligned} \max_{\theta, N} \quad & U_{\mathrm{D}} \\ \textbf{s.t.} \quad & \mathrm{C}_1 : 0 \leq \theta \leq 1 \\ & \mathrm{C}_2 : 0 \leq N \leq N_{max} \end{aligned} \tag{5.22}$$

5.5.2.2 Nash Equilibrium Solution

Definition: The NE of this security game refers to the situation (θ^*, ϕ^*, N^*) when both players (i.e., attacker and defender) cannot maximise their utility functions, i.e., $(U_A[t], U_D[t])$ with any other action. This can be written as follows:

$$U_D(\theta^*, N^*, \phi^*) \geq U_D(\theta, N, \phi^*) \tag{5.23}$$

$$U_A(\theta^*, N^*, \phi^*) \geq U_D(\theta^*, N^*, \phi) \tag{5.24}$$

Thus, based on Equations 5.16 and 5.18, the payoff of the defender and the attacker can be written as follows, respectively.

$$U_{\mathrm{D}} = d_1\theta\phi N - d_2(1-\theta)\phi N - d_3((1-\theta)N - N_r)^2 - d_4 N, \tag{5.25}$$

$$U_{\mathrm{A}} = a_1(1-\theta)\phi N - a_2\theta\phi N - a_3\phi N \tag{5.26}$$

According to the previous equations, the presence and derivation of the NE will be investigated.

Lemma: If the NE exists - $\phi^* \in \{0, 1\}$.

Proof: Supposing that $\theta^*, N^*, \phi^{'}$ is the NE and $\phi^* \in \{0, 1\}$ then:

$$a_1(1-\theta)\phi' - a_2\theta\phi' - a_3\phi' \geq a_1(1-\theta) - a_2\theta - a_3 \tag{5.27}$$

i.e., $\phi' \geq 1$, which contradicts the assumption.

Theorem: The NE is given by Equation 5.28.

$$(\theta^*, N^*, \phi^*) = \begin{cases} \left(0, \frac{2d_3N_r - d_4}{2d_3}, 0\right), \text{ if } 0 \leq \frac{2d_3N_r-d_4}{2d_3} \leq N_{max} \text{ and } a_1 \leq a_3 \\ (0,0,0), \text{if } \frac{2d_3N_r-d_4}{2d_3} < 0 \\ \left(\dfrac{d_1+d_2+2d_3N_{max}-2d_3N_r}{2d_3N_{max}}, N_{max}, 1\right), \text{if} \\ 0 \leq \dfrac{d_1+d_2+2d_3N_{max}-2d_3N_r}{2d_3} \leq N_{max} \text{ and } d1 > d_4 \text{ and} \\ (a_1+a_2)N_r \geq (a_2+a_3)N_{max} + \dfrac{(a_1+a_2)(d_1+d_2)}{2d_3} \\ \left(0, N_r - \frac{d_2+d_4}{2d_3}, 1\right), \text{ if } \frac{d_1+d_2+2d_3N_{max}-2d_3N_r}{2d_3} < 0 \text{ and } a_1 > a_3 \end{cases} \tag{5.28}$$

Proof: Suppose that $\phi^* = 0$. This can be valid only if:

$$a_1(1-\theta) - a_2\theta - a_3 \leq 0 \tag{5.29}$$

When $\phi* = 0$, then U_D is a decreasing function with respect to θ^*, i.e., $\theta^* = 0$. Thus, $\phi = 0$ can belong to the equilibrium if $a_1 \leq a_3$. By setting $\frac{\partial U_D}{\partial N} = 0, \theta = 0$ and $\phi = 0$, then:

$$N^* = \left[\frac{2d_3N_r - d_4}{2d_3}\right]_0^{N_{max}} \tag{5.30}$$

where $[\cdot]_0^{N_{max}} = min\{max\{\cdot, 0\}, N_{max}\}$

Suppose that $\phi^* = 1$, by setting $\frac{\partial U_D}{\partial \theta} = 0$, then:

$$\theta = \frac{d_1 + d_2 + 2d_3N - 2d_3N_r}{2d_3N} \tag{5.31}$$

Next, suppose that $0 < \frac{d_1+d_2+2d_3N-2d_3N_r}{2d_3N} < 1$, $\frac{\partial U_D}{\partial N} \geq 0$ if $d_1 \geq d_4$, in this case, $N = N_{max}$ and $\phi = 1$ belongs to the equilibrium if:

$$U_{A,\phi=1} \geq U_{A,\phi=0} \tag{5.32}$$

which can be written as:

$$(a_1 + a_2)N_r \geq (a_2 + a_3)N_{max} + \frac{(a_1 + a_2)(d_1 + d_2)}{2d_3} \tag{5.33}$$

If $\frac{d_1+d_2+2d_3N_{max}-2d_3N_r}{2d_3N} < 0$, then based on $\frac{\partial U_D}{\partial N}$:

$$N^* = N_r - \frac{d_2 + d_4}{2d_3} \tag{5.34}$$

Obviously, $\phi^* = 1$ belongs to the equilibrium if $a_1 > a_3$ since then $U_{A,\phi=1} \geq U_{A,\phi=0}$. Finally, $\theta = 1$ cannot belong to the equilibrium since, in this case, $\phi^* = 0$.

5.5.2.3 MaxMin-based Honeypot Deployment

As observed in the previous subsection, the NE does not always exist. Thus, to meet the requirements of practical scenarios, a different framework is required when the NE does not exist. In this case, the strategy of the defender can be chosen by using a max-min analysis, which focuses on maximising the payoff of the defender in the worst-case scenario. Therefore, the max-min equation for the defender is defined as follows.

$$\underset{0\leq\theta\leq1, 0\leq N\leq N_{max}}{max} \; \underset{\phi}{min} \, U_D \tag{5.35}$$

To solve Equation 5.35, it should be observed that U_D is either an increasing or a decreasing value of ϕ, for specific values of θ and N. Thus, the attacker can force the defender to receive the lowest value by either choosing 1 or 0. When $\phi = 1$:

$$U_D = d_1\theta N - d_2(1-\theta)N - d_3((1-\theta)N - N_r)^2 - d_4N \tag{5.36}$$

while when $\phi = 0$:

$$U_D = -d_3((1-\theta)N - N_r)^2 - d_4N \tag{5.37}$$

$$\begin{aligned} \max_{\theta,N} \quad & y \\ \textbf{s.t.} \quad & \mathrm{C}_1 : 0 \leq \theta \leq 1 \\ & \mathrm{C}_2 : 0 \leq N \leq N_{max} \\ & \mathrm{C}_3 : d_1\theta n - d_2(1-\theta)N - d_3((1-\theta)N - N_r)^2 - d_4 N \geq y \\ & \mathrm{C}_4 : -d_3((1-\theta)N - N_r)^2 - d_4 N \geq y \end{aligned} \tag{5.38}$$

The previous problem is non-convex and cannot be solved easily. Consequently, by setting $\theta N = N_1$ and $(1-\theta)N = N_2$, the above problem can be written as:

$$\begin{aligned} \max_{N_1,N_2} \quad & y \\ \textbf{s.t.} \quad & \mathrm{C}_1 : N_1 + N_2 \leq N_{max} \\ & \mathrm{C}_2 : d_1 N_1 - d_2 N_2 - d_3(N_2 - N_r)^2 - d_4(N_1 + N_2) \geq y \\ & \mathrm{C}_3 : -d_3(N_2 - N_r)^2 - d_4(N_1 + N_2) \geq y \\ & \mathrm{C}_4 : N_1, N_2 \geq 0 \end{aligned} \tag{5.39}$$

The optimisation problem in Equation 5.39 is a convex one and can be solved by standard convex optimization methods. In this thesis, this problem is solved through the cvxpy Python library.

5.5.2.4 AI-powered Honeypot Deployment

An alternative solution for deploying the various honeypots in a sophisticated manner is to take advantage of RL. In particular, based on the security alerts generated by NCME, the goal is to set the appropriate ratio θ in order to maximise $U_D[t]$ each time. To re-define, the appropriate number of θ for each security alert in the time t can be expressed as a MAB problem, where exploitation intends to maximise $U_D[t]$ (Equation 5.18) and exploration aims to test different values of θ to discover more information for the $Attacker$ in terms of Equation 5.16. The proposed solution plays the role of the gambler, and the various values of $theta$ represent the slot machines. To solve the MAB problem, the $e-Greedy$ method is adopted, where commonly, the mean of $theta$ is chosen, providing the maximum value $U_D[t]$ (Equation 5.18). Moreover, there is a small probability e where other values of θ are selected in order to discover how Equation 5.18 ranges. Algorithm 3 reflects how the proposed solution takes the decision to deploy θ honeypots, utilising $e-Greedy$.

5.6 Chapter Summary

This chapter summarises the technical achievements during this PhD program in terms of a unified SDN-enabled SIEM solution. Three independent IDPS undertake to monitor the underlying IIoT/SG

Algorithm 3: AI-Powered Honeypot Deployment

```
Data: $N_{max}$, $N$, $UD\_Matrix$, $sum\_\theta\_Matrix$, $mean\_\theta\_Matrix$, $max\_mean$,
      securityAlertCounter, $a_1$, $a_2$, $a_3$, $d_1$, $d_2$, $d_3$, $d_4$
Result: $\theta_{selected}$
$size\_\theta\_Matrix = []$, $UD\_Matrix = []$, $sum\_\theta\_Matrix = []$, $mean\_\theta\_Matrix = []$,
  securityAlertCounter = 0, $max\_mean$ = 0, $\theta_{selected}$ = 0, $a_1$, $a_2$, $a_3$, $d_1$, $d_2$, $d_3$, $d_4$ = init();
while True do
    Receive a security alert;
    securityAlertCounter = securityAlertCounter +1;
    $max\_mean$ = 0;
    p = random number in [0,1];
    if $p < e$ then
        $\theta_{selected}$ = random integer number in [1, N];
        $UD\_Matrix[\theta] = d_1 \sum_{i=1}^{N} \frac{1-S_{d,i}}{2} s_{a,i} - d_2 \sum_{i=1}^{N} \frac{1+S_{d,i}}{2} s_{a,i} - d_3 \sum_{i=1}^{N} \frac{1+s_{d,i}}{2} - d_4 N$;
        $sum\_\theta\_Matrix[\theta] = sum\_\theta\_Matrix[\theta] + UD\_Matrix[\theta]$;
        $mean\_\theta\_Matrix = sum\_\theta\_Matrix[\theta]$ / securityAlertCounter;
    end
    else
        for $\theta \leftarrow 1$ to $N$ by $1$ do
            $UD\_Matrix[\theta] = d_1 \sum_{i=1}^{N} \frac{1-S_{D,i}}{2} s_{A,i} - d_2 \sum_{i=1}^{N} \frac{1+S_{D,i}}{2} s_{A,i} - d_3 \sum_{i=1}^{N} \frac{1+s_{D,i}}{2} - d_4 N$;
            $sum\_\theta\_Matrix[\theta] = sum\_\theta\_Matrix[\theta] + UD\_Matrix[\theta]$;
            $mean\_\theta\_Matrix = sum\_\theta\_Matrix[\theta]$ / securityAlertCounter;
            if $mean\_\theta\_Matrix[\theta] > max\_mean$ then
                $max\_mean = mean\_\theta\_Matrix[\theta]$; $\theta_{selected} = \theta$;
            end
        end
    end
end
```

infrastructure in near real-time, generating relevant security events. For this purpose, multiple collaborative ML/DL models were implemented for each IDPS. Next, NCME receives, normalises and correlates the security events with each other, thus composing security alerts. For the normalisation process, the AlienVault OSSIM format was used, while the correlation mechanism relies on custom security rules. Based on the security alerts, NCME also instructs appropriately the SDN-C to isolate the malicious network flows in a timely manner. For this purpose, TS is used. Finally, NCME includes sophisticated honeypot deployment mechanisms that also act as a mitigation strategy, enhancing the resilience of the underlying IIoT/SG infrastructure. These mechanisms rely on a honeypot security game between the attacker and the defender. First, the NE solution is identified. If NE does not exist, two alternative options are also provided, taking full advantage of a max-min analysis and the epsilon-greedy technique, respectively.

Chapter 6

Evaluation Analysis

This chapter focuses on the evaluation of the detection and mitigation mechanisms described in the previous chapter. In particular, the detection efficiency of (a) NF-IDPS, (b) H-IDPS and (c) V-IDPS is evaluated, while also the mitigation mechanisms of NCME (i.e., (a) SDN-based mitigation and (b) Honeypot Mitigation and Resilience) are assessed. For this purpose, four actual IIoT/SG evaluation environments were considered from the H2020 SPEAR project. It is worth mentioning that the validation activities took place under testing conditions for each evaluation environment without affecting their normal operation. Therefore, based on appropriate datasets and simulation experiments, the corresponding evaluation results are calculated in terms of particular metrics. In addition, a set of appropriate comparative methods is defined for each category, thus identifying the most efficient method. Finally, the evaluation results are provided and discussed for each component of the proposed SDN-enabled SIEM.

6.1 Evaluation Environments

The detection and mitigation mechanisms of the SDN-enabled SIEM presented previously were implemented and validated successfully, using both actual and fictitious data originating from four IIoT/SG use cases from the SPEAR project [142]: (a) hydropower plant, (b) substation, (c) power plant and (e) smart home. The first three cases use logic controllers, such as PLCs and RTUs, that monitor and control the operation of the entire infrastructure and mainly the functionality of industrial devices, such as turbines, transformers and generators. These controllers communicate with an MTU. Finally, through an HMI, the system operator can monitor and handle the operation of PLCs and RTUs, sending the appropriate commands via the corresponding IIoT application-layer protocols (e.g., Modbus, DNP3 and IEC 61850). Finally, the smart home environment includes smart meters that measure energy consumption and relevant statistics. This information is also stored in an MTU, using the corresponding IIoT application-layer protocols. Through SPAN, both NF-IDPS and V-IDPS can monitor the network traffic data of the previous use cases, thus generating relevant security events. On the other side, H-IDPS can retrieve from MTU the operational data (i.e., time series electricity data) of each use case, thus detecting potential anomalies. For this purpose, an Elastic Stack API was utilised.

6.2 Datasets

According to the operational characteristics of the evaluation environments discussed in the previous section, appropriate datasets were created and used to train and test the ML and DL models of NF-IDPS, V-IDP and H-IDPS. These datasets were synthesised either by creating them from scratch with the emulation and execution of the respective cyberattacks and anomalies or by combining existing intrusion detection datasets (such as CSE-CIC-IDS 2018) with the normal records coming from each evaluation environment (i.e., hydropower plant, substation, power plant and smart home). In particular, regarding the ML and DL models of NF-IDPS and V-IDPS, new datasets were created for Modbus/TCP, DNP3 and IEC 60870-5-104. These datasets will be publicly available in IEEE Dataport and Zenodo. On the other side, regarding the ML and DL models of H-IDPS, suitable datasets were produced from scratch based on the guidelines of security and safety experts from each evaluation environment. Due to the sensitive nature of these datasets, they cannot be published in the context of this thesis. However, Appendices E-L summarise the features used in each case.

6.3 Evaluation Metrics and Comparative Methods

Before presenting and discussing the evaluation results of the detection and mitigation mechanisms of the SDN-enabled SIEM, the evaluation metrics should be introduced first. Therefore, True Positives (TP) denotes the number of cyberattacks or anomalies detected as a malicious/anomalous behaviour. True Negatives (TN) indicates the number of normal activities recognised correctly as a normal behaviour. On the other side, FP implies the number of normal activities classified as a malicious behaviour. Finally, FN indicates the number of of cyberattacks/anomalies recognised as a normal behaviour. Based on the above terms, the following evaluation metrics are defined.

Accuracy: Accuracy (Equation (6.1)) indicates the ratio between the correct predictions and the total number of samples. Accuracy can be used as an unbiased metric if the training dataset includes an equal number of data samples for all classes. For instance, if the training dataset includes 90% data samples that present normal behaviour and 10% data samples with anomalies, then the Accuracy can reach 90% by predicting each instance as normal.

$$Accuracy(ACC) = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

True Positive Rate (TPR): TPR (Equation (6.2)) represents the ratio of cyberattacks or anomalies classified correctly as a malicious/anomalous behaviour.

$$TPR = \frac{TP}{TP + FN} \tag{6.2}$$

False Positive Rate (FPR): FPR (Equation (6.3)) refers to the ratio of normal activities recognised as intrusions or anomalies.

$$FPR = \frac{FP}{FP + TN} \tag{6.3}$$

F1-Score: The F1-Score (Equation 6.4 refers to the golden ratio between TPR and precision. Precision refers to another evaluation metric, expressing the ratio between the malicious/anomalous activities detected correctly and the overall number of the malicious/anomalous activities.

$$F1 = \frac{2 \times Precision \times TPR}{Precision + TPR} \text{ where } Precision = \frac{TP}{TP + FP} \tag{6.4}$$

For each IDPS of the proposed SDN-enabled SIEM, a set of ML and DL methods was used in a comparative analysis in order to identify the most efficient model in each case. In particular, regarding the intrusion detection models of NF-IDPS, multiple ML/DL methods were used, including Logistic Regression, LDA, Decision Trees, Naive Bayes, SVM, Random Forest, Adaboost, MLP, QDA and KNN. On the other side, regarding the anomaly detection models of NF-IDPS, the following outlier detection methods were used: ABOD, Isolation Forest, PCA, MCD and Autoencoder(s). Similarly to the anomaly detection models of NF-IDPS, almost the same outlier detection methods are used for the detection models of H-IDPS, including also ARIES GAN. The aforementioned ML and DL methods were discussed in the previous chapters. Finally, regarding V-IDPS, several pre-trained CNNs were used, including (a) DenseNet121 [67], (b) DenseNet169 [67], (c) DenseNet201 [67], (d) EfficientNetB0 [182], (e) EfficientNetB7 [182], (f) MobileNet [64], (g) MobileNetV2 [164], (h) NASNetLarge [223], (i) NASNetMobile [223], (j) ResNet50 [61], (k) ResNet50V2 [61], (l) ResNet101 [61], (m) ResNet101V2 [61], (n) ResNet152 [61], (o) ResNet152V2 [61], (p) VGG16 [176], (r) VGG19 [176] and (s) Xception [35]. Finally, the detection efficiency of V-IDPS was also compared with other typical ML and DL methods, such as Logistic Regression, LDA, Decision Trees, Naive Bayes, SVM, Random Forest, Adaboost, MLP, QDA and KNN.

On the other hand, regarding the mitigation mechanisms of NCME, first, the SDN-based mitigation mechanism is evaluated. In particular, the range of the posterior probability is investigated based on the various number of security alerts for each mitigation strategy (i.e., s_1: NCME will instruct SDN-C to isolate the assets affected by the security alerts, thus corrupting entirely the malicious network flows, s_2: NCME will instruct SDN-C to drop some of the malicious network flows with a probability p_c, thus trying to thwart the cyberattackers' plans and s_3: NCME will wait for the security administrator to decide.). For this purpose, a simulation experiment took place based on the Modbus/TCP Intrusion Detection Dataset (it will be publicly available in IEEE Dataport and Zenodo). Finally, the accuracy of the TS method in choosing the optimal mitigation strategy is compared with another relevant method called Upper Confident Bound (UCB) [59]. Finally, three experiments were carried out to investigate the strategies of the defender regarding the deployment of honeypots. The first experiment refers to when the NE is available. The second experiment focuses on when the NE does not exist, and the MaxMin-based honeypot deployment mechanism is chosen. Finally, the third experiment focuses on when the AI-powered honeypot deployment mechanism is selected when also NE is not available.

6.4 Evaluation Results

Next, based on the previous evaluation metrics and comparative methods, the evaluation results of (a) NF-IDPS, (b) H-IDPS, (c) V-IDPS and (d) NCME are provided and discussed.

6.4.1 NF-IDPS Evaluation Results

This subsection summarises the evaluation results of the intrusion and anomaly detection models that compose the IDE of NF-IDPS. In particular, the comprehensive ML/DL comparative analysis of NF-IDPS is provided in Appendix M. It is worth mentioning that all ML and DL methods were fine-tuned after several experiments. Fig. 6.1 and Fig. 6.2 summarise the detection performance of the NF-IDPS intrusion and anomaly detection models, respectively. In particular, Fig. 6.1 focuses on the intrusion detection models, while Fig. 6.2 shows the efficiency of the anomaly detection models.

Therefore, according to chapter 5, NF-IDPS includes two Modbus/TCP-related detection models, namely: (a) Modbus/TCP Intrusion Detection Model and (b) Modbus/TCP Anomaly Detection Model. The first one adopts a decision tree, where $ACC = 0.964$, $TPR = 0.749$, $FPR = 0.019$ and $F1 = 0.749$. On the other hand, the Modbus/TCP Anomaly Detection Model uses the proposed autoencoder discussed in Chapter 5. The detection performance of this autoencoder is defined by $ACC = 0.950$, $TPR = 0.999$, $FPR = 0.099$ and $F1 = 0.952$.

Similarly, NF-IDPS contains three intrusion and anomaly detection models about DNP3: (a) DNP3 Intrusion Detection Model, (b) DNP3 TCP/IP Intrusion Detection Model and (c) DNP3 TCP/IP Anomaly Detection Model. For the first model, a CART decision tree is utilised, where $ACC = 0.959$, $TPR = 0.959$, $FPR = 0.0051$ and $F1 = 0.9594$. The second model also uses a CART decision tree, where $ACC = 0.797$, $TPR = 0.797$, $FPR = 0.0203$ and $F1 = 0.7821$. Finally, the third model relies on ABOD, where $ACC = 0.951$, $TPR = 0.999$, $FPR = 0.097$ and $F1 = 0.953$.

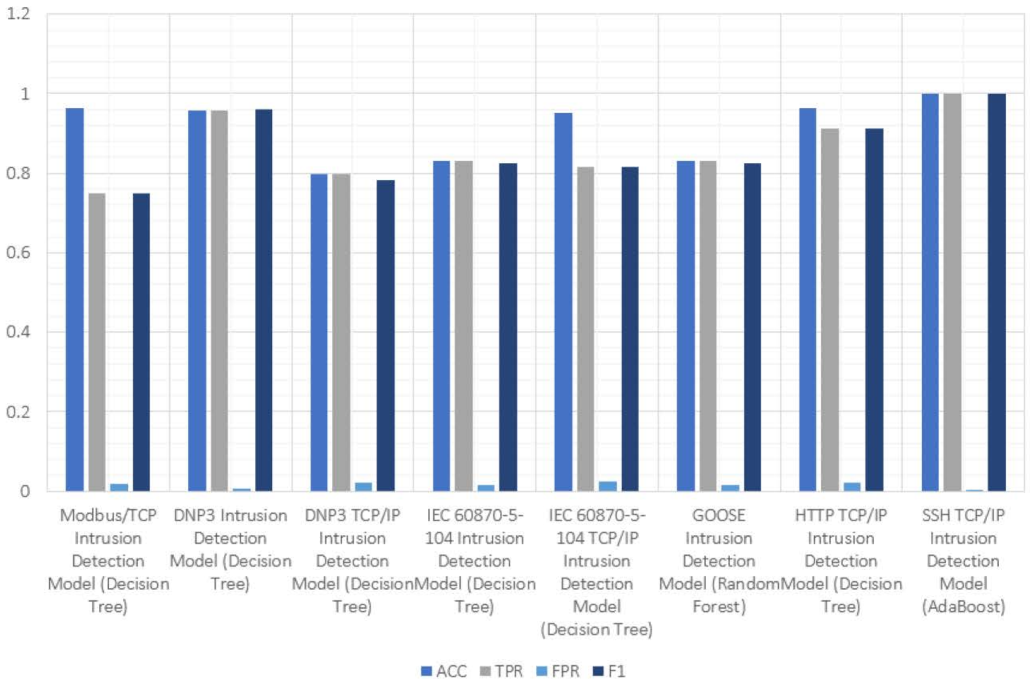


FIGURE 6.1: Evaluation Results of NF-IDPS Intrusion Detection Models

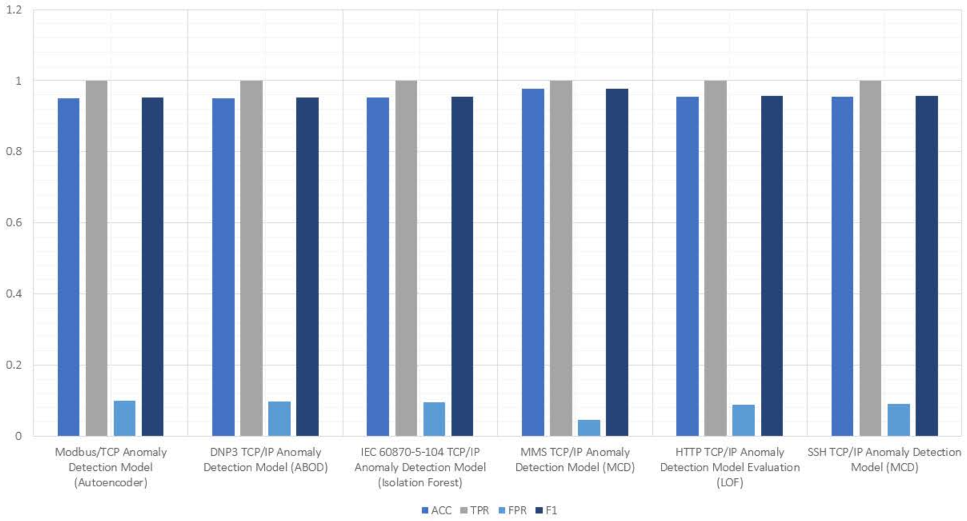


FIGURE 6.2: Evaluation Results of NF-IDPS Anomaly Detection Models

Next, three intrusion and anomaly detection models were implemented for the IEC 60870-5-104 protocol, namely: (a) IEC 60870-5-104 Intrusion Detection Model, (b) IEC 60870-5-104 TCP/IP Intrusion Detection Model and (c) IEC 60870-5-104 TCP/IP Anomaly Detection Model. The first two models adopt a CART decision tree. The ACC, TPR, FPR and the F1-Score of the first model are equal to $ACC = 0.831, TPR = 0.831$, $FPR = 0.015$ and $F1 = 0.825$. On the other hand, for the second model, $ACC = 0.953, TPR = 0.815$, $FPR = 0.026$ and $F1 = 0.815$. Finally, the third model uses the Isolation Forest method, where the ACC, TPR, FPR and F1-Score are equal to 0.952, 0.999, 0.0941 and 0.9550, respectively.

Next, NF-IDPS includes two detection models for the IEC 61850 standard: (a) GOOSE Intrusion Detection Model and (b) MMS TCP/IP Anomaly Detection Model. The first one focuses on GOOSE attacks, while the second is responsible for detecting MMS anomalies. Therefore, the first model adopts the Random Forest method, where $ACC = 0.831, TPR = 0.831, FPR = 0.015$ and $F1 = 0.825$. On the other hand, the second model uses MCD, where the ACC, TPR, FPR and F1-Score are equal to 0.977, 0.999, 0.045 and 0.9777, respectively.

Subsequently, two detection models are used for the HTTP protocol, namely: (a) HTTP TCP/IP Intrusion Detection Model and (b) HTTP TCP/IP Anomaly Detection Model. The first model uses CART, while the second model relies on LOF. The ACC, TPR, FPR and the F1 score of the first model are equal to 0.964, 0.911, 0.022 and 0.9111, respectively. On the other side, the detection performance of the second model is also efficient where $ACC = 0.955, TPR = 0.999$, $FPR = 0.089$ and $F1 = 0.957$.

Finally, two detection models are also used for the SSH protocol: (a) SSH TCP/IP Intrusion Detection Model and (b) SSH TCP/IP Anomaly Detection Model. The first model is based on AdaBoost, where $ACC = 0.999, TPR = 0.999$, $FPR = 0.001$ and $F1 = 0.999$. In contrast, the second model uses MCD, where $ACC = 0.954$, $TPR = 0.999$, $FPR = 0.0916$ and $F1 = 0.9561$.

6.4.2 H-IDPS Evaluation Results

Fig. 6.3 summarises the detection performance of H-IDPS in terms of the corresponding ML/DL models for each use case. In particular, the ARIES GAN [140] is applied in three use cases: (a) hydropower plant, (b) power plant and (c) smart home. In the hydropower plant use case, ACC, TPR, FPR and F1 are equal to $ACC = 0.746$, $TPR = 0.978$, $FPR = 0.311$ and $F1 = 0.607$. Similarly, in the power plant use case, the detection efficiency of ARIES GAN is characterised by $ACC = 0.851, TPR = 0.982$, $FPR = 0.188$ and $F1 = 0.755$. Finally, in the smart home use case, the ACC of ARIES GAN reaches 0.859, while TPR, FPR and the F1 are equal to 0.976, 0.167 and 0.725. In contrast, in the substation use case, the LOF method is used, where $ACC = 0.873, TPR = 0.993, FPR = 0.157$ and $F1 = 0.759$. The comprehensive ML/DL comparative analysis of H-IDPS is provided in Appendix N.

6.4.3 V-IDPS Evaluation Results

Fig. 6.4 shows how the accuracy of the Active ResNet50-based CNN increases based on the updates of the new training dataset. In particular, the x-axis denotes the time when a new training dataset is created and used, following Algorithm 1. On the other hand, the y-axis indicates the new classification accuracy of the Active ResNet50-based CNN after each re-training process with the new training dataset. Consequently, each training process of the Active ResNet50-based CNN with a new training dataset corresponds to an accuracy value. Moreover, Table 6.1 summarises the evaluation metrics related to the pre-trained CNN models mentioned earlier after the last training process. The pre-trained

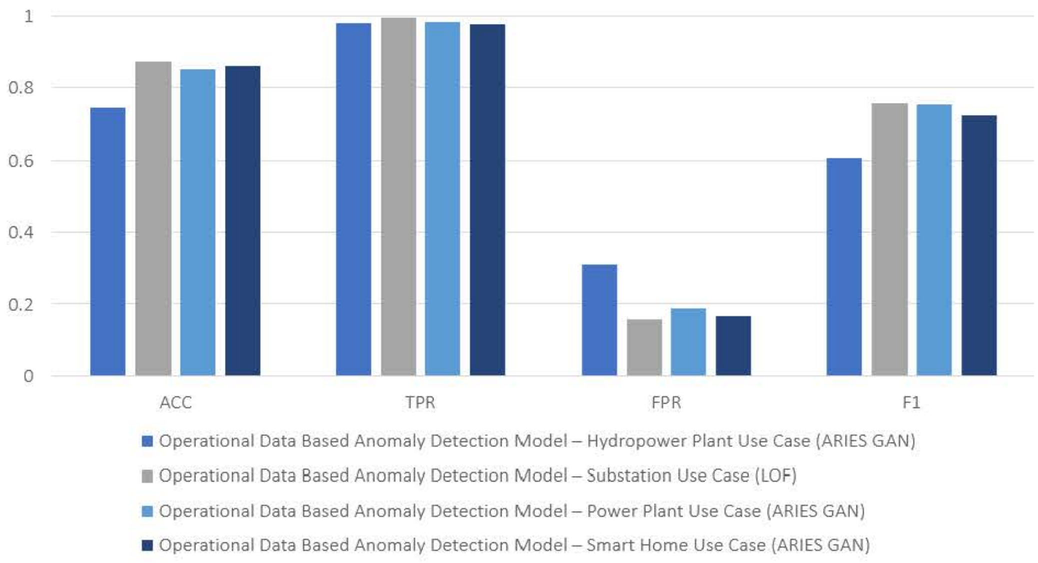


FIGURE 6.3: Evaluation Results of H-IDPS Anomaly Detection Models

CNN models of Table 6.1 were re-trained under the same conditions based on the Modbus/TCP intrusion detection dataset provided by this work. The best detection performance is accomplished by ResNet50: $Accuracy = 0.984$, $TPR = 0.885$, $FPR = 0.008$ and $F1score = 0.885$. In addition, Fig. 6.5 illustrates how the loss function related to Active ResNet50-based CNN ranges per epoch. Totally, 200 epochs were used. On the other side, NASNetMobile achieves the worst performance: $Accuracy = 0.961$, $TPR = 0.704$, $FPR = 0.020$ and $F1score = 0.709$. In general, it is worth mentioning that the efficiency of the pre-trained CNNs with the visual representations overcomes Modbus/TCP intrusion detection models of NF-IDPS.

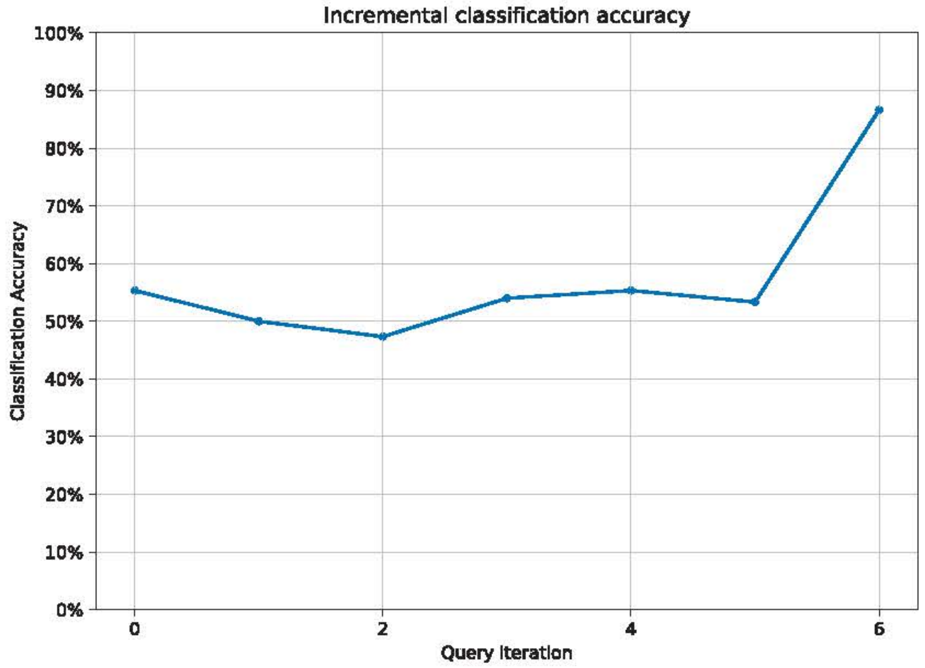


FIGURE 6.4: Accuracy increment of Active ResNet50-based CNN during the re-training phases

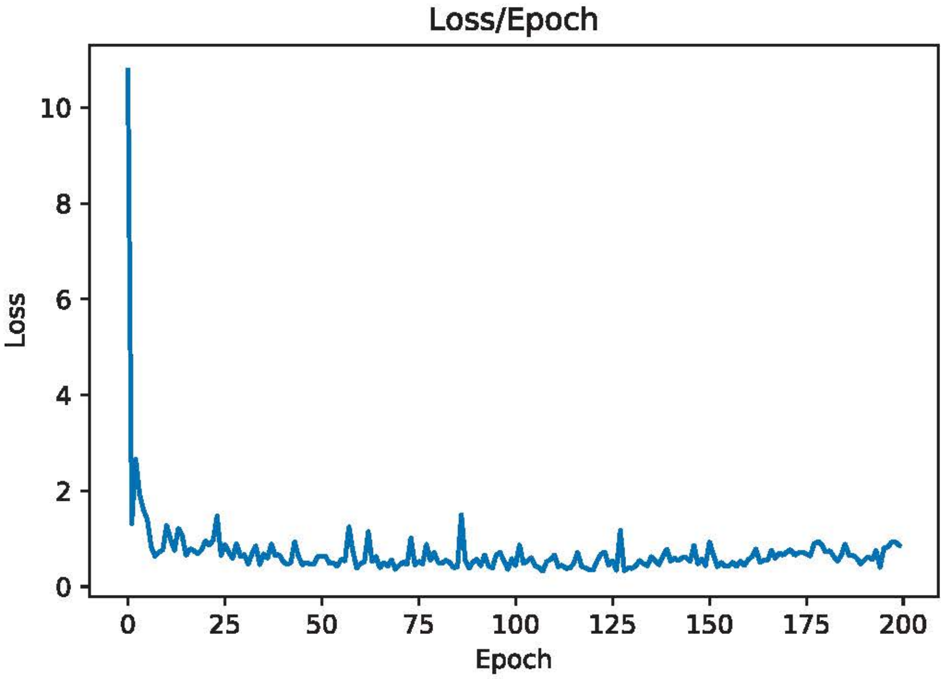


FIGURE 6.5: Range of Active-based ResNet50 CNN Loss

TABLE 6.1: Evaluation results of the pre-trained CNN models

Pre-trained CNN Model	**Accuracy**	**TPR**	**FPR**	**F1**
DenseNet121	0.975	0.814	0.013	0.814
DenseNet169	0.975	0.818	0.012	0.819
DenseNet201	0.979	0.837	0.010	0.843
EfficientNetB0	0.981	0.858	0.009	0.859
EfficientNetB7	0.962	0.697	0.018	0.713
MobileNet	0.981	0.862	0.009	0.862
MobileNetV2	0.980	0.850	0.010	0.850
NASNetLarge	0.964	0.714	0.017	0.728
NASNetMobile	0.961	0.704	0.020	0.709
ResNet50	**0.984**	**0.885**	**0.008**	**0.885**
ResNet50V2	0.980	0.854	0.010	0.854
ResNet101	0.981	0.864	0.009	0.864
ResNet101V2	0.980	0.853	0.010	0.853
ResNet152	0.982	0.865	0.009	0.865
ResNet152V2	0.978	0.805	0.009	0.831
VGG16	0.977	0.822	0.011	0.829
VGG19	0.981	0.863	0.009	0.863
Xception	0.975	0.806	0.012	0.812

6.4.4 NCME Evaluation Results: SDN-based Mitigation

Regarding the mitigation performance, Fig. 6.6-6.14 illustrate how the posterior probability $p(\mu|X,\tau)$ ranges based on the number of 5, 10, 20, 50, 100, 200, 500, 1000, 1500 and 2000 security events. In particular, we observe that the more security events, the taller and skinnier the Probability Density

Function (PDF) for each strategy is, thus increasing our belief for the proper action. In our experiments, s_1 seems to be the appropriate strategy, where NCME will instruct SDN-C to corrupt all the malicious Modbus/TCP network flows. However, the choice differs from one IIoT/SG environment to another IIoT/SG environment since the related costs for each strategy are different. Finally, Fig. 6.16 compares TS and UCB with respect to selecting the optimal strategy. In general, TS overcomes UCB though the accuracy values are relatively close to each other.

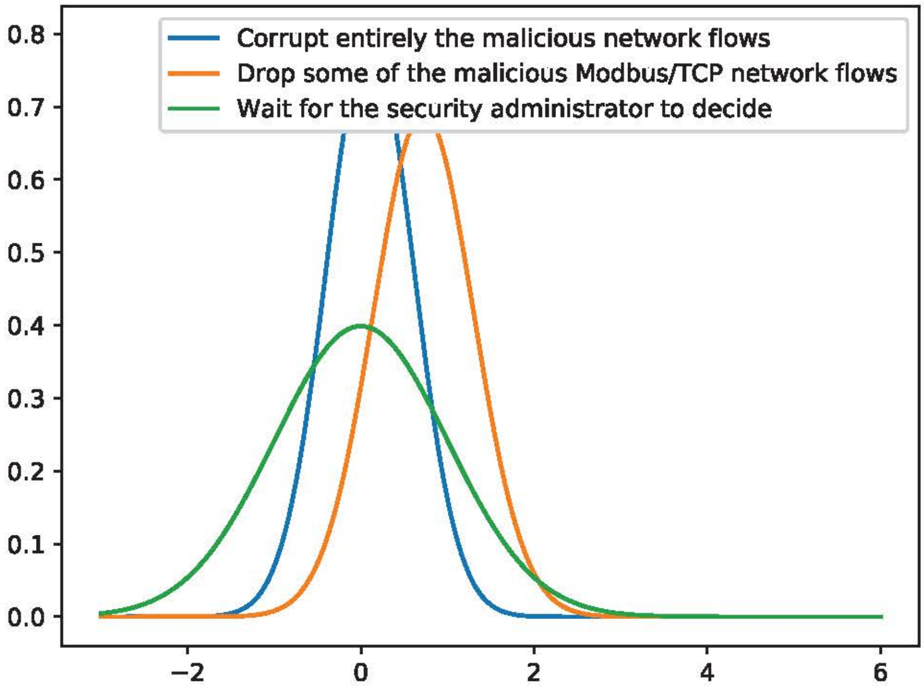


FIGURE 6.6: Posterior probability after 5 security alerts

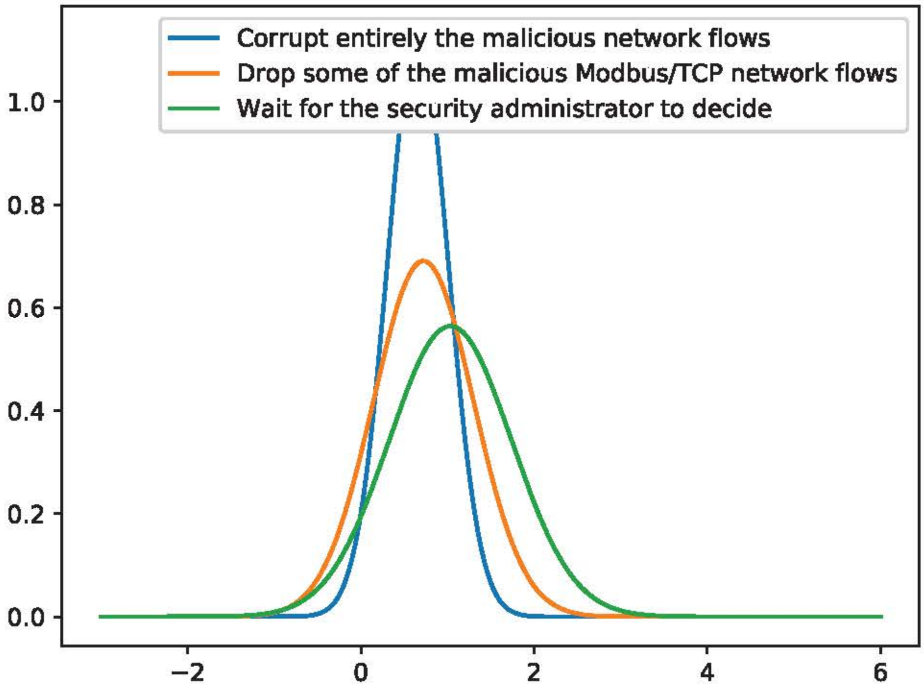


FIGURE 6.7: Posterior probability after 10 security alerts

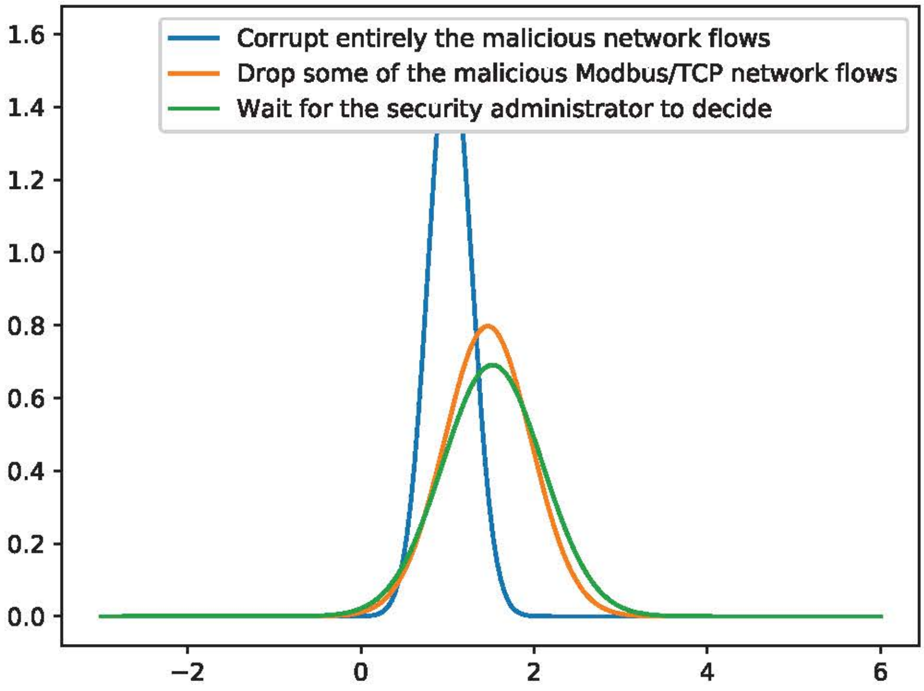


FIGURE 6.8: Posterior probability after 20 security alerts

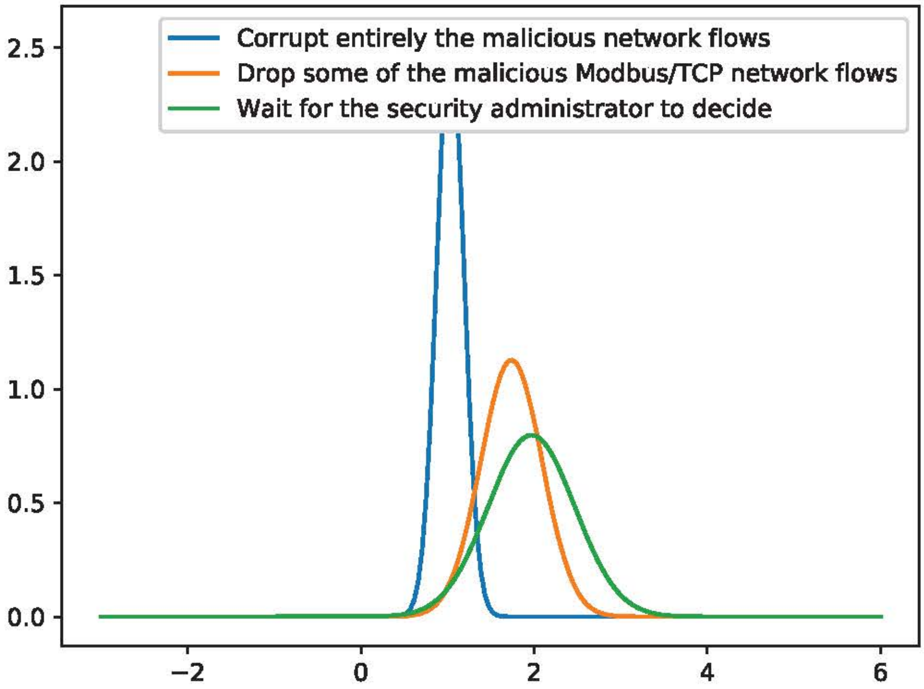


FIGURE 6.9: Posterior probability after 50 security alerts

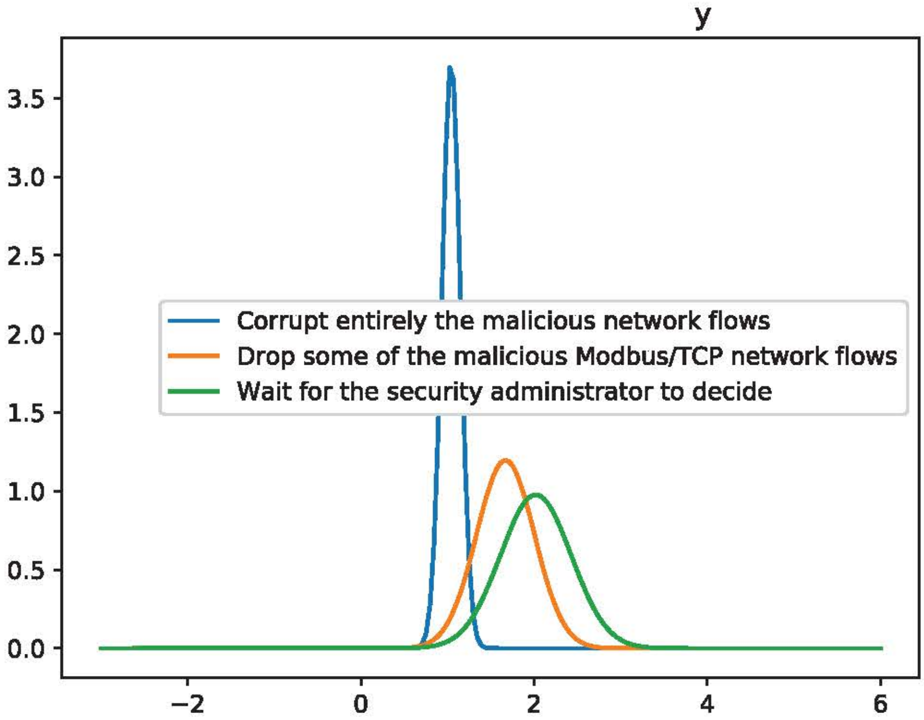


FIGURE 6.10: Posterior probability after 100 security alerts

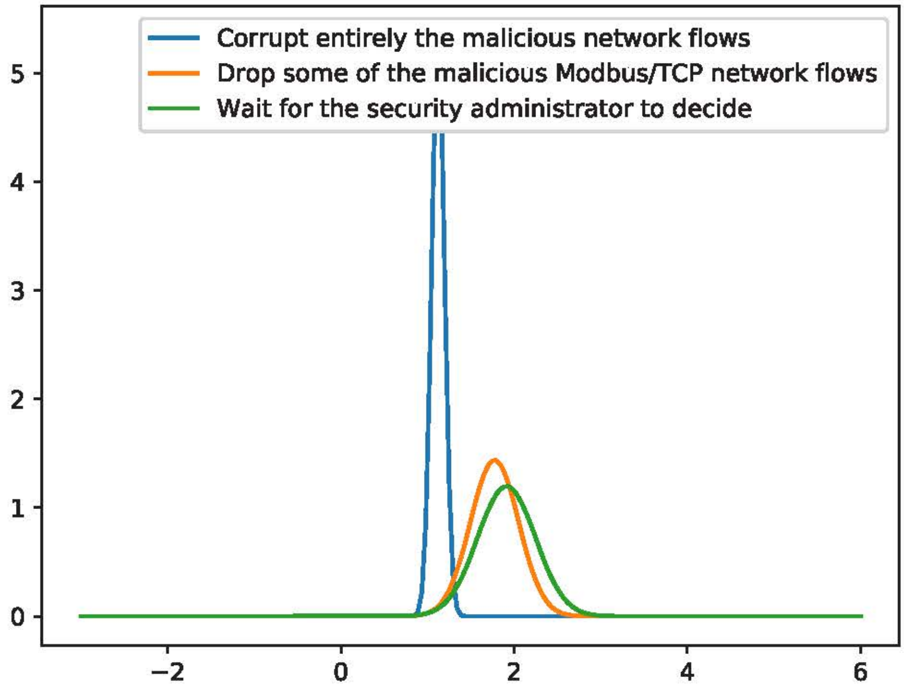


FIGURE 6.11: Posterior probability after 200 security alerts

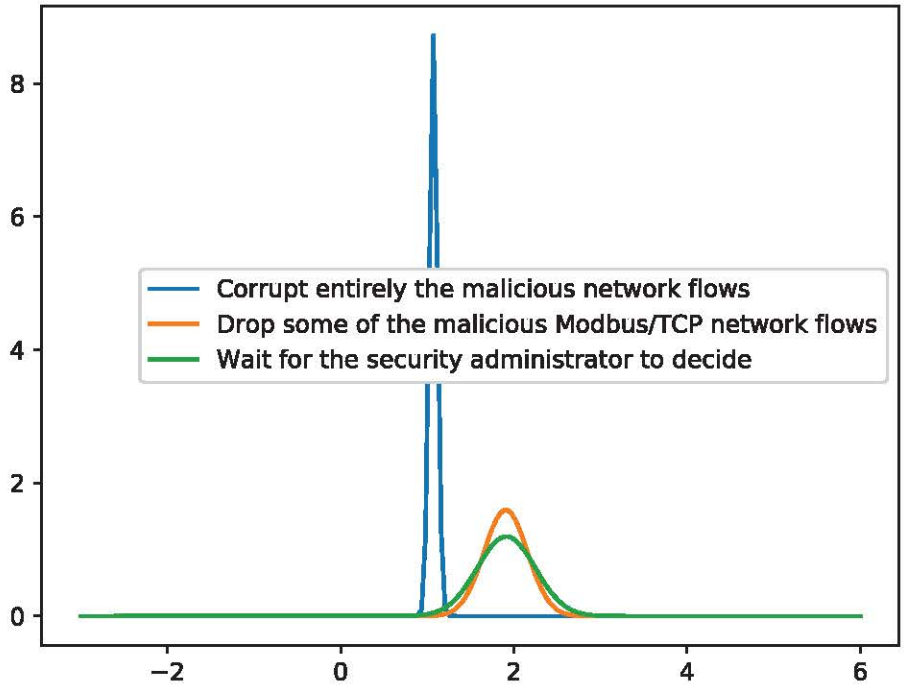


FIGURE 6.12: Posterior probability after 500 security alerts

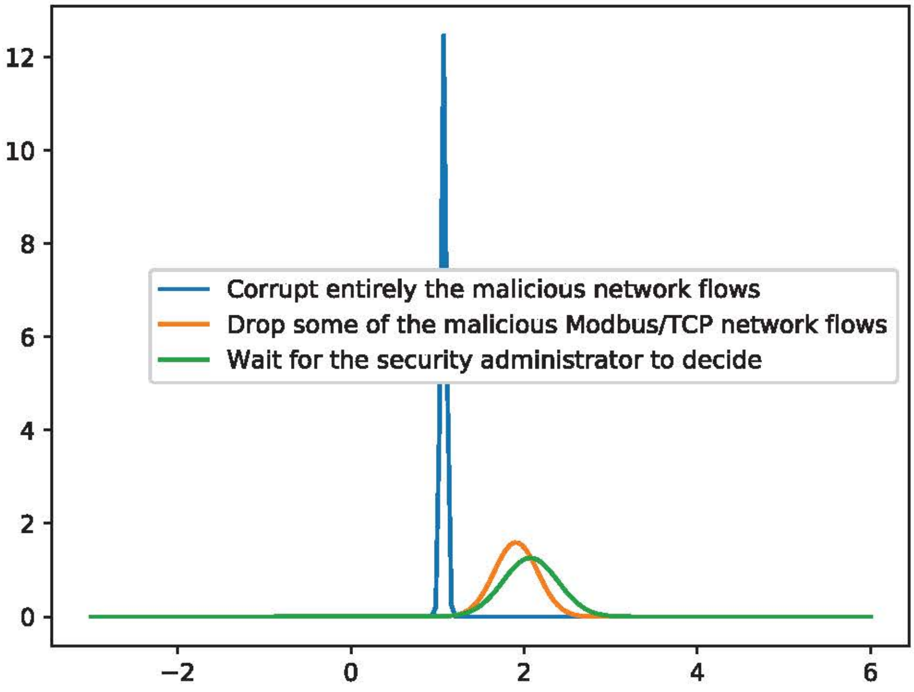


Figure 6.13: Posterior probability after 1000 security alerts

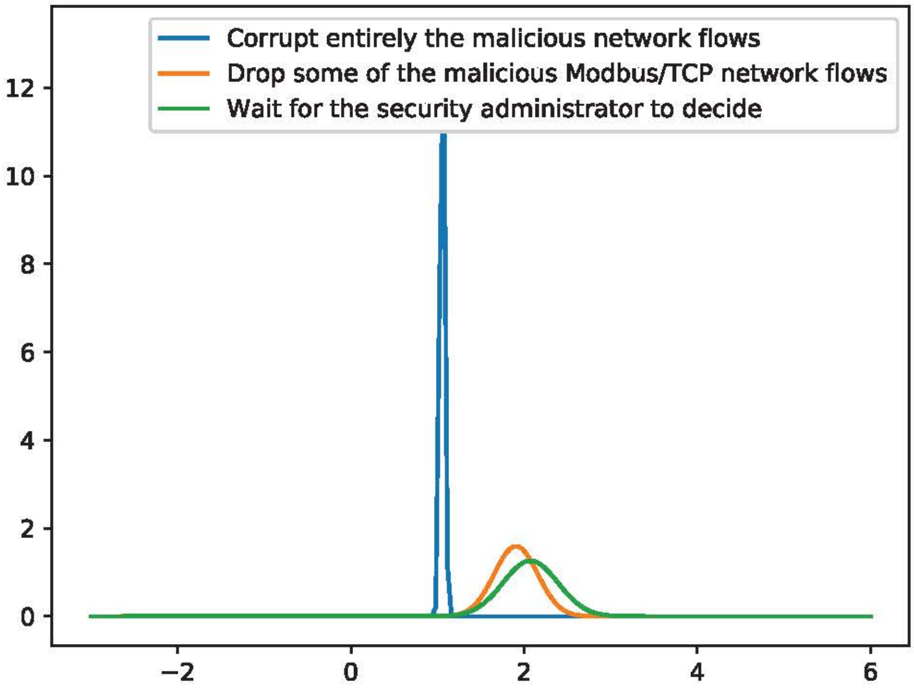


Figure 6.14: Posterior probability after 1500 security alerts

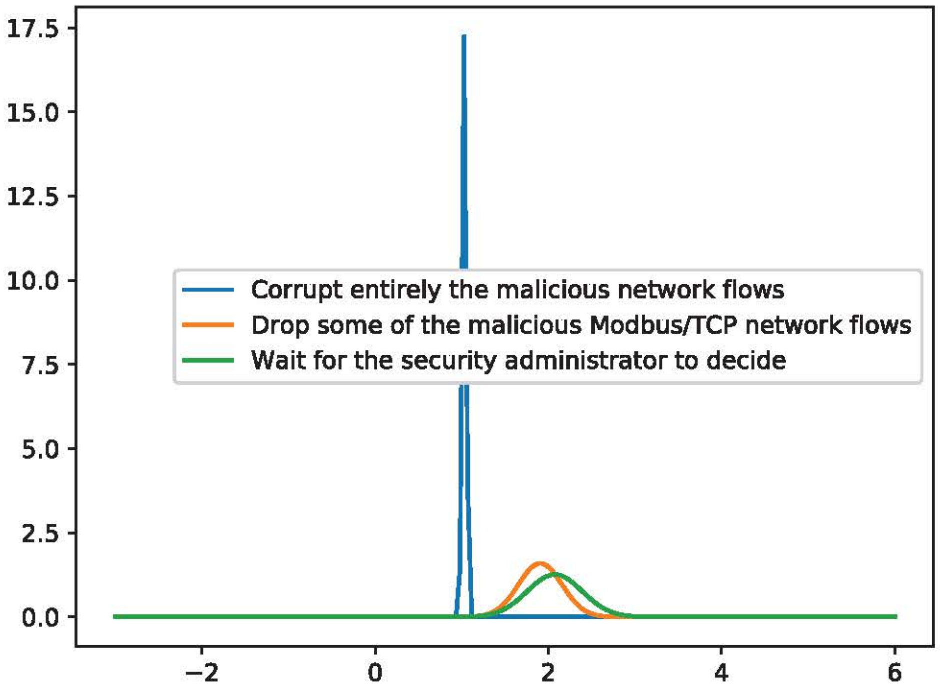


FIGURE 6.15: Posterior probability after 2000 security alerts

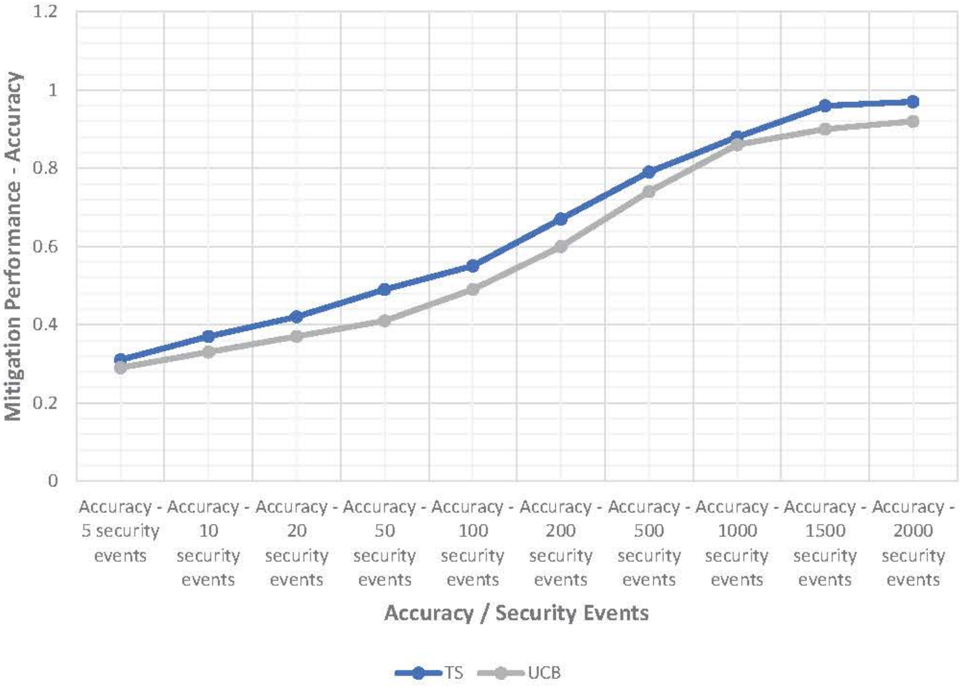


FIGURE 6.16: Comparison between TS and UCB with respect to the mitigation accuracy

6.4.5 NCME Evaluation Results: Honeypot Mitigation and Resilience

Three experiments were carried out in order to evaluate the effectiveness of the NCPE honeypot deployment mechanisms. Each experiment refers to the corresponding method (i.e., (a) NE solution, (b) maxmin-based honeypot deployment and (c) AI-powered honeypot deployment). The parameters that were used for the honeypot security game are provided in Table 6.2. In this experiment, the optimal strategy for the attacker and the defender is compared with 2000 random solutions in order to verify that the equilibrium indeed yields the maximum payoff, considering that the opponent always chooses the best strategy.

Table 6.2: Simulation parameters for the one-shot game

Parameter	**Value**
N_r	3
N_{max}	10
ϕ_{max}	1
$a_{\{1,2,3\}}$	$[0.76, 0.01, 0.10]$
$d_{\{1,2,3,4\}}$	$[0.03, 0.40, 0.45, 0.01]$
Random solutions for θ	2000
Random solutions for ϕ	2000

Figs. 6.17 and 6.18 verify that the payoffs of both players are optimal when the game reaches its equilibrium state. The red bullet in each graph denotes the payoff in the equilibrium state. In more detail, Fig. 6.17 shows that the payoff achieved in the equilibrium state (red bullet) is higher compared to 2000 random strategies ϕ, assuming that $N\theta$ remains at the optimal state. Similarly, the payoff achieved for the defender in Fig. 6.18 is higher compared to $2000N_{max}$ random combinations of $N\theta$, assuming that the opponent always chooses the best possible strategy. Moreover, it is notable that the payoffs follow a specific pattern when N remains constant and θ varies.

The second experiment examines the situation in which the game parameters do not result in equilibrium, thus, the defender applies a max-min analysis to maximise the worst-case scenario as described in Chapter 5. The parameters of this experiment are provided in table 6.3. The convex optimisation problem was solved by using the CVXPY Python library. Fig. 6.19 depicts the maximum worst-case payoff that corresponds to the solution received by the convex optimisation problem described in Chapter 5. This solution is compared to the worst-case payoffs that are received for different values of $N\theta$. The results prove that the defender successfully chooses the best possible strategy that yields the maximum payoff, assuming that the attacker always chooses the best strategy.

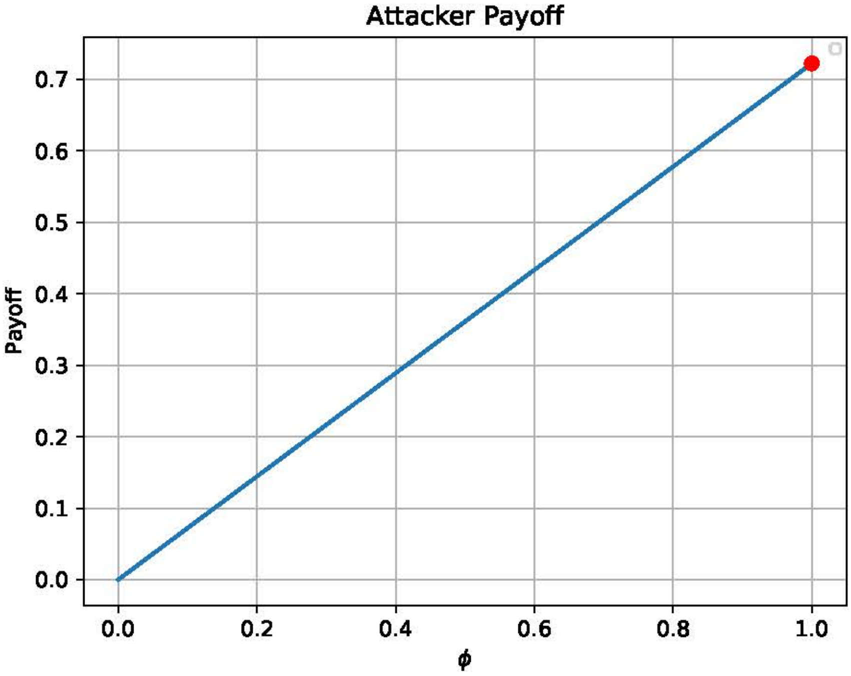


FIGURE 6.17: Payoff of the Attacker for different strategies

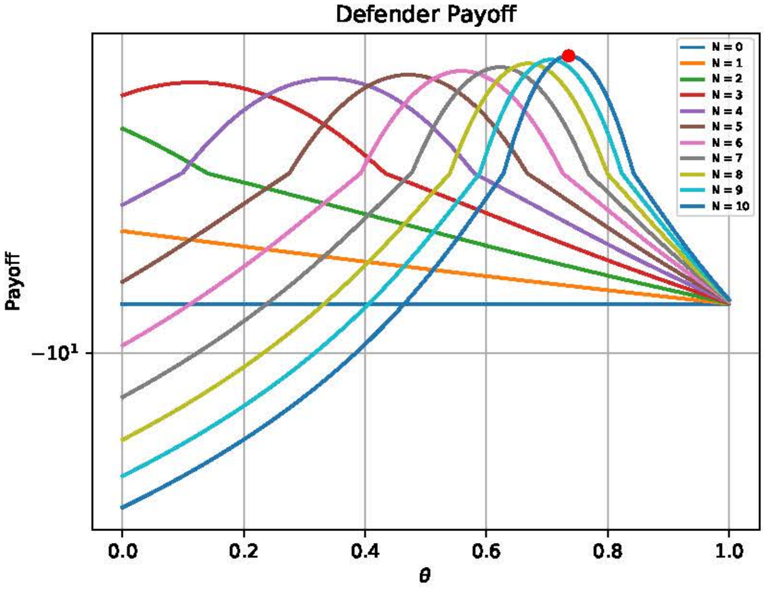


FIGURE 6.18: Payoff of the Defender for different strategies

TABLE 6.3: Simulation parameters for the one-shot game, when equilibrium does not exist

Parameter	Value
N_r	3
N_{max}	10
ϕ_{max}	1
$a_{\{1,2,3\}}$	$[0.81, 0.01, 0.06]$
$d_{\{1,2,3,4\}}$	$[0.31, 0.24, 0.81, 0.14]$
Random solutions for θ	2000

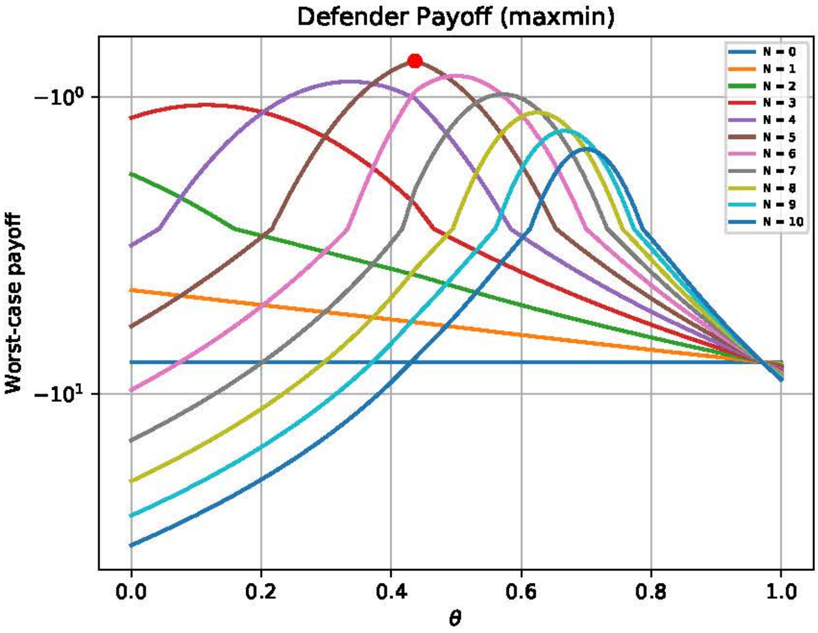


FIGURE 6.19: MaxMin Honeypot Deployment - Defender's worst-case payoff when equilibrium does not exist

Finally, in the last experiment, the AI-powered honeypot deployment mechanism is tested. For this purpose, a simulation experiment of 260 security alerts was utilised, using the dataset used by P.Radoglou-Grammatikis et al. in [134] and different values of e: $e = 0.1$, $e = 0.2$, $e = 0.3$, $e = 0.4$ and $e = 0.5$. Moreover, considering a constrained IIoT/SG environment, only 3 honeypots can be deployed. Thus, Fig. 6.20 shows the cumulative reward for the various values of e.

6.5 Chapter Summary

In this chapter, the evaluation results of the detection and mitigation solutions of the proposed SDN-enabled SIEM were discussed. For this purpose, different IIoT/SG evaluation environments and datasets

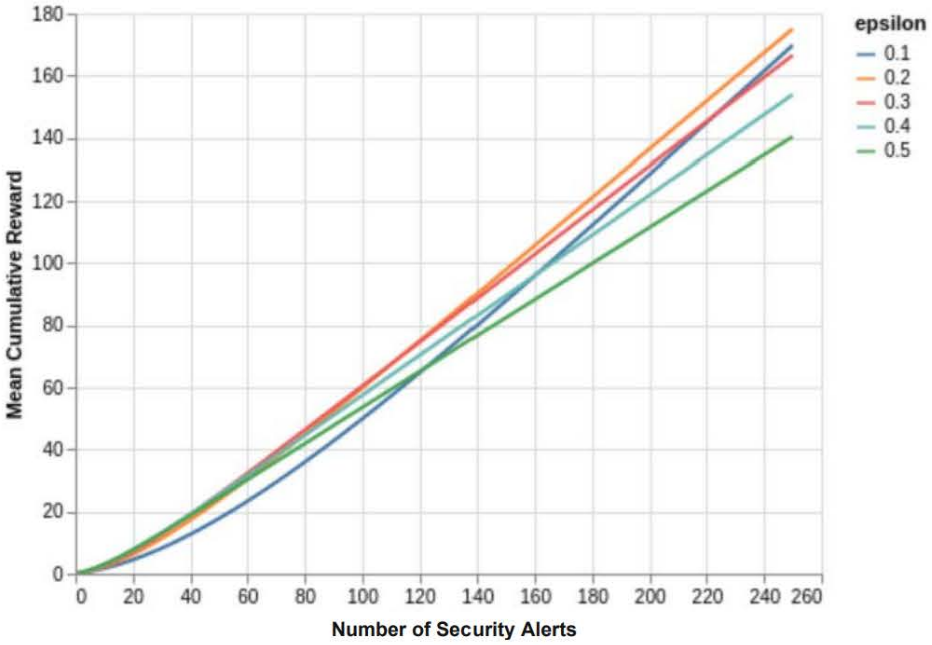


Figure 6.20: AI-powered Honeypot Deployment - Mean Cumulative Reward from 260 Security Alerts

were used while also appropriate simulation experiments took place. According to the evaluation results, the detection efficiency of (a) NF-IDPS, (b) H-IDPS and (c) V-IDPS is demonstrated. Their performance relies mainly on the characteristics of the respective data. Furthermore, the mitigation mechanisms of NCPE were tested. In particular, first, the effectiveness of NCPE was tested in terms of choosing the best mitigation strategy that will be executed by SDN-C. Next, the efficacy of the NCPE honeypot deployment mechanisms (i.e., (a) NE solution, (b) MaxMin-based Honeypot Deployment and (c) AI-powered Honeypot Deployment) was validated based on three relevant experiments. Consequently, it is evident that the proposed SDN-enabled SIEM is an overall efficient solution in order to secure and protect IIoT/SG environments.

Chapter 7

Conclusions & Future Work

Based on the previous chapters about the analysis of the IoT security issues and the proposed security solutions, this chapter summarises the concluding remarks of this PhD thesis, including also potential directions for future research work. In particular, after the summary of the main points in this PhD thesis, five research directions are discussed: (a) Intrusion and Anomaly Detection using Federated Learning (FL), (b) Correlation Mechanisms using Association Learning, (c) RL-based Mitigation Strategies, (d) SDN-powered recovery mechanisms using Graph Neural Networks (GNN) and (e) Explainable AI (XAI) Techniques for AI Detection and Mitigation Models.

7.1 Conclusions

It is evident that IoT is characterised by a wide range of cyberthreats that require the simultaneous evolution of the corresponding countermeasures. In particular, although the advent of IoT brings important advantages like real-time data collection, pervasive control, and improved productivity, it raises new cybersecurity and privacy issues due to the existence of legacy systems and the new security gaps of the new technologies. On the one hand, the legacy systems rely on insecure communication protocols, while zero-day vulnerabilities can characterise the new technologies of the IoT paradigm. In addition, it is worth mentioning that the IoT entities cannot fully support heavy security mechanisms in terms of computing resources. Accordingly, conventional security mechanisms are not fully effective. Moreover, the IoT ecosystems reflect an attractive goal of cyberattackers due to their valuable data and services, as in the case of CIs. Finally, it is noteworthy that the interconnected and independent nature of the IoT applications can result in security-related cascading effects with disastrous consequences. Therefore, according to the previous remarks, the primary objective of this PhD thesis was to investigate the security issues of IoT and provide appropriate detection and mitigation solutions, combining novel technologies. For this purpose, IIoT environments and particularly the SG (i.e., the largest IoT application) were utilised as a proof of concept, investigating and improving the security status of relevant use cases (such as substation, hydropower plant, power plant and smart home).

First, the IoT security requirements were investigated, taking into consideration the special characteristics of the IoT entities. The main security principles, such as confidentiality, integrity and availability, are obviously still valid as in any computing system; however, they are now characterised by new assumptions and constraints generated by the IoT entities, services and applications. Similarly, secondary security principles, such as authenticity and accountability, are also important, but they have to take into account the IoT features. In particular, several challenges should be considered, such as the interoperability of the security mechanisms, their resilience and scalability, the vast amount of data (generated by IoT entities), the limited computing and storage resources of the IoT entities and finally, the privacy of the involved users and systems. Furthermore, the automatic and autonomous nature of the IoT entities (in terms of interacting with external factors without any human intervention and control) can make the development and deployment of the relevant security solutions harder. It is worth mentioning that the previous challenges can differ and range depending on the functional and non-functional characteristics of each IoT environment. Therefore, the relevant security solutions should be selected and adjusted appropriately. In particular, based on the needs and the user requirements of each IoT environment, the relevant specifications will be produced, thus leading to the implementation of the appropriate security measures.

Second, based on the previous requirements, the next step was to identify the main IoT security threats. For this purpose, an IoT architectural model of four layers was studied, including (a) the Perception Layer, (b) the Communication Layer, (c) the Support Layer and (e) the Business Layer. For each layer, the corresponding threats were identified and analysed. In particular, the Perception Layer includes two main threats: (a) Natural Disasters and (b) Environmental Threats and Human-caused Physical Threats. Next, the Communication Layer comprises multiple threats, such as reconnaissance attacks, MITM attacks, DoS attacks and various kinds of routing attacks like sybil attacks, sinkhole attacks, wormhole attacks and hello flood attacks. Subsequently, the Support Layer is characterised by two main threats: (a) Unauthorised Access and Malicious Insiders and (b) Insecure Services and Unkown Risk Profile. Finally, the Business Layer also includes several threats, such as buffer overflow, backdoors, social engineering and web application attacks. It is worth mentioning that there are also defined cyberthreats that refer to more than one of the previous architectural layers, such as cryptanalytic attacks and malware. Moreover, for each threat, the corresponding CAPEC codes are provided, thus identifying particular cyberattacks. Finally, a special attention was given to cyberattacks against IIoT environments, focusing on the SG. In particular, complex cyberattacks and APT campaigns against the energy sector were investigated, utilising MITRE ATT&CK.

Next, based on the aforementioned IoT threats, an analysis of the available countermeasures follows.

For each layer of the proposed IoT architecture, the corresponding security countermeasures are summarised. A particular attention is given to the security mechanisms of the IoT communication protocols, such as IEEE 802.15.4, ZigBee, Z-Wave, BLE, LoRaWan, 6LoWPAN, RPL and DTLS. In more detail, the encryption, authentication and authorisation are examined. Based on this analysis, it is obvious that the current security solutions are effective; however, they cannot fully address unauthorised activities and malicious insiders. Moreover, it is challenging for the current solutions to handle the large amount of data generated by the IoT ecosystems. Accordingly, the presence of appropriate intrusion detection and mitigation mechanisms is necessary in terms of detecting, correlating and mitigating security alerts. Therefore, next, the role and types of the IDPS systems were investigated, taking into account the impact of novel technologies, such as AI, SDN, honeypots and SIEM. In addition, a comprehensive review of the IDPS system in the energy sector and the SG paradigm was conducted, identifying the security gaps of the current IDPS in this research area. More specifically, the current solutions do not fully consider the operational characteristics of the IIoT/SG environments in terms of their industrial communications protocols. Instead, they adopt ML and DL models, utilising pre-existing datasets that are not related to IIoT/SG. It is noteworthy that such datasets are rarely available due to their sensitive nature. Moreover, given the vast amount of security events, appropriate correlation mechanisms are necessary. However, they are not adequately supported by the existing solutions. Although existing SIEM systems, such XL-SIEM, AlienVault OSSIM and IBM QRadar, include correlation rules and directives, they usually refer to security events related to the traditional computing systems. On the other side, regarding the mitigation of the various attacks and anomalies, first traditional measures like firewall systems cannot fully address the large number of security events and alerts that can be generated in IIoT/SG ecosystems. SDN can solve this issue; however, the impact of the mitigation strategies should be further investigated, especially in the case of IIoT/SG environments, since they can result in more devastating effects than the actual attacks. Finally, after the mitigation procedures, the resilience of the underlying IIoT/SG infrastructures should be further enhanced and guaranteed, considering the operational characteristics of the IIoT/SG environments. Both production and research honeypots can be used for this purpose; however, they should be orchestrated appropriately.

Based on the previous analysis, this PhD thesis presents an SDN-enabled SIEM system for IIoT/SG environments, combining AI, SDN and honeypots. In particular, the proposed SDN-enabled SIEM system includes three IDPS, namely (a) NF-IDPS, (b) H-IDPS and (c) V-IDPS, that can recognise successfully a wide range of cyberattacks and anomalies against IIoT/SG environments. Next, NCME undertakes to normalise, correlate and mitigate the various security events, taking advantage of AI and SDN. Finally, NCME undertakes to further protect the target infrastructure by recalculating how many production honeypots can be used. For this purpose, a honeypot security game was defined, considering two players: (a) attacker(s) and (b) defender(s). For the proposed game, NE was identified, while when NE is not available, two alternative solutions are provided: (a) MaxMin-based Honeypot Deployment and (b) AI-powered Honeypot Deployment.

More specifically, first, NF-IDPS is capable of detecting a wide range of cyberattacks and anomalies against several IoT/SG protocols, such as Modbus/TCP, DNP3, IEC 60870-5-104, IEC 61850, HTTP and SSH. For each of the previous protocols, the corresponding cyberattacks were identified and analysed based on the attributes and functionality of each protocol. Next, collaborative ML/DL-based intrusion and anomaly detection models were implemented. For this purpose, appropriate intrusion detection datasets were implemented and combined with existing ones. On the other hand, H-IDPS focuses on detecting anomalies, utilising operational data (i.e., time series electricity measurements). In particular, H-IDPS focuses on three IIoT/SG proof of concept use cases, including (a) substation environments, (b) hydropower-plant environments, (c) power-plant environments and (d) smart home environments. Based on these operational data, appropriate ML/DL-based anomaly detection models were trained and implemented for each use case. Finally, V-IDPS focuses on Modbus/TCP threats, combining binary visual representations and pre-trained ResNet50. However, in a similar manner, V-IDPS can be used with other industrial communication protocols. It is worth mentioning that V-IDPS incorporates a self-active learning mechanism, which allows the retraining of ResNet50 during the inference mode of V-IDPS. According to the evaluation analysis of the previous components in Chapter 6, their detection efficiency is demonstrated.

On the other side, for the normalisation process, NCME adopts the AlienVault OSSIM format, while custom security rules are used to associate the normalised security events with each other. In particular, these security rules refer to Modbus/TCP cyberattacks; however, similar rules can be specified and used with other industrial communication protocols and cyberattacks. Subsequently, the mitigation process relies on SDN and honeypots. Through TS, NCME chooses first the appropriate mitigation strategy that will be applied by SDN-C. Next, NCME can use NE (if available), the maxmin analysis or epsilon-greedy in order to select the appropriate number of honeypots to be deployed in the underlying IIoT/SG infrastructure, thus increasing its resilience. Chapter 6 describes in detail the proposed mitigation mechanisms, while the evaluation analysis in Chapter 7 demonstrates their efficiency.

7.2 Future Work

According to the previous detection and mitigation mechanisms, first, future research efforts can focus on detecting and mitigating cyberattacks against other application-layer IIoT communication protocols, such as Profinet, Profibus [168], EtherCAT [4], MQTT [15], Advanced Message Queuing Protocol (AMQP) [127], Constrained Application Protocol (COAP) [217] and Websocket [100]. For the detection process, novel AI mechanisms can be used, such as FL [19, 27, 149]. In particular, FL is evolving as the next big step of AI, ensuring the data privacy of the underlying infrastructures. In particular, by taking full advantage of encryption methods like homomorphic encryption, secret sharing, secure multiparty computation and differential privacy, current FL implementations can protect the data required for the training procedure. Therefore, the knowledge from different infrastructures can be used without distributing sensitive data. Both centralised and decentralised FL architectures can be used. In the first case, there are two main components: (a) Federated Server and (b) Federated Clients. The Federated Server is responsible for organising the federated training procedure, while the federated clients undertake to train and generate the local AI models.

Second, future work can focus on sophisticated correlation and mitigation mechanisms. In the first case, sophisticated mechanisms will associate the normalised security events with each other in an automated manner. For this purpose, MITRE ATT&CK [7] and association rule learning techniques [125] can be used, such as the Eclat and Apriori algorithms. Therefore, APT campaigns could be detected in a timely manner. On the other hand, it is evident that SDN can adequately support the mitigation services. However, the current works do not consider the security issues of SDN. For instance, a single-point failure related to the presence of the SDN controller is possible. In addition, it is worth mentioning that some SDN-based mitigation actions can lead to more catastrophic effects. For instance, the termination of a network flow can further impact the target system rather than a reconnaissance attack. Consequently, a solution with multiple SDN controllers synchronised and coordinated with each other is necessary. Moreover, additional RL techniques, such as Deep Q Learning, Deep Deterministic Policy Gradient (DDPG) and Twin-Delayed DDPG, can be used to optimise the mitigation process. Finally, new honeypots and digital twins can be implemented in order to hide and protect the actual assets. To this end, AI generative techniques, such as GAN, can be used.

After detecting and mitigating the various cyberattacks, sophisticated recovery mechanisms can start. For this purpose, SDN can also be used to recover and reconnect the network connections. However, despite the benefits of SDN, such as more effective network administration, improved dependability, cost savings, and faster scalability, a lack of advanced network modelling functions do not allow SDN to automate this process. In particular, efficient Key Performance Indicators (KPIs), such as latency, jitter and loss, cannot be accomplished. Many works, such as [101, 160, 190, 205, 216] examine these issues. Nevertheless, the rapid progression of SDN allows resilient and scalable interconnecting environments based on application-layer services. Consequently, based on the aforementioned remarks, SDN can be used to restore and fully reconnect the network topology schema, utilising GNNs that can understand the complicated relationships between routing, network topology, and input traffic to produce a precise estimation of per-source/destination per-packet delay distribution and loss.

Finally, despite the fact that AI can support the decisions about the security status, explainability mechanisms are required so that the security administrator(s) can trust the decisions of the AI models. Many studies already investigate such solutions, utilising XAI techniques. Characteristic examples are [28, 95, 96, 150, 150]. However, various XAI layers are necessary. The first layer can focus on the data used for the detection, mitigation and recovery solutions. The second layer can utilise feature engineering and selection methods, thus showing the importance of various features. Next, in the third layer, visualisation techniques can show how the AI models are trained. Subsequently, the fourth layer can use XAI methods, such as Interpretable Model-Agnostic Explanations (LIME), SHapley Additive exPlanation (SHAP), and Explain Like I am Five (ELI5) in order to interpret the decisions of the trained AI models. Finally, the last layer can focus on the security of the AI models, addressing adversarial attacks.

Appendix A

IoT Threats: A CAPEC Taxnonomy

The following table provides a summary of the various IoT threats discussed in Chapter 2 and corresponds them to the relevant CAPEC codes.

Table A.1: IoT Threats: A CAPEC Taxonomy

Layer	Threat	CAPEC Codes
Perception Layer	Natural Disasters and Environmental Threats	- CAPEC-547: Physical Destruction of Device or Component
Perception Layer	Human-caused Physical Threats	- CAPEC-74: Manipulating State - CAPEC-124: Shared Resource Manipulation - CAPEC-390: Bypassing Physical Security - CAPEC-401: Physically Hacking Hardware - CAPEC-402: Bypassing ATA Password Security - CAPEC-438: Modification During Manufacture - CAPEC-440: Hardware Integrity Attack - CAPEC-444: Development Alteration - CAPEC-452: Infected Hardware - CAPEC-516: Hardware Component Substitution During Baselining - CAPEC-520: Counterfeit Hardware Component Inserted During Product Assembly - CAPEC-521: Hardware Design Specifications Are Altered - CAPEC-522: Malicious Hardware Component Replacement - CAPEC-534: Malicious Hardware Update - CAPEC-547: Physical Destruction of Device or Component - CAPEC-583: Disabling Network Hardware - CAPEC-603: Blockage - CAPEC-607: Obstruction - CAPEC-624: Hardware Fault Injection Improperly - CAPEC-625: Mobile Device Fault Injection - CAPEC-638: Altered Component Firmware - CAPEC-671: Requirements for ASIC Functionality Maliciously Altered - CAPEC-672: Malicious Code Implanted During Chip Programming - CAPEC-679: Exploitation of Improperly Configured or Implemented Memory Protections

Perception Layer	Human-caused Physical Threats	- CAPEC-37: Retrieve Embedded Sensitive Data - CAPEC-176: Configuration/Environment Manipulation - CAPEC-391: Bypassing Physical Locks - CAPEC-392: Lock Bumping - CAPEC-393: Lock Picking - CAPEC-399: Cloning RFID Cards or Chips - CAPEC-400: RFID Chip Deactivation or Destruction - CAPEC-413: Pretexting via Tech Support - CAPEC-457: USB Memory Attacks - CAPEC-507: Physical Theft - CAPEC-626: Smudge Attack - CAPEC-627: Counterfeit GPS Signals - CAPEC-663: Exploitation of Transient Instruction Execution - CAPEC-681: Exploitation of Improperly Controlled Hardware Security Identifiers
Communication Layer	Reconnaissance Attacks	- CAPEC-85: AJAX Footprinting - CAPEC-149: Explore for Predictable Temporary File Names - CAPEC-169: Footprinting - CAPEC-287: TCP SYN Scan - CAPEC-290: Enumerate Mail Exchange (MX) Records - CAPEC-291: DNS Zone Transfers - CAPEC-292: Host Discovery - CAPEC-293: Traceroute Route Enumeration - CAPEC-285: ICMP Echo Request Ping - CAPEC-294: ICMP Address Mask Request - CAPEC-295: Timestamp Request - CAPEC-296: ICMP Information Request - CAPEC-297: TCP ACK Ping - CAPEC-298: UDP Ping - CAPEC-299: TCP SYN Ping - CAPEC-300: Port Scanning - CAPEC-301: TCP Connect Scan - CAPEC-302: TCP FIN Scan - CAPEC-303: TCP Xmas Scan - CAPEC-304: TCP Null Scan - CAPEC-305: TCP ACK Scan - CAPEC-306: TCP Window Scan - CAPEC-307: TCP RPC Scan - CAPEC-308: UDP Scan - CAPEC-309: Network Topology Mapping - CAPEC-497: File Discovery - CAPEC-529: Malware-Directed Internal Reconnaissance - CAPEC-573: Process Footprinting - CAPEC-574: Services Footprinting - CAPEC-575: Account Footprinting - CAPEC-576: Group Permission Footprinting - CAPEC-577: Owner Footprinting - CAPEC-580: System Footprinting - CAPEC-581: Security Software Footprinting - CAPEC-612: WiFi MAC Address Tracking - CAPEC-613: WiFi SSID Tracking - CAPEC-618: Cellular Broadcast Message Request - CAPEC-619: Signal Strength Tracking - CAPEC-643: Identify Shared Files/Directories on System

Communication Layer	Reconnaissance Attacks	- CAPEC-646: Peripheral Footprinting
Communication Layer	Denial of Service Attacks	- CAPEC-2: Inducing Account Lockout - CAPEC-25: Forced Deadlock - CAPEC-125: Flooding - CAPEC-147: XML Ping of the Death - CAPEC-197: Exponential Data Expansion - CAPEC-221: Data Serialization External Entities Blowup - CAPEC-229: Serialized Data Parameter Blowup - CAPEC-230: Serialized Data with Nested Payloads - CAPEC-263: Force Use of Corrupted Files - CAPEC-271: Schema Poisoning - CAPEC-469: HTTP DoS - CAPEC-482: TCP Flood - CAPEC-486: UDP Flood - CAPEC-487: ICMP Flood - CAPEC-488: HTTP Flood - CAPEC-489: SSL Flood - CAPEC-490: Amplification - CAPEC-491: Quadratic Data Expansion - CAPEC-494: TCP Fragmentation - CAPEC-499: Android Intent Intercept - CAPEC-528: XML Flood - CAPEC-572: Artificially Inflate File Sizes - CAPEC-635: Alternative Execution Due to Deceptive Filenames - CAPEC-666: BlueSmacking
Communication Layer	Sybil Attacks	- CAPEC-161: Infrastructure Manipulation - CAPEC-481: Contradictory Destinations in Traffic Routing Schemes - CAPEC-582: Route Disabling - CAPEC-584: BGP Route Disabling - CAPEC-594: Traffic Injection - CAPEC-607: Obstruction
Communication Layer	Sinkhole Attacks	- CAPEC-161: Infrastructure Manipulation - CAPEC-481: Contradictory Destinations in Traffic Routing Schemes - CAPEC-582: Route Disabling - CAPEC-584: BGP Route Disabling - CAPEC-594: Traffic Injection - CAPEC-607: Obstruction
Communication Layer	Wormhole Attacks	- CAPEC-161: Infrastructure Manipulation - CAPEC-481: Contradictory Destinations in Traffic Routing Schemes - CAPEC-582: Route Disabling - CAPEC-584: BGP Route Disabling - CAPEC-594: Traffic Injection - CAPEC-607: Obstruction
Communication Layer	HELLO Flood Attacks	- CAPEC-161: Infrastructure Manipulation - CAPEC-481: Contradictory Destinations in Traffic Routing Schemes - CAPEC-582: Route Disabling - CAPEC-584: BGP Route Disabling - CAPEC-594: Traffic Injection - CAPEC-607: Obstruction

Communication Layer	Passive Network Traffic Analysis	- CAPEC-65: Sniff Application Code - CAPEC-117: Interception - CAPEC-157: Sniffing Attacks - CAPEC-158: Sniffing Network Traffic
Communication Layer	MiTM	- CAPEC-31: Accessing/Intercepting/Modifying HTTP Cookies - CAPEC-94: Adversary in the Middle (AiTM) - CAPEC-102: Session Sidejacking - CAPEC-117: Interception - CAPEC-192: Protocol Analysis - CAPEC-217: Exploiting Incorrectly Configured SSL - CAPEC-384: Application API Message Manipulation via Man-in-the-Middle - CAPEC-466: Leveraging Active Adversary in the Middle Attacks to Bypass Same Origin Policy - CAPEC-593: Session Hijacking
Support Layer	Unauthorised Access and Malicious Insiders	- CAPEC-34: HTTP Response Splitting - CAPEC-60: Reusing Session IDs - CAPEC-102: Session Sidejacking - CAPEC-114: Authentication Abuse - CAPEC-117: Interception - CAPEC-122: Privilege Abuse - CAPEC-180: Exploiting Incorrectly Configured Access Control Security Levels - CAPEC-212: Functionality Misuse - CAPEC-234: Hijacking a privileged process - CAPEC-248: Command Injection - CAPEC-402: Bypassing ATA Password Security - CAPEC-555: Remote Services with Stolen Credentials - CAPEC-594: Traffic Injection - CAPEC-629: Unauthorized Use of Device Resources - CAPEC-651: Eavesdropping - CAPEC-652: Use of Known Kerberos Credentials - CAPEC-653: Use of Known Windows Credentials
Support Layer	Insecure Services and Unknown Riks Profile	-
Business Layer	Buffer Overflow	- CAPEC-8: Buffer Overflow in an API Call - CAPEC-9: Buffer Overflow in Local Command-Line Utilities - CAPEC-10: Buffer Overflow via Environment Variables - CAPEC-14: Client-side Injection-induced Buffer Overflow - CAPEC-46: Overflow Variables and Tags - CAPEC-100: Overflow Buffers - CAPEC-24: Filter Failure through Buffer Overflow - CAPEC-42: MIME Conversion - CAPEC-44: Overflow Binary Resource File - CAPEC-45: Buffer Overflow via Symbolic Links - CAPEC-47: Buffer Overflow via Parameter Expansion - CAPEC-52: Embedding NULL Bytes - CAPEC-67: String Format Overflow in syslog() - CAPEC-77: Manipulating User-Controlled Variables - CAPEC-92: Forced Integer Overflow - CAPEC-123: Buffer Manipulation - CAPEC-256: SOAP Array Overflow

Business Layer	Backdoor	- CAPEC-206: Signing Malicious Code - CAPEC-438: Modification During Manufacture - CAPEC-443: Malicious Logic Inserted Into Product Software by Authorized Developer - CAPEC-444: Development Alteration - CAPEC-445: Malicious Logic Insertion into Product Software via Configuration Management Manipulation - CAPEC-446: Malicious Logic Insertion into Product Software via Inclusion of 3rd Party Component Dependency - CAPEC-511: Infiltration of Software Development Environment - CAPEC-523: Malicious Software Implanted - CAPEC-538: Open-Source Library Manipulation - CAPEC-539: ASIC With Malicious Functionality - CAPEC-558: Replace Trusted Executable - CAPEC-669: Alteration of a Software Update - CAPEC-670: Software Development Tools Maliciously Altered - CAPEC-673: Developer Signing Maliciously Altered Software - CAPEC-678: System Build Data Maliciously Altered
Business Layer	Social Engineering	- CAPEC-21: Exploitation of Trusted Identifiers - CAPEC-98: Phishing - CAPEC-103: Clickjacking - CAPEC-151: Identity Spoofing - CAPEC-154: Resource Location Spoofing - CAPEC-163: Spear Phishing - CAPEC-164: Mobile Phishing - CAPEC-173: Action Spoofing - CAPEC-178: Cross-Site Flashing - CAPEC-194: Fake the Source of Data - CAPEC-195: Principal Spoof - CAPEC-383: Harvesting Information via API Event Monitoring - CAPEC-406: Dumpster Diving - CAPEC-407: Pretexting - CAPEC-413: Pretexting via Tech Support - CAPEC-414: Pretexting via Delivery Person - CAPEC-415: Pretexting via Phone - CAPEC-416: Manipulate Human Behavior - CAPEC-417: Influence Perception - CAPEC-418: Influence Perception of Reciprocation - CAPEC-420: Influence Perception of Scarcity - CAPEC-421: Influence Perception of Authority - CAPEC-422: Influence Perception of Commitment and Consistency - CAPEC-423: Influence Perception of Liking - CAPEC-424: Influence Perception of Consensus or Social Proof - CAPEC-425: Target Influence via Framing - CAPEC-426: Influence via Incentives - CAPEC-427: Influence via Psychological Principles - CAPEC-428: Influence via Modes of Thinking - CAPEC-429: Target Influence via Eye Cues - CAPEC-433: Target Influence via The Human Buffer Overflow - CAPEC-434: Target Influence via Interview and Interrogation - CAPEC-506: Tapjacking - CAPEC-656: Voice Phishing

Business Layer	Social Engineering	- CAPEC-435: Target Influence via Instant Rapport - CAPEC-467: Cross Site Identification - CAPEC-543: Counterfeit Websites - CAPEC-544: Counterfeit Organizations - CAPEC-585: DNS Domain Seizure - CAPEC-611: BitSquatting - CAPEC-616: Establish Rogue Location - CAPEC-630: TypoSquatting - CAPEC-631: SoundSquatting - CAPEC-632: Homograph Attack via Homoglyphs - CAPEC-652: Use of Known Kerberos Credentials - CAPEC-667: Bluetooth Impersonation AttackS (BIAS)
Business Layer	Web Application Attacks	- CAPEC-7: Blind SQL Injection - CAPEC-18: XSS Targeting Non-Script Elements - CAPEC-32: XSS Through HTTP Query Strings - CAPEC-62: Cross Site Request Forgery - CAPEC-63: Cross-Site Scripting (XSS) - CAPEC-66: SQL Injection - CAPEC-84: XQuery Injection - CAPEC-86: XSS Through HTTP Headers - CAPEC-101: Server Side Include (SSI) Injection - CAPEC-107: Cross Site Tracing - CAPEC-108: Command Line Execution through SQL Injection - CAPEC-109: Object Relational Mapping Injection - CAPEC-110: SQL Injection through SOAP Parameter Tampering - CAPEC-193: PHP Remote File Inclusion - CAPEC-198: XSS Targeting Error Pages - CAPEC-199: XSS Using Alternate Syntax - CAPEC-209: XSS Using MIME Type Mismatch - CAPEC-242: Code Injection - CAPEC-243: XSS Targeting HTML Attributes - CAPEC-244: XSS Targeting URI Placeholders - CAPEC-245: XSS Using Doubled Characters - CAPEC-247: XSS Using Invalid Characters - CAPEC-248: Command Injection - CAPEC-251: Local Code Inclusion - CAPEC-252: PHP Local File Inclusion - CAPEC-253: Remote Code Inclusion - CAPEC-465: Transparent Proxy Abuse - CAPEC-470: Expanding Control over the Operating System from the Database - CAPEC-588: DOM-Based XSS - CAPEC-591: Reflected XSS - CAPEC-592: Stored XSS
Multiple Layers	Cryptanalytic Attacks	- CAPEC-20: Encryption Brute Forcing - CAPEC-97: Cryptanalysis - CAPEC-49: Password Brute Forcing - CAPEC-112: Brute Force - CAPEC-192: Protocol Analysis - CAPEC-463: Padding Oracle Crypto Attack - CAPEC-475: Signature Spoofing by Improper Validation - CAPEC-485: Signature Spoofing by Key Recreation - CAPEC-608: Cryptanalysis of Cellular Encryption

Multiple Layers	Malware Attacks	- CAPEC-206: Signing Malicious Code - CAPEC-270: Modification of Registry Run Keys - CAPEC-387: Navigation Remapping To Propagate Malicious Content - CAPEC-529: Malware-Directed Internal Reconnaissance - CAPEC-542: Targeted Malware - CAPEC-549: Local Execution of Code - CAPEC-550: Install New Service - CAPEC-551: Modify Existing Service - CAPEC-552: Install Rootkit - CAPEC-556: Replace File Extension Handlers - CAPEC-558: Replace Trusted Executable - CAPEC-564: Run Software at Logon - CAPEC-568: Capture Credentials via Keylogger - CAPEC-579: Replace Winlogon Helper DLL - CAPEC-642: Replace Binaries

Appendix B

APT Campaigns against the Energy Sector

The following table summarises the main APT campaigns against the energy sector based on MITRE ATT&CK. For each APT campaign, a brief description and the relevant techniques and software are given.

Table B.1: Cybersecurity Incidents and APT Campaigns against Energy-related Organisations

APT Campaign	Description	Techniques	Software
APT33	Suspected Iranian group executing cyberattacks against energy-related organisations from US, Saudi Arabia, and South Korea	- T107.001: Application Layer Protocol: Web Protocols - T1560.001: Archive Collected Data: Archive via Utility - T1547.001: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder - T1110.003: Brute Force: Password Spraying - T1059.001: Command and Scripting Interpreter: PowerShell - T1059.005: Command and Scripting Interpreter: Visual Basic - T1555.003: Credentials from Web Browsers - T1132.001: Data Encoding: Standard Encoding - T1573.001: Encrypted Channel: Symmetric Cryptography	- S0363: Empire - S0095: FTP - S0349: LaZagne - S0002: Mimikatz - S0336: NanoCore - S0039: Net - S0198: NETWIRE

APT33	Suspected Iranian group executing cyberattacks against energy-related organisations from US, Saudi Arabia, and South Korea	- T1546.003: Event Triggered Execution: Windows Management Instrumentation Event Subscription - T1048.003: Exfiltration Over Alternative Protocol: Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol - T1203: Exploitation for Client Execution - T1068: Exploitation for Privilege Escalation - T1105: Ingress Tool Transfer - T1040: Network Sniffing - T1571: Non-Standard Port - T1027: Obfuscated Files or Information - T1588.002: Obtain Capabilities: Tool - T1003.001: OS Credential Dumping: LSASS Memory - T1003.004: OS Credential Dumping: LSA Secrets - T1003.005: OS Credential Dumping: Cached Domain Credentials - T1566.001: Phishing: Spearphishing Attachment - T1566.002: Phishing: Spearphishing Link - T1053.005: Scheduled Task/Job: Scheduled Task - T1552.001: Unsecured Credentials: Credentials In Files - T1552.006: Unsecured Credentials: Group Policy Preferences - T1204.001: User Execution: Malicious Link - T1204.002: User Execution: Malicious File - T1078: Valid Accounts - T1078.004: Cloud Accounts	- S0129: AutoIt backdoor - S0378: PoshC2 - S0194: PowerSploit - S0371: POWERTON - S0192: Pupy - S0358: Ruler - S0380: StoneDrill - S0199: TURNEDUP
Operation Wocao	Chinese APT group targeting government and energy related organisations	- T1087.002: Account Discovery: Domain Account - T1560.001: Archive Collected Data: Archive via Utility - T1119: Automated Collection - T1115: Clipboard Data - T1059.001: Command and Scripting Interpreter: PowerShell - T1059.003: Command and Scripting Interpreter: Windows Command Shell - T1059.005: Command and Scripting Interpreter: Visual Basic - T1059.006: Command and Scripting Interpreter: Python - T1555.005: Credentials from Password Stores: Password Managers	- S0521: BloodHound - S0105: dsquery - S0357: Impacket - S0002: Mimikatz - S0104: netstat - S0194: PowerSploit

Operation Wocao	Chinese APT group targeting government and energy related organisations	- T1005: Data from Local System - T1001: Data Obfuscation - T1074.001: Data Staged: Local Data Staging - T1573.002: Encrypted Channel: Asymmetric Cryptography - T1041: Exfiltration Over C2 Channel - T1190: Exploit Public-Facing Application - T1133: External Remote Services - T1083: File and Directory Discovery - T1083: File and Directory Discovery - T1562.004: Impair Defenses: Disable or Modify System Firewall - T1070.001: Indicator Removal on Host: Clear Windows Event Logs - T1070.004: Indicator Removal on Host: File Deletion - T1105: Ingress Tool Transfer - T1056.001: Input Capture: Keylogging - T1570: Lateral Tool Transfer - T1112: Modify Registry - T1106: Native API - T1046: Network Service Scanning - T1135: Network Share Discovery - T1095: Non-Application Layer Protocol - T1027: Obfuscated Files or Information - T1027.005: Indicator Removal from Tools - T1003.001: OS Credential Dumping: LSASS Memory - T1003.006: OS Credential Dumping: DCSync - T1120: Peripheral Device Discovery - T1069.001: Permission Groups Discovery: Local Groups - T1057: Process Discovery - T1055: Process Injection - T1090: Proxy - T1090.001: Internal Proxy - T1090.003: Multi-hop Proxy - T1012: Query Registry - T1021.002: Remote Services: SMB/Windows Admin Shares	

Operation Wocao	Chinese APT group targeting government and energy related organisations	- T1018: Remote System Discovery - T1053.005: Scheduled Task/Job: Scheduled Task - T1505.003: Server Software Component: Web Shell - T1518: Software Discovery - T1518.003: Steal or Forge Kerberos Tickets: Kerberoasting - T1082: System Information Discovery - T1016: System Network Configuration Discovery - T1049: System Network Connections - T1033: System Owner/User Discovery - T1007: System Service Discovery - T1569.002: System Services: Service Execution - T1124: System Time Discovery - T1111: Two-Factor Authentication Interception - T1552.004: Unsecured Credentials: Private Keys - T1078: Valid Accounts - T1078.002: Domain Accounts - T1078.003: Local Accounts - T1078.003: Windows Management Instrumentation	
Dragonfly	Espionage APT campaign discovered in 2011, targeting energy-related companies.	- T1189: Drive-by Compromise - T1588.002: Obtain Capabilities: Tool - T1566: Phishing - T1195.002: Supply Chain Compromise: Compromise Software Supply Chain	- S0093: Backdoor.Oldrea - S0488: CrackMapExec - S0002: Mimikatz - S0029: PsExec - S0094: Trojan.Karagany
Tonto Team	Espionage APT campaign discovered in 2011, targeting energy-related companies from South Korea, Japan, Taiwan, US and other Asian and eastern European countries	- T1059.001: Command and Scripting Interpreter: PowerShell - T1059.006: Command and Scripting Interpreter: Python - T1203: Exploitation for Client Execution - T1068: Exploitation for Privilege Escalation - T1210: Exploitation of Remote Services - T1574.001: Hijack Execution Flow: DLL Search Order Hijacking - T1105: Ingress Tool Transfer - T1056.001: Input Capture: Keylogging - T1135: Network Share Discovery	- S0268: Bisonal - S0008: gsecdump - S0349: LaZagne - S0002: Mimikatz - S0590: NBTscan - S0596: ShadowPad

Tonto Team	Espionage APT campaign discovered in 2011, targeting energy-related companies from South Korea, Japan, Taiwan, US and other Asian and eastern European countries	- T1003: OS Credential Dumping - T1069.001: Permission Groups Discovery: Local Groups - T1566.001: Phishing: Spearphishing Attachment - T1090.002: Proxy: External Proxy - T1505.003: Server Software Component: Web Shell - T1204.002: User Execution: Malicious File - T1059.001: Command and Scripting Interpreter: PowerShell - T1059.006: Command and Scripting Interpreter: Python - T1203: Exploitation for Client Execution - T1068: Exploitation for Privilege Escalation - T1210: Exploitation of Remote Services - T1574.001: Hijack Execution Flow: DLL Search Order Hijacking - T1105: Ingress Tool Transfer - T1056.001: Input Capture: Keylogging - T1135: Network Share Discovery - T1059.001: Command and Scripting Interpreter: PowerShell - T1059.006: Command and Scripting Interpreter: Python - T1203: Exploitation for Client Execution - T1068: Exploitation for Privilege Escalation - T1210: Exploitation of Remote Services - T1574.001: Hijack Execution Flow: DLL Search Order Hijacking - T1105: Ingress Tool Transfer - T1056.001: Input Capture: Keylogging - T1135: Network Share Discovery	
OilRig	OilRig is a potential Iranian threat group targeting a wide range of critical domains, such as energy, telecommunications and finance.	- T1087.001: Account Discovery: Local Account - T1087.002: Account Discovery: Domain Account - T1071.001: Application Layer Protocol: Web Protocols - T1071.004: Application Layer Protocol: DNS - T1119: Automated Collection - T1110: Brute Force - T1059: Command and Scripting Interpreter - T1059.001: PowerShell	- S0360: BONDUPDATER - S0160: certutil - S0095: FTP - S0170: Helminth - S0100: ipconfig - S0189: ISMInjector - S0349: LaZagne

OilRig	OilRig is a potential Iranian threat group targeting a wide range of critical organisations, related to energy, telecommunications and finance.	- T1059.003: Windows Command Shell - T1059.005: Visual Basic - T1555: Credentials from Password Stores - T1555.003: Credentials from Web Browsers - T1555.004: Windows Credential Manager - T1140: Deobfuscate/Decode Files or Information - T1573.002: Encrypted Channel: Asymmetric Cryptography - T1048.003: Exfiltration Over Alternative Protocol: Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol - T1133: External Remote Services - T1008: Fallback Channels - T1070.004: Indicator Removal on Host: File Deletion - T1105: Ingress Tool Transfer - T1056.001: Input Capture: Keylogging - T1036: Masquerading - T1046: Network Service Scanning - T1027: Obfuscated Files or Information - T1027.005: Indicator Removal from Tools - T1137.004: Office Application Startup: Outlook Home Page - T1003.001: OS Credential Dumping: LSASS Memory - T1003.004: OS Credential Dumping: LSA Secrets - T1003.005: OS Credential Dumping: Cached Domain Credentials - T1201: Password Policy Discovery - T1120: Peripheral Device Discovery - T1069.001: Permission Groups Discovery: Local Groups - T1069.002: Permission Groups Discovery: Domain Groups - T1566.001: Phishing: Spearphishing Attachment - T1566.002: Phishing: Spearphishing Link - T1566.003: Phishing: Spearphishing via Service - T1053.005: Scheduled Task/Job: Scheduled Task - T1113: Screen Capture - T1505.003: Server Software Component: Web Shell - T1218.001: Signed Binary Proxy Execution: Compiled HTML File - T1082: System Information Discovery	- S0002: Mimikatz - S0039: Net - S0104: netstat - S0264: OopsIE - S0184: POWRUNER - S0029: PsExec - S0269: QUADAGENT - S0495: RDAT - S0075: Reg - S0258: RGDoor - S0185: SEASHARPEE - S0610: SideTwist - S0096: Systeminfo - S0057: Tasklist

OilRig	OilRig is a potential Iranian threat group targeting a wide range of critical organisations, related to energy, telecommunications and finance.	- T1016: System Network Configuration Discovery - T1049: System Network Connections Discovery - T1033: System Owner/User Discovery - T1007: System Service Discovery - T1552.001: Unsecured Credentials: Credentials In Files - T1204.001: User Execution: Malicious Link - T1204.002: User Execution: Malicious File - T1078: Valid Accounts - T1497.001: Virtualization/Sandbox Evasion: System Checks - T1047: Windows Management Instrumentation	
Sharpshooter	Sharpshooter is a cyber espionage APT discovered 2018, giving emphasis to nuclear, energy and financial-related organisations.	- T1547.001: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder - T1059.005: Command and Scripting Interpreter: Visual Basic - T1105: Ingress Tool Transfer - T1559.002: Inter-Process Communication: Dynamic Data Exchange - T1106: Native API - T1566.001: Phishing: Spearphishing Attachment - T1055: Process Injection - T1204.002: User Execution: Malicious File	- S0448: Rising Sun
APT19	APT19 is a Chinese group targetting a plethora of organisations related to energy, telecommunications, healthcare and finance.	- T1071.001: Application Layer Protocol: Web Protocols - T1547.001: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder - T1059: Command and Scripting Interpreter - T1059.001: PowerShell - T1543.003: Create or Modify System Process: Windows Service - T1132.001: Data Encoding: Standard Encoding - T1140: Deobfuscate/Decode Files or Information - T1189: Drive-by Compromise - T1564.003: Hide Artifacts: Hidden Window - T1574.002: Hijack Execution Flow: DLL Side-Loading - T1112: Modify Registry - T1027: Obfuscated Files or Information	- S0154: Cobalt Strike - S0363: Empire

APT19	APT19 is a Chinese group targetting a plethora of organisations related to energy, telecommunications, healthcare and finance.	- T1588.002: Obtain Capabilities: Tool - T1566.001: Phishing: Spearphishing Attachment - T1218.010: Signed Binary Proxy Execution: Regsvr32 - T1218.011: Signed Binary Proxy Execution: Rundll32 - T1082: System Information Discovery - T1016: System Network Configuration Discovery - T1033: System Owner/User Discovery - T1204.002: User Execution: Malicious File	
menuPass	menuPass is a threat group discovered in 2006, targeting a wide range of sectors, such as energy, finance, aerospace, maritime and healthcare.	- T1087.002: Account Discovery: Domain Account - T1583.001: Acquire Infrastructure: Domains - T1560: Archive Collected Data - T1560.001: Archive via Utility - T1119: Automated Collection - T1059.001: Command and Scripting Interpreter: PowerShell - T1059.003: Command and Scripting Interpreter: Windows Command Shell - T1005: Data from Local System - T1039: Data from Network Shared Drive - T1074.001: Data Staged: Local Data Staging - T1074.002: Data Staged: Remote Data Staging - T1140: Deobfuscate/Decode Files or Information - T1568.001: Dynamic Resolution: Fast Flux DNS - T1190: Exploit Public-Facing Application - T1210: Exploitation of Remote Services - T1083: File and Directory Discovery - T1574.001: Hijack Execution Flow: DLL Search Order Hijacking - T1574.002: Hijack Execution Flow: DLL Side-Loading - T1070.003: Indicator Removal on Host: Clear Command History - T1070.004: Indicator Removal on Host: File Deletion - T1105: Ingress Tool Transfer - T1056.001: Input Capture: Keylogging - T1036: Masquerading - T1036.003: Rename System Utilities	- S0552: AdFind - S0160: certutil - S0144: ChChes - S0106: cmd - S0154: Cobalt Strike - S0624: Ecipekac - S0404: esentutl - S0152: EvilGrab - S0628: FYAnti - S0357: Impacket - S0002: Mimikatz - S0039: Net - S0626: P8RAT - S0097: Ping - S0013: PlugX - S0012: PoisonIvy - S0194: PowerSploit - S0029: PsExec - S0006: pwdump - S0262: QuasarRAT

menuPass	menuPass is a threat group discovered in 2006, targeting a wide range of sectors, such as energy, finance, aerospace, maritime and healthcare.	- T1036.005: Match Legitimate Name or Location - T1106: Native API - T1046: Network Service Scanning - T1027: Obfuscated Files or Information - T1588.002: Obtain Capabilities: Tool - T1003.002: OS Credential Dumping: Security Account Manager - T1003.003: OS Credential Dumping: NTDS - T1003.004: OS Credential Dumping: LSA Secrets - T1566.001: Phishing: Spearphishing Attachment - T1055.012: Process Injection: Process Hollowing - T1090.002: Proxy: External Proxy - T1021.001: Remote Services: Remote Desktop Protocol - T1021.004: Remote Services: SSH - T1018: Remote System Discovery - T1053.005: Scheduled Task/Job: Scheduled Task - T1218.004: Signed Binary Proxy Execution: InstallUtil - T1553.002: Subvert Trust Controls: Code Signing - T1016: System Network Configuration Discovery - T1049: System Network Connections Discovery - T1199: Trusted Relationship - T1204.002: User Execution: Malicious File - T1078: Valid Accounts - T1047: Windows Management Instrumentation	- S0153: RedLeaves - S0159: SNUGRIDE - S0627: SodaMaster - S0275: UPPERCUT
Threat Group-3390	Threat Group-3390 is a Chinese APT group targeting in a similar manner various sectors, such as energy, aerospace and defence.	- T1548.002: Abuse Elevation Control Mechanism: Bypass User Account Control - T1087.001: Account Discovery: Local Account - T1071.001: Application Layer Protocol: Web Protocols - T1560.002: Archive Collected Data: Archive via Library - T1119: Automated Collection - T1547.001: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder - T1059.001: Command and Scripting Interpreter: PowerShell - T1059.003: Command and Scripting Interpreter: Windows Command Shell - T1543.003: Create or Modify System Process: Windows Service	- S0073: ASPXSpy - S0020: China Chopper - S0032: gh0st RAT - S0008: gsecdump - S0070: HTTPBrowser - S0398: HyperBro - S0357: Impacket - S0100: ipconfig - S0002: Mimikatz

Threat Group-3390	Threat Group-3390 is a Chinese APT group targeting in a similar manner various sectors, such as energy, aerospace and defence.	- T1543.003: Create or Modify System Process: Windows Service - T1005: Data from Local System - T1074.001: Data Staged: Local Data Staging - T1074.002: Data Staged: Remote Data Staging - T1030: Data Transfer Size Limits - T1140: Deobfuscate/Decode Files or Information - T1189: Drive-by Compromise - T1203: Exploitation for Client Execution - T1068: Exploitation for Privilege Escalation - T1210: Exploitation of Remote Services - T1133: External Remote Services - T1574.001: Hijack Execution Flow: DLL Search Order Hijacking - T1574.002: Hijack Execution Flow: DLL Side-Loading - T1562.002: Impair Defenses: Disable Windows Event Logging - T1070.004: Indicator Removal on Host: File Deletion - T1070.005: Indicator Removal on Host: Network Share Connection Removal - T1105: Ingress Tool Transfer - T1056.001: Input Capture: Keylogging - T1112: Modify Registry - T1046: Network Service Scanning - T1027: Obfuscated Files or Information - T1588.002: Obtain Capabilities: Tool - T1003.001: OS Credential Dumping: LSASS Memory - T1003.002: OS Credential Dumping: Security Account Manager - T1003.004: OS Credential Dumping: LSA Secrets - T1055.012: Process Injection: Process Hollowing - T1012: Query Registry - T1021.006: Remote Services: Windows Remote Management - T1018: Remote System Discovery - T1053.002: Scheduled Task/Job: At (Windows) - T1505.003: Server Software Component: Web Shell - T1608.002: Stage Capabilities: Upload Tool - T1608.004: Stage Capabilities: Drive-by Target	- S0590: NBTscan - S0039: Net - S0072: OwaAuth - S0013: PlugX - S0006: pwdump - S0005: Windows Credential Editor - S0412: ZxShell

Threat Group-3390	Threat Group-3390 is a Chinese APT group targeting in a similar manner various sectors, such as energy, aerospace and defence.	- T1608.004: Stage Capabilities: Drive-by Target - T1016: System Network Configuration Discovery - T1049: System Network Connections Discovery - T1078: Valid Accounts - T1047: Windows Management Instrumentation	

Appendix C

Well-known Malware against ICS

The following table summarises well-known malware and attacks against Industrial Control Systems (ICS) based on MITRE ATT&CK. For each malware, the respective techniques are also given.

TABLE C.1: Well-known Malware against Energy-related Organisations

Malware	Description	Techniques
Stuxnet	Stuxnet was the first ICS-related malware targeting the nuclear programme of Iran, by taking full advantage of multiple zero-day vulnerabilities.	- T1134.001: Access Token Manipulation: Token Impersonation/Theft - T1087.001: Account Discovery: Local Account - T1087.002: Account Discovery: Domain Account - T1071.001: Application Layer Protocol: Web Protocols - T1560.003: Archive Collected Data: Archive via Custom Method - T1547.009: Boot or Logon Autostart Execution: Shortcut Modification - T1543.003: Create or Modify System Process: Windows Service - T1132.001: Data Encoding: Standard Encoding - T1140: Deobfuscate/Decode Files or Information - T1573.001: Encrypted Channel: Symmetric Cryptography - T1480: Execution Guardrails - T1041: Exfiltration Over C2 Channel - T1068: Exploitation for Privilege Escalation - T1210: Exploitation of Remote Services - T1008: Fallback Channels - T1083: File and Directory Discovery - T1562: Impair Defenses - T1070: Indicator Removal on Host - T1070.004: File Deletion - T1070.004: Timestomp - T1570: Lateral Tool Transfer - T1112: Modify Registry - T1106: Native API - T1135: Network Share Discovery - T1027: Obfuscated Files or Information

Stuxnet	Stuxnet was the first ICS-related malware targeting the nuclear programme of Iran, by taking full advantage of multiple zero-day vulnerabilities.	- T1120 Peripheral Device Discovery - T1055.001: Process Injection: Dynamic-link Library Injection - T1090.001: Proxy: Internal Proxy - T1012: Query Registry - T1021: Remote Services - T1021.002: SMB/Windows Admin Shares - T1091: Replication Through Removable Media - T1014: Rootkit - T1053.005: Scheduled Task/Job: Scheduled Task - T1505.001: Server Software Component: SQL Stored Procedures - T1129: Shared Modules - T1518.001: Software Discovery: Security Software Discovery - T1553.002: Subvert Trust Controls: Code Signing - T1082: System Information Discovery - T1016: System Network Configuration Discovery - T1124: System Time Discovery - T1080: Taint Shared Content - T1078.001: Valid Accounts: Default Accounts - T1078.002: Valid Accounts: Domain Accounts - T1047: Windows Management Instrumentation
Duqu	Duqu is a similar to Stuxnet. It adopts a modular approach to expand its functionality.	- T1134: Access Token Manipulation - T1087.001: Account Discovery: Local Account - T1071: Application Layer Protocol - T1010: Application Window Discovery - T1560.003: Archive Collected Data: Archive via Custom Method - T1543.003: Create or Modify System Process: Windows Service - T1001.002: Data Obfuscation: Steganography - T1074.001: Data Staged: Local Data Staging - T1573.001: Encrypted Channel: Symmetric Cryptography - T1056.001: Input Capture: Keylogging - T1057: Process Discovery - T1055.001: Process Injection: Dynamic-link Library Injection - T1055.002: Process Hollowing - T1572: Protocol Tunneling - T1090.001: Proxy: Internal Proxy - T1021.002: Remote Services: SMB/Windows Admin Shares - T1053.005: Scheduled Task/Job: Scheduled Task - T1218.007: Signed Binary Proxy Execution: Msiexec - T1016: System Network Configuration Discovery - T1049: System Network Connections Discovery - T1078: Valid Accounts

Flame	An espionage malware used to collect information, targeting mainly Middle East countries	- T1123: Audio Capture - T1547.002: Boot or Logon Autostart Execution: Authentication Package - T1136.001: Create Account: Local Account - T1011.001: Exfiltration Over Other Network Medium: Exfiltration Over Bluetooth - T1210: Exploitation of Remote Services - T1091: Replication Through Removable Media - T1113: Screen Capture - T1218.011: Signed Binary Proxy Execution: Rundll32
BlackEnergy	BlackEnergy is a malware toolkit designed to form botnets executing DDoS attacks. It was used to target Ukrainian organisations.	- T1548.002:Abuse Elevation Control Mechanism: Bypass User Account Control - T1071.001: Application Layer Protocol: Web Protocols - T1547.001: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder - T1547.009: Boot or Logon Autostart Execution: Shortcut Modification - T1543.003: Create or Modify System Process: Windows Service - T1555.003: Credentials from Password Stores: Credentials from Web Browsers - T1485: Data Destruction - T1008: Fallback Channels - T1083: File and Directory Discovery - T1574.010: Hijack Execution Flow: Services File Permissions Weakness - T1070: Indicator Removal on Host - T1070.001: Clear Windows Event Logs - T1056.001: Input Capture: Keylogging - T1046: Network Service Scanning - T1120: Peripheral Device Discovery - T1057: Process Discovery - T1055.001: Process Injection: Dynamic-link Library Injection - T1021.002: Remote Services: SMB/Windows Admin Shares - T1113: Screen Capture - T1553.006: Subvert Trust Controls: Code Signing Policy Modification - T1082: System Information Discovery - T1016: System Network Configuration Discovery - T1049: System Network Connections Discovery - T1552.001: Unsecured Credentials: Credentials In Files - T1047: Windows Management Instrumentation
Industroyer	Industroyer or otherwise Crashoverride is a sophisticated malware targeting ICS and especially electrical substations. Industroyer was used to attack the Ukrainian power grid in December 2016, causing a power outage for more than 225000 households	- T1071.001: Application Layer Protocol: Web Protocols - T1554: Compromise Client Software Binary - T1543.003: Create or Modify System Process: Windows Service - T1485: Data Destruction - T1140: Deobfuscate/Decode Files or Information - T1499.004: Endpoint Denial of Service: Application or System Exploitation - T1041: Exfiltration Over C2 Channel - T1083: File and Directory Discovery - T1105: Ingress Tool Transfer

Industroyer	Industroyer or otherwise Crashoverride is a sophisticated malware targeting ICS and especially electrical substations. Industroyer was used to attack the Ukrainian power grid in December 2016, causing a power outage for more than 225000 households	- T1046: Network Service Scanning - T1027: Obfuscated Files or Information - T1572: Protocol Tunneling - T1090.003: Proxy: Multi-hop Proxy - T1012: Query Registry - T1018: Remote System Discovery - T1489: Service Stop - T1082: System Information Discovery - T1016: System Network Configuration Discovery - T1078: Valid Accounts

Appendix D

Summary of IDPS for the Smart Electrical Grid

The following table summarises the IDPS systems discussed in Chapter 4.

TABLE D.1: Summary of IDPS for the smart electrical grid

Literature	Target System	Detection Technique	Protocols	Attacks	Performance	Dataset	Software
A. Patel et al. [129]	Etire SG ecosystem	Anomaly-based	Not provided	1. Dos Attacks 2. Packet splitting 3. Command insertion 4. Shellcode mutation 5. Brute force attacks 6. Payload mutation 7. Duplicate Insertion	AUC = 0.99451	1. KDD CUP 1999 2. Simulated data	Protege
Y. Zhang et al. [222]	Entire SG ecosystem	Anomaly-based	Not provided	1. DoS attacks 2. U2R attacks 3. R2L attacks 4. Probing attacks	1. CLONALG ACC = [80.1%, 99.7%] 2. AIRS2Parallel ACC = [82.1%, 98.7%]	NSL-KDD	1. Matlab 2. WEKA

M.A. Faisal et al. [44]	AMI	Anomaly-based	Not provided	1. DoS attacks 2. R2L attacks 3. U2R attacks 4. Probing attacks	1. ACC, FPR, FNR, Size, Running time, RAM-Hours of Active Classifier = 94.67%, 3.31%, 9.13%, 134.55 KB, 3.46 secs., 1.23E-7. 2. ACC, FPR, FNR, Size, Running time, RAM-Hours of Leveraging Bagging = 98.33%, 0.78%, 5.15%, 401.01 KB, 20.92 secs., 2.22E-6. 3. ACC, FPR, FNR, Size Running time, RAM-Hours of Single Classifier Drift = 97.74%, 1.07%, 6.79%, 187.30KB, 6.74 secs., 3.34E-7.	1. KDD CUP 1999 2. NSL-KDD	MOA
R. Vijayanand [199]	AMI	Anomaly-based	Not provided	1. Exploits 2. DoS attacks 3. Fuzzers 4. Backdoor attacks 5. Worms 6. Generic attacks	1. ACC $>$ 90% 2. TPR $=$ 89.2% 3. TNR $=$ 93.4%	ADFA-LD	Matlab
Y. Li et al [109]	AMI	Anomaly-based	Not provided	Not provided	1. ACC $= 97.239\%$ 2. FPR $= 5.897\%$ 3. FNR $= 3.614\%$	CER Smart Metering Project	Not provided
P.Y. Chen [34]	AMI	Anomaly-based	Not provided	False data injection attacks	1. FPR of the first attack $= 0\%$ 2. FPR of the second attack $= 0.43\%$	Not required	Not provided
N. Boumkheld et al. [26]	AMI	Anomaly-based	AODV	Blackhole attacks	1. TPR $= 100\%$ 2. ACC $= 99\%$ 3. Precision $= 66\%$ 4. AUC $= 1$	Simulated data	1. NS2 2. WEKA

I. Ullah and H. Mahmoud [196]	AMI	Anomaly-based	Not provided	1. DoS attacks 2. L2L attacks 3. Secure shell attacks 4. Botnet	1. Precision = 99.70% 2. TPR = 99.60%	ISCX2012	WEKA
F.A.A. Alseiari and Z. Aung [12]	AMI	Anomaly-based	Not provided	1. DoS attacks 2. Port scanning	Figures present the values of TPR and FPR.	Simulated data	Not provided
V. Gulisano et al. [53]	AMI	Anomaly-based	Not provided	Energy exfiltration attacks	TPR = 91%	Not provided	Not provided
R. Berthier and W.H. Sanders [24]	AMI	Specification-based	ANSI C12.22	1. Meter reading attacks 2. Service switch attacks	1. TPR = 100% 2. TNR = 99.57% 3. CPU Consumption = 0.3% 4. RAM Consumption = 10MB	Not required	1. Table TstBench 2. VirtualBox 3. Python
X. Liu et al. [112]	AMI	Specification-based	Not provided	False data injection attacks	Figures present the values of TPR	Not required	Not provided
R. Mitchell and R. Chen [119]	AMI	Specification-based	Not provided	1. Reckless attacks 2. Random attacks	1. TPR = 100% 2. FPR of reckless attacks $\leq$ 0.2% 3. FPR of random attacks $\leq$ 0.6% 4. ROC curves are presented	Not required	Not provided
P.Jokar and V.Leung [77]	AMI	Specification-based	1. ZigBee	1. Spoofing attacks 2. Radio Jamming 3. Replay attacks 4. Stenography attacks 5. Back-off manipulation 6. DoS against CFP 7. DoS against GTS	1. Theoretical analysis 2. ROC curves are presented	Not required	Matlab
M. Attia et al. [18]	AMI	Specification-based	Not provided	1. Blackhole attacks 2. Time delay attacks	1. TPR = 90% 2. FPR = 6%	Not required	Matlab
T.H. Morris et al. [120]	SCADA	Signature-based	Modbus	Not provided	Not provided	Not required	Snort
H. Li et al. [106]	SCADA	Signature-based	DNP3	1. Protocol anomalies 2. Reconnaissance 3. DoS 4. Mixed attacks	Not provided	Not required	Snort

E. Hodo et al. [62]	SCADA	Anomaly-based	IEC-104	1. ARP attacks 2. DoS attacks 3. Replay attacks	1. TPR, FPR, Precision, AUC of Naive Bayes = 0.846, 0.055, 0.907, 0.905 2. TPR, FPR, Precision, AUC of IBk = 0.847, 0.300, 0.850, 0.766 3. TPR, FPR, Precision, AUC of J48 = 0.917, 0.090, 0928, 0.929 4. TPR, FPR, Precision, AUC of RandomForest = 0.914, 0.136, 0.919, 0.965 5. TPR, FPR, Precision, AUC of RandomTree = 0.894, 0.210, 0.895, 0.843 6. TPR, FPR, Precision, AUC of DecisionTable = 0.917, 0.062, 0.933, 0.963 7. TPR, FPR, Precision, AUC of OneR = 0.846, 0.328, 0.845, 0.759	IEC-104 dataset generated by the authors	WEKA
N. Goldenberg and A. Wool [50]	SCADA	Anomaly-based	Modbus	Not Provided	1. ACC = 100% 2. Precision = 100% 3.TPR = 100% 4. TNR = 100% 5. FPR = 0% 6. FNR = 0%	Real datasets from the authors	1. Wireshark 2. Pcapy 3. Impacket

S.D. Anton et al. [14]	SCADA	Anomaly-based	Modbus	Not provided	1. ACC of SVM with DS1, DS2 and DS3 is 100%, 100% and 99.99% respectively 2. ACC of Random Forest with DS1, DS2 and DS3 is 100%, 99.99% and 99.99% 3. ACC of KNN with DS1, DS2 and DS3 is 99.7%, 99.9% and 99.9%. 4. ACC of k-means with DS1, DS2 and DS3 is 98.1%, 55.62% and 63.36%	Lemay and Fernandez [104]	Not provided
P.H. Wang et al. [201]	SCADA	Anomaly-based	Modbus	1. Reconnaissance attacks 2. DoS attacks	1. TPR of reconnaissance attacks = 90% 2. TPR of DoS attacks = 95.12%	Data from a honeypot	1. Conpot, 2. Python 2.7 3. MongoDB
Y. Yang et al. [212]	SCADA	Specification-based	IEC 60870-5-104	1. Packet injection attacks 2. Replay attacks 3. Data manipulation	1. ACC = 100% 2. Precision = 100% 3. TPR = 100% 4. TNR = 100% 5. FPR = 0% 6. FNR = 0%	Not required	ITACA

Y. Yang et al. [210]	SCADA	Hybrid	IEC 69870-5-104	1. Unauthorized read commands 2. Unauthorized reset commands 3. Unauthorized remote control and adjustment commands 4. Spontaneous packets storm 5. Unauthorized interrogation commands 6. Buffer overflows 7. Unauthorized broadcast requests 8. IEC-104 port communication	1. ACC = 100% 2. Precision = 100% 3. TPR = 100% 4. TNR = 100% 5. FPR = 0% 6. FNR = 0%	Not required	Snort
Z.Feng et al. [45]	SCADA	Hybrid	Profinet	1. Reconnaissance attacks 2. DoS attacks 3. MiTM attacks 4. Protocol anomalies	Numerical results are not provided	Not required	Snort
S.C. Li et al. [108]	SCADA	Anomaly-based	Modbus	1. Reconnaissance 2. Response injection 3. Command injection 4. DoS	1. ACC of j48 = 99.8361% 2. ACC of 1st neural network = 97.4185% 3. ACC of 2nd neural network = 97.4603% 4. ACC of 3rd neural network = 97.3876%	Simulated dataset	1. Wireshark 2. WEKA
B. Kang et al. [79]	Substation	Signature-based	IEC 61850 MMS	Active power limitation attacks	Two examples that were detected.	Not required	Suricata

Y. Kwon et al. [99]	Substation	Specification-based	IEC 61850 GOOSE IEC 61850 MMS	1. DoS attacks 2. Port scanning 3. Portable executable 4. GOOSE attacks 5. MMS attacks 6. SNMP attacks	1. FPR = 0% 2. FNR = 1.1% 3. TPR = 98.9% 4. Precision = 100%	Real data from a substation in South Korea	Wireshark
Y. Yang et al. [213]	Substation	Specification-based	IEC 61850 GOOSE IEC 61850 MMS IEC 61850 SMV	1. DoS 2. MITM attacks 3. Packet injection	Not provided	Real data from a substation in China	1. ITACA 2. Wireshark
M. Kabir-Querrec et al. [78]	Substation	Specification-based	IEC 61850 GOOSE	Not Provided	Not provided	Not required	Not Provided
H. Yoo and T. Shon [215]	Substation	Anomaly-based	IEC 61850 GOOSE IEC 61850 MMS	Not Provided	FPR = [1%, 6%]	Real data from a substation	WEKA
U. Premaratne et al. [132]	Substation	Hybrid	IEC 61850	1. DoS attacks 2. Traffic analysis 3. Password cracking	Not provided	Real data from a substation	1. Snort 2. THC Hydra 3. Seringe
J. Hong et al. [63]	Substation	Specification-based	IEC 61850 GOOSE IEC 61850 SMV	1. DoS attacks 2. Replay attacks	FPR= 1.61×10^{-4}	Not required	1. Wireshark 2. Colasoft Packet Builder 3. Nmap
Y. Yang et al. [209]	Substation	Specification-based	IEC 61850 GOOSE IEC 61850 MMS IEC 61850 SMV	1. DoS 2. MITM attacks 3. Packet injection	Not provided	Real data from a substation in China	1. ITACA 2. Wireshark
S.Pan et al. [128]	Synchrophasor	Hybrid	Not provided	1. Single line-to-ground faults 2. Replay attacks 3. Command injection attacks 4. Disable relay attacks	ACC = 90.4%	Simulated data	1. Snort 2. OpenPDC

R.Khan et al. [85]	Synchrophasor	Hybrid	IEEE C37.118	1. ARP spoofing 2. Port scanning 3. GPS spoofing 4. Packet drop attacks 5. Replay attacks 6. Command injection 7. Physical attacks	Not provided	Not required	1. NRL core 2. OpenPMU 3. C/C++
Y. Yang et al. [211]	Synchrophasor	Specification-based	IEEE C37.118	1. Reconnaissance attacks 2. MiTM attacks 3. DoS attacks	FPR= 0%	Not required	1. ITACA 2. Nmap 3. Metasploit 4. hping

Appendix E

TCP/IP Flow Statistics

The following table enumerates and describes the TCP/IP statistics used by the TCP/IP-based Intrusion Detection Models of Chapter 5.

Table E.1: TCP/IP Flow Statistics

Feature	Description
Flow ID	ID of the flow
Src IP	Source IP address
Src Port	Source TCP/UDP port
Dst IP	Destination IP address
Protocol	The protocol related to the corresponding flow
Timestamp	Flow timestamp
Flow Duration	Duration of the flow in Microsecond
Tot Fwd Pkts	Total packets in the forward direction
Tot Bwd Pkts	Total packets in the backward direction
TotLen Fwd Pkts	Total size of packets in forward direction
TotLen Bwd Pkts	Total size of packets in backward direction
Fwd Pkt Len Max	Maximum size of packet in forward direction
Fwd Pkt Len Min	Minimum size of packet in forward direction
Fwd Pkt Len Mean	Mean size of packet in forward direction
Fwd Pkt Len Std	Standard deviation size of packet in forward direction
Fwd Pkt Len Std	Standard deviation size of packet in forward direction
Bwd Pkt Len Max	Maximum size of packet in backward direction
Bwd Pkt Len Min	Minimum size of packet in backward direction
Bwd Pkt Len Mean	Mean size of packet in backward direction
Bwd Pkt Len Std	Standard deviation size of packet in backward direction
Flow Byts/s	Number of flow bytes per second
Flow Pkts/s	Number of flow packets per second
Flow IAT Mean	Mean time between two packets sent in the flow
Flow IAT Std	Mean time between two packets sent in the flow
Flow IAT Max	Maximum time between two packets sent in the flow
Flow IAT Min	Minimum time between two packets sent in the flow
Fwd IAT Tot	Total time between two packets sent in the forward direction
Fwd IAT Mean	Mean time between two packets sent in the forward direction
Fwd IAT Std	Standard deviation time between two packets sent in the forward direction
Fwd IAT Max	Maximum time between two packets sent in the forward direction

Fwd IAT Min	Minimum time between two packets sent in the forward direction
Bwd IAT Tot	Total time between two packets sent in the backward direction
Bwd IAT Mean	Mean time between two packets sent in the backward direction
Bwd IAT Std	Standard deviation time between two packets sent in the backward direction
Bwd IAT Max	Maximum time between two packets sent in the backward direction
Bwd IAT Min	Minimum time between two packets sent in the backward direction
Fwd PSH Flags	Number of times the PSH flag was set in packets travelling in the forward direction (0 for UDP)
Bwd PSH Flags	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)
Fwd URG Flags	Number of times the URG flag was set in packets travelling in the forward direction (0 for UDP)
Bwd URG Flags	Number of times the URG flag was set in packets travelling in the backward direction (0 for UDP)
Fwd Header Len	Total bytes used for headers in the forward direction
Bwd Header Len	Total bytes used for headers in the backward direction
Fwd Pkts/s	Number of forward packets per second
Bwd Pkts/s	Number of backward packets per second
Pkt Len Min	Minimum length of a packet
Pkt Len Max	Maximum length of a packet
Pkt Len Mean	Mean length of a packet
Pkt Len Mean	Standard deviation length of a packet
Pkt Len Var	Variance length of a packet
FIN Flag Cnt	Number of packets with FIN
SYN Flag Cnt	Number of packets with SYN
RST Flag Cnt	Number of packets with RST
PSH Flag Cnt	Number of packets with PUSH
ACK Flag Cnt	Number of packets with ACK
URG Flag Cnt	Number of packets with URG
CWE Flag Count	Number of packets with CWE
ECE Flag Cnt	Number of packets with ECE
Down/Up Ratio	Download and upload ratio
Pkt Size Avg	Average size of packet
Fwd Seg Size Avg	Average size observed in the forward direction
Bwd Seg Size Avg	Average size observed in the backward direction
Fwd Byts/b Avg	Average number of bytes bulk rate in the forward direction
Fwd Pkts/b Avg	Average number of packets bulk rate in the forward direction
Fwd Blk Rate Avg	Average number of bulk rates in the forward direction
Bwd Byts/b Avg	Average number of bytes bulk rate in the backward direction
Bwd Pkts/b Avg	Average number of packets bulk rate in the backward direction
Bwd Blk Rate Avg	Average number of bulk rates in the backward direction
Subflow Fwd Pkts	The average number of packets in a sub flow in the forward direction
Subflow Fwd Byts	The average number of bytes in a sub flow in the forward direction
Subflow Bwd Pkts	The average number of packets in a sub flow in the backward direction
Subflow Bwd Byts	The average number of bytes in a sub flow in the backward direction
Init Fwd Win Byts	The total number of bytes sent in initial window in the forward direction
Init Bwd Win Byts	The total number of bytes sent in initial window in the backward direction
Fwd Act Data Pkts	Count of packets with at least 1 byte of TCP data payload in the forward direction
Fwd Seg Size Min	Minimum segment size observed in the forward direction
Active Mean	Mean time a flow was active before becoming idle
Active Std	Standard deviation time a flow was active before becoming idle
Active Max	Maximum time a flow was active before becoming idle
Active Min	Minimum time a flow was active before becoming idle
Idle Mean	Mean time a flow was idle before becoming active

Idle Std	Standard deviation time a flow was idle before becoming active
Idle Max	Maximum time a flow was idle before becoming active
Idle Min	Minimum time a flow was idle before becoming active
Label	The attack name

Appendix F

DNP3 Flow Statistics

The following table enumerates and describes the DNP3 statistics used by the DNP3 intrusion detection model described in Chapter 5.

TABLE F.1: DNP3 Flow Statistics

Feature	Description
flow ID	ID of the flow
source IP	Source IP address
destination IP	Destination IP address
source port	Source TCP/UDP Port
destination port	Destination TCP/UDP port
protocol	The protocol related to the corresponding flow
date	Flow timestamp
TotalFwdPkts	The total number of the DNP3 packets in the forward direction
TotalBwdPkts	The total number of the DNP3 packets in the backyard direction
TotLenfwdDL	The total size of the DNP3 payload at the link layer in the forward direction
TotLenfwdTR	The total size of the DNP3 payload at the transport layer in the forward direction
TotLenfwdAPP	The total size of the DNP3 payload at the application layer in the forward direction
TotLenbwdDL	The total size of the DNP3 payload at the link layer in the backyard direction
TotLenbwdTR	The total size of the DNP3 payload at the transport layer in the backyard direction
TotLenbwdAPP	The total size of the DNP3 payload at the application layer in the backyard direction
DLfwdPktLenMAX	The maximum size of the DNP3 payload at the link layer in the forward direction
DLfwdPktLenMIN	The minimum size of the DNP3 payload at the link layer in the forward direction
DLfwdPktLenMEAN	The mean of the DNP3 payload at the link layer in the forward direction
DLfwdPktLenSTD	The standard deviation of the DNP3 payload at the link layer in the forward direction
TRfwdPktLenMAX	The maximum size of the DNP3 payload at the transport layer in the forward direction
TRfwdPktLenMIN	The minimum size of the DNP3 payload at the transport layer in the forward direction
TRfwdPktLenMEAN	The mean of the DNP3 payload at the transport layer in the forward direction
TRfwdPktLenSTD	The standard deviation of the DNP3 payload at the transport layer in the forward direction
APPfwdPktLenMAX	The maximum size of the DNP3 payload at the application layer in the backyard direction
APPfwdPktLenMIN	The minimum size of the DNP3 payload at the application layer in the backyard direction
APPfwdPktLenMEAN	The mean of the DNP3 payload at the application layer in the backyard direction
APPfwdPktLenSTD	The standard deviation of the DNP3 payload at the application layer in the backyard direction

DLbwdPktLenMAX	The maximum size of the DNP3 payload at the link layer in the backyard direction
DLbwdPktLenMIN	The minimum size of the DNP3 payload at the link layer in the backyard direction
DLbwdPktLenMEAN	The mean of the DNP3 payload at the link layer in the backyard direction
DLbwdPktLenSTD	The standard deviation of the DNP3 payload at the link layer in the backyard direction
TRbwdPktLenMAX	The maximum size of the DNP3 payload at the transport layer in the backyard direction
TRbwdPktLenMIN	The minimum size of the DNP3 payload at the transport layer in the backyard direction
TRbwdPktLenMEAN	The mean of the DNP3 payload at the transport layer in the backyard direction
TRbwdPktLenSTD	The standard deviation of the DNP3 payload at the transport layer in the backyard direction
APPbwdPktLenMAX	The maximum size of the DNP3 payload at the application layer in the backyard direction
APPbwdPktLenMIN	The minimum size of the DNP3 payload at the application layer in the backyard direction
APPbwdPktLenMEAN	The mean of the DNP3 payload at the application layer in the backyard direction
APPbwdPktLenSTD	The standard deviation of the DNP3 payload at the application layer in the backyard direction
DLflowBytes/sec	How many bytes of the DNP3 link-layer were transmitted per second
TRflowBytes/sec	How many bytes of the DNP3 transport layer were transmitted per second
APPflowBytes/sec	How many bytes of the DNP3 application layer were transmitted per second
FlowPkts/sec	How many DNP3 packets were transmitted per second
FlowIAT˙MEAN	The mean of the DNP3 packets interarrival time
FlowIAT˙STD	The standard deviation of the DNP3 packets interarrival time
FlowIAT˙MAX	The maximum value of the DNP3 packets interarrival time
FlowIAT˙MIN	The minimum value of the DNP3 packets interarrival time
TotalFwdIAT	The sum of the DNP3 packets interarrival time in the forward direction
fwdIAT˙MEAN	The mean of the DNP3 packets interarrival time in the forward direction
fwdIAT˙STD	The standard deviation of the DNP3 packets interarrival time in the forward direction
fwdIAT˙MAX	The maximum value of the DNP3 packets interarrival time in the forward direction
fwdIAT˙MIN	The minimum value of the DNP3 packets interarrival time in the forward direction
TotalBwdIAT	The sum of the DNP3 packets interarrival time in the backyard direction
bwdIAT˙MEAN	The mean of the DNP3 packets interarrival time in the backyard direction
bwdIAT˙STD	The standard deviation of the DNP3 packets interarrival time in the backyard direction
bwdIAT˙MAX	The maximum value of the DNP3 packets interarrival time in the backyard direction
bwdIAT˙MIN	The minimum value of the DNP3 packets interarrival time in the backyard direction
DLfwdHdrLen	The sum of the DNP3 headers at the link layer in the forward direction
TRfwdHdrLen	The sum of the DNP3 headers at the transport layer in the forward direction
APPfwdHdrLen	The sum of the DNP3 headers at the application layer in the forward direction
DLbwdHdrLen	The sum of the DNP3 headers at the link layer in the backyard direction
TRbwdHdrLen	The sum of the DNP3 headers at the transport layer in the backyard direction
APPbwdHdrLen	The sum of the DNP3 headers at the application layer in the backyard direction
fwdPkts/sec	How many DNP3 packets per second in the forward direction
bwdPkts/sec	How many DNP3 packets per second in the backyard direction
DLpktLenMEAN	The mean of the bytes at the DNP3 link layer
DLpktLenMIN	The minimum value of the bytes at the DNP3 link layer
DLpktLenMAX	The maximum value of the bytes at the DNP3 link layer
DLpktLenSTD	The standard deviation of the bytes at the DNP3 link layer

DLpktLenVAR	The variance of the bytes at the DNP3 link layer
TRpktLenMEAN	The mean of the bytes at the DNP3 transport layer
TRpktLenMIN	The minimum value of the bytes at the DNP3 transport layer
TRpktLenMAX	The maximum value of the bytes at the DNP3 transport layer
TRpktLenSTD	The standard deviation of the bytes at the DNP3 transport layer
TRpktLenVAR	The variance of the bytes at the DNP3 transport layer
APPpktLenMEAN	The mean of the bytes at the DNP3 application layer
APPpktLenMIN	The minimum value of the bytes at the DNP3 application layer
APPpktLenMAX	The maximum value of the bytes at the DNP3 application layer
APPpktLenSTD	The standard deviation of the bytes at the DNP3 application layer
APPpktLenVAR	The variance of the bytes at the DNP3 application layer
ActiveMEAN	The time-mean where the flow was active
ActiveSTD	The time standard deviation where the flow was active
ActiveMAX	The maximum value of the time where the flow is active
ActiveMIN	The maximum value of the time where the flow is active.
IdleMEAN	The time-mean where the flow was idle before becoming active
IdleSTD	The standard deviation of the time where the flow was idle before becoming active
IdleMAX	The maximum value of the time where the flow was idle before becoming active
IdleMIN	The minimum value of the time where the flow was idle before becoming active
frameSrc	The source MAC address
frameDst	The destination MAC address
TotPktsInFlow	The total number of the DNP3 packets
firstPacketDIR	Whether the flow was initiated by a DNP3 master device or DNP3 slave device
mostCommonREQ˙FUNC˙CODE	The DNP3 function code which was used mostly in the DNP3 request packets
mostCommonRESP˙FUNC˙CODE	The DNP3 function code which was used mostly in the DNP3 response packets
corruptConfigFragments	How many responses were sent by the slave, setting the corruptConfig bit in the IIN value
deviceTroubleFragments	How many responses were sent by the slave, setting the deviceTrouble bit in the IIN value
deviceRestartFragments	How many responses were sent by the slave, setting the deviceRestart bit in the IIN value
pktsFromMASTER	How many packets that transmitted by a DNP3 master device
pktsFromSLAVE	How many packets that transmitted by a DNP3 slave device
Label	Attack label

Appendix G

IEC 60870-5-104 Flow Statistics

The following table enumerates and describes the IEC 60870-5-104 statistics used by the IEC 60870-5-104 intrusion detection model described in Chapter 5.

Table G.1: IEC 60870-5-104 Flow Statistics

Feature	**Description**
flow id	ID of the flow
protocol	The relevant protocol of the flow. It equals IEC 60870-5-104
src ip	The source IP address of the flow. It is defined with the first relevant packet.
dst ip	The destination IP address of the flow.
src port	The source TCP/UDP port.
dst port	The destination TCP/UDP port.
flow idle time max	The maximum time where the flow was idle
flow idle time min	The minimum time where the flow was idle
flow idle time mean	The time mean where the flow was idle
flow idle time std	The time standard deviation where the flow was idle
flow idle time variance	The time variance where the flow was idle
flow active time max	The maximum time where the flow was active
flow active time min	The minimum time where the flow was active
flow active time mean	The time mean where the flow was active
flow active time std	The time standard deviation where the flow was active
flow active time variance	The time variance where the flow was active
flow IAT max	The maximum interarrival time
fw IAT max	The maximum interarrival time in the forward direction
bw IAT max	The maximum interarrival time in the backyard direction
flow IAT min	The minimum interarrival time
fw IAT min	The minimum interarrival time in the forward direction
bw IAT min	The minimum interarrival time in the backyard direction
fw IAT mean	The mean of the interarrival time in the forward direction
bw IAT mean	The mean of the interarrival time in the backyard direction
flow IAT std	The standard deviation of the inter arrival time
fw IAT std	The standard deviation of the inter arrival time in the forward direction

bw IAT std	The standard deviation of the inter arrival time in the backyard direction
flow IAT tot	The total number of the interarrival times
fw iAT tot	The total number of the interarrival times in the forward direction
bw IAT tot	The total number of the interarrival times in the backyard direction
flow iec104 packts/s	The number of IEC 60870-51-04 packets per second
fw iec104 packts/s	The number of IEC 60870-51-04 packets per second in the forward direction
bw iec104 packts/s	The number of IEC 60870-51-04 packets per second in the backyard direction
flow iec104 bytes/s	The sum of APDU lengths per second
fw iec104 bytes/s	The sum of APDU lengths per second in the forward direction
bw iec104 bytes/s	The sum of APDU lengths per second in the backyard direction
flow packet APDU length max	The maximum value of the APDU lengths
flow packet APDU length min	The minimum value of the APDU lengths
flow packet APDU length mean	Mean of the APDU lengths
flow packet APDU length std	The standard deviation of the APDU lengths
flow packet APDU length var	Variance of the APDU lengths
fw packet APDU length max	The maximum value of the APDU lengths in the forward direction
fw packet APDU length min	The minimum value of the APDU lengths in the forward direction
fw packet APDU length mean	Mean of the APDU lengths in the forward direction
fw packet APDU length std	The standard deviation of the APDU lengths in the forward direction
fw packet APDU length var	The variance of the APDU lengths in the forward direction
bw packet APDU length max	The maximum value of the APDU lengths in the backyard direction
bw packet APDU length min	The minimum value of the APDU lengths in the backyard direction
bw packet APDU length mean	Mean of the APDU lengths in the backyard direction
bw packet APDU length std	The standard deviation of the APDU lengths in the backyard direction
bw packet APDU length var	The variance of the APDU lengths in the backyard direction
total flow packets	Total flow packets
total fw packets	Total flow packets in the forward direction
total bw packets	Total flow packets in the backyard direction
flow packets APDU total length	The sum of all APDU lengths
fw packets APDU total length	The sum of all APDU lengths in the forward direction
bw packets APDU total length	The sum of all APDU lengths in the backyard direction
flow duration	Flow duration in seconds
flow down/up ratio	The fraction between the IEC 60870-5-104 packets in the backyard direction and the IEC 60870-5-104 packets in the forward direction
flow total IEC104˙I˙Message˙S eqIOA packets	The total number of the I-format APCI packets that have more than one information objects
fw total IEC104˙I˙Message˙S eqIOA packets	The total number of the I-format APCI packets that have more than one information objects in the forward direction
bw total IEC104˙I˙Message˙S eqIOA packets	The total number of the I-format APCI packets that have more than one information objects in the backyard direction
flow total IEC104˙I˙Message˙Si ngleIOA packets	The total number of the I-format APCI packets that have one information object in ASDU
fw total IEC104˙I˙Message˙Si ngleIOA packets	The total number of the I-format APCI packets that have one information object in ASDU in the forward direction
bw total IEC104˙I˙Message˙Si ngleIOA packets	The total number of the I-format APCI packets that have one information object in ASDU in the backyard direction
flow total IEC104˙S˙Message packets	The total number of the S-format APCI packets
fw total IEC104˙S˙Message packets	The total number of the S-format APCI packets in the forward direction

bw total IEC104˙S˙Message packets	The total number of the S-format APCI packets in the backyard direction
flow total IEC104˙U˙Message packets	The total number of the U-format APCI packets
fw total IEC104˙U˙Message packets	The total number of the U-format APCI packets in the forward direction
bw total IEC104˙U˙Message packets	The total number of the U-format APCI packets in the backyard direction
fw URG flag amount	The number of the URG flags in the forward direction
fw PSH flag amount	The number of the PSH flags in the forward direction
bw URG flag amount	The number of the URG flags in the backyard direction
bw PSH flag amount	The number of the PSH flags in the backyard direction
flow SYN flag count	The number of the TCP SYN packets
flow RST flag count	The number of the TCP RST packets
flow PSH flag count	The number of the TCP PSH packets
flow ACK flag count	The number of the TCP ACK packets
flow URG flag count	The number of the TCP URG packets
flow CWE flag count	The number of the TCP CWE packets
flow ECE flag count	The number of the TCP ECE packets
fw˙subflow˙packets	The average number of packets in a sub flow in the forward direction
bw˙subflow˙packets	The average number of packets in a sub flow in the backward direction
fw˙subflow˙bytes	The average number of bytes in a sub flow in the forward direction
bw˙subflow˙bytes	The average number of bytes in a sub flow in the backward direction
flow start timestamp	The timestamp of the flow. It is defined with the first relevant packet.
fw avg bytes/bulk	Average number of bytes bulk rate in the forward direction
bw avg bytes/bulk	Average number of bytes bulk rate in the backyard direction
fw avg bulk rate	Average number of bulk rate in the forward direction
bw avg bulk rate	Average number of bulk rate in the backyard direction
fw avg packets/bulk	Average number of packets bulk rate in the forward direction
bw avg packets/bulk	Average number of packets bulk rate in the backyard direction
init fw window bytes	The window size of the first packet in the forward direction
init bw window bytes	The window size of the first packet in the backyard direction
fw TCP total header length	The length of the TCP headers in the forward direction
bw TCP total header length	The length of the TCP headers in the backyard direction
cot=1	The total number of the IEC 60870-5-104 packets where COT = 1 (periodic,cyclic)
cot=2	The total number of the IEC 60870-5-104 packets where COT = 2 (background interrogation)
cot=3	The total number of the IEC 60870-5-104 packets where COT = 3 (spontaneous)
cot=4	The total number of the IEC 60870-5-104 packets where COT = 4 (initialized)
cot=5	The total number of the IEC 60870-5-104 packets where COT = 5 (interrogation)
cot=6	The total number of the IEC 60870-5-104 packets where COT = 6 (activation)
cot=7	The total number of the IEC 60870-5-104 packets where COT = 7 (confirmation activation)
cot=8	The total number of the IEC 60870-5-104 packets where COT = 8 (deactivation)

cot=9	The total number of the IEC 60870-5-104 packets where COT = 9 (confirmation deactivation)
cot=10	The total number of the IEC 60870-5-104 packets where COT = 10 (termination activation)
cot=11	The total number of the IEC 60870-5-104 packets where COT = 11 (feedback, caused by distant command)
cot=12	The total number of the IEC 60870-5-104 packets where COT = 12 (feedback, caused by local command)
cot=13	The total number of the IEC 60870-5-104 packets where COT = 13 (COT data transmission)
cot=20	The total number of the IEC 60870-5-104 packets where COT = 20 (interrogated by general interrogation)
type_id_process_information_in_monitor_direction	The total number of the IEC 60870-5-104 packets where TypeID is in the range 1-40
type_id_process_information_in_control_direction	The total number of the IEC 60870-5-104 packets where TypeID is in the range 45- 51
type_id_system_information_in_monitor_direction	The total number of the IEC 60870-5-104 packets where TypeID is in the range 70
type_id_system_information_in_control_direction	The total number of the IEC 60870-5-104 packets where TypeID is in the range 100- 106
type_id_parameter_in_control_direction	The total number of the IEC 60870-5-104 packets where TypeID is in the range 110- 113
type_id_file_transfer	The total number of the IEC 60870-5-104 packets where TypeID is in the range 120- 126
Label	Attack label

Appendix H

Operational Data/Features of the Hydropower Plant Scenario

The following table enumerates and describes the operational data/features of the hydropower plant scenario.

Table H.1: Operational Data/Features of the Hydropower Plant Scenario

Feature	**Description**
DE	Temperature of DE bearing of the generator
Power	Power (active energy) of the plant
Waterlevel	Water level in the upper basin
NDE	Temperature of NDE bearing of the generator
nozzles	Position of turbine guide vanes in %

Appendix I

Operational Data/Features of the Substation Scenario

The following table enumerates and describes the operational data/features of the substation scenario.

Table I.1: Operational Data/Features of the Substation Scenario

Feature	Description
FRECUENCY_SOE	Frequency (Typical value: 50 Hz)
TEMPERATURE_SOE	Temperature (Typical value: 25 C)
VOLTAGE_SOE	Voltage: (Typical value: 230 V)
CURRENT_SOE	Current: (Typical value: 100 A)
APPARENT_POWER_SOE	VOLTAGE_SOE × CURRENT_SOE
ACTIVE_POWER_SOE	Active Power
REACTIVE_POWER_SOE	Reactive Power
TRAFOS_POSITION_SOE	Trafos position

Appendix J

Operational Data/Features of the Power Plant Scenario

The following table enumerates and describes the operational data/features of the power plant scenario.

Table J.1: Operational Data/Features of the Power Plant Scenario

Feature	Description
24V Batteries	24 V Batteries voltage
60V Batteries	60 V Batteries voltage
Generator Speed	Generator motor speed
Gen Motor Voltage	Generator motor voltage
Gen Motor Current	Generator motor current
Exc Motor Voltage	Exciter motor voltage
Exc Motor Current	Exciter motor current
Incom Cooling Water	Temperature of incoming cooling water
Gen Status Winding2	Temperature of generator winding at point 2
Gen Outlet Air	Temperature of outlet air
Exc Set Bearing2	Temperature of exciter winding at point 2
Grid Phase R	Indicates that voltage exists on the L1 phase
Grid Phase S	Indicates that voltage exists on the L2 phase
Grid Phase T	Indicates that voltage exists on the L3 phase
Main MG Nn	The generator has acquired rated rounds per minutes (rpms)
Exc MG Nn	The exciter has acquired rated rpms
Overvolt Main Gen	Indicates that overvoltage on the main generator exists
Overcur Main Gen	Indicates that overcurrent on the main generator exists

Appendix K

Operational Data/Features of the Smart Home Scenario

The following table enumerates and describes the operational data/features of the smart home scenario.

Table K.1: Operational Data/Features of the Smart Home Scenario

Feature	Description
PinPhL1	Frequency (Input Apparent Power Line 1 (VA)
PinPhL2	Input Apparent Power Line 2 (VA)
PinPhL3	Input Apparent Power Line 3 (VA)
PoutPhL1	Output Apparent Power Line 1 (VA)
PoutPhL2	Output Apparent Power Line 2 (VA)
PoutPhL3	Output Apparent Power Line 2 (VA)
VoutPhL1	Voltage Line 1 (V)
VoutPhL2	Voltage Line 2 (V)
VoutPhL3	Voltage Line 3 (V)
PsetPhL1	ESS power setpoint phase 1 (W)
PsetPhL2	ESS power setpoint phase 2 (W)
PsetPhL3	ESS power setpoint phase 3 (W)
Ein3Ph	MG 3 Phase Energy Flow (kWh)
ESS_DC_Quarter_kWh	ESS DC Energy Flow (kWh)
ChargeFlag	ESS disable charge flag phase (-)
FeedbackFlag	ESS disable feedback flag phase (-)
Vdc	Battery Voltage (V)
BattVolt	Battery Voltage (MasterVolt) (V)
AoutPhL1	Amperage Line 1 (A)
AoutPhL2	Amperage Line 2 (A)
AoutPhL3	Amperage Line 3 (A)
AinLimit	Input Amperage Limit (A)
Adc	Battery Amperage (A)
BattAmp	Battery Amperage (MasterVolt) (A)
SoC	State Of Charge (%))
BattSoC	State Of Charge (MasterVolt) (%)
Fout	Frequency (Hz)

State	VE Bus State
SwitchPos	Switch Position
CapacityCons	Capacity Consumed (Mastervolt) (Ah)
BattTemp	Battery Temperature (Mastervolt)(oC)
TempAlarm	High Temperature Alarm
LowBatAlarm	Low Battery Alarm
OverLoAlarm	Overload Alarm
VEBusError	VE Bus Error

Appendix L

IEC 61850 (GOOSE) Flow Statistics

The following table enumerates and describes the GOOSE statistics used by the GOOSE Intrusion Detection Model described in Chapter 5.

Table L.1: GOOSE Flow Statistics

Feature	**Description**
flow ID	ID of the flow
source MAC	The source MAC address of the flow, according to the first packet
destination MAC	The destination MAC address of the flow, according to the first packet
date	Timestamp of the first relevant packet
duration	Duration of the flow
TotalFwdPkts	Total GOOSE packets sent in the forward direction
TotalBwdPkts	Total GOOSE packets sent in the backyard direction
TotLenfwd	The total length of the GOOSE packets sent in the forward direction
TotLenbwd	The total length of the GOOSE packets sent in the backyard direction
fwdPktLenMAX	The maximum value of the GOOSE payload size observed in the GOOSE packets in the forward direction
fwdPktLenMIN	The minimum value of the GOOSE payload size observed in the GOOSE packets in the forward direction
fwdPktLenMEAN	The mean of the GOOSE payload size observed in the GOOSE packets in the forward direction
fwdPktLenSTD	The standard deviation of the GOOSE payload size observed in the GOOSE packets in the forward direction
bwdPktLenMAX	The maximum value of the GOOSE payload size observed in the GOOSE packets in the backyard direction
bwdPktLenMIN	The minimum value of the GOOSE payload size observed in the GOOSE packets in the backyard direction
bwdPktLenMEAN	The mean of the GOOSE payload size observed in the GOOSE packets in the backyard direction
bwdPktLenSTD	The standard deviation of the GOOSE payload size observed in the GOOSE packets in the backyard direction
flowBytes/sec	GOOSE payload bytes per second transmitted in a flow
FlowPkts/sec	GOOSE payload packets per second transmitted in a flow
FlowIAT˙MEAN	Mean of the GOOSE packets interarrival time
FlowIAT˙STD	Standard deviation of the GOOSE packets interarrival time
FlowIAT˙MAX	The maximum value of the GOOSE packets interarrival time
FlowIAT˙MIN	The minimum value of the GOOSE packets interarrival time

TotalFwdIAT	Total number of the GOOSE packets interarrival time in the forward direction
fwdIAT˙MEAN	Mean of the GOOSE packets interarrival time in the forward direction
fwdIAT˙STD	Standard deviation of the GOOSE packets interarrival time in the forward direction
fwdIAT˙MAX	The maximum value of the GOOSE packets interarrival time in the forward direction
fwdIAT˙MIN	The minimum value of the GOOSE packets interarrival time in the forward direction
TotalBwdIAT	Total number of the GOOSE packets interarrival time in the backyard direction
bwdIAT˙MEAN	Mean of the GOOSE packets interarrival time in the backyard direction
bwdIAT˙STD	Standard deviation of the GOOSE packets interarrival time in the backyard direction
bwdIAT˙MAX	The maximum value of the GOOSE packets interarrival time in the backyard direction
bwdIAT˙MIN	The minimum value of the GOOSE packets interarrival time in the backyard direction
fwdHdrLen	Sum of the GOOSE header in the forward direction
bwdHdrLen	Sum of the GOOSE header in the backyard direction
fwdPkts/sec	Number of packets transmitted per second in the forward direction
bwdPkts/sec	Number of packets transmitted per second in the backyard direction
pktLenMEAN	Mean of the GOOSE payload
pktLenMIN	The minimum value of the GOOSE payload
pktLenMAX	The maximum value of the GOOSE payload
pktLenSTD	The standard deviation of the GOOSE payload
pktLenVAR	The variance of the GOOSE payload
ActiveMEAN	Time-mean where the flow was active before becoming idle
ActiveSTD	The standard deviation where the flow was active before becoming idle
ActiveMAX	The maximum time where the flow was active before becoming idle
ActiveMIN	The minimum time where the flow was active before becoming idle
IdleMEAN	Time-mean where the flow was idle
IdleSTD	The standard deviation where the flow was idle
IdleMAX	The maximum time where the flow was idle
IdleMIN	The minimum time where the flow was idle
Data_Change_Cnt	The minimum time where the flow was idle
Data_Change_IAT_mean	The mean interarrival time where the GOOSE dataset's values were changed
Data_Change_IAT_std	The interarrival time standard deviation where the GOOSE dataset's values were changed
Data_Change_IAT_max	The maximum interarrival time where the GOOSE dataset's values where changed
Data_Change_IAT_min	The minimum interarrival time where the GOOSE dataset's values where changed
DataSet_Conf_Change_Cnt	How many times the GOOSE dataset configuration was changed
DataSet_Conf_Change_IAT_mean	The mean interarrival time where the GOOSE dataset configuration was changed
DataSet_Conf_Change_IAT_std	The interarrival time standard deviation where the GOOSE dataset configuration was changed
DataSet_Conf_Change_IAT_max	The maximum interarrival time where the GOOSE dataset configuration was changed
DataSet_Conf_Change_IAT_min	The minimum interarrival time where the GOOSE dataset configuration was changed
GOOSE_msg_Cnt	How many GOOSE messages were transmitted
GOOSE_msg_IAT_mean	Mean interarrival time of the GOOSE messages

GOOSE_msg_IAT_std	The interarrival time standard deviation of the GOOSE messages
GOOSE_msg_IAT_max	The maximum interarrival time of the GOOSE messages
GOOSE_msg_IAT_min	The minimum interarrival time of the GOOSE messages
DataSet_Entries_mean	Mean of dataset entries
DataSet_Entries_max	The maximum number of the dataset entries
DataSet_Entries_min	The minimum number of the dataset entries
numGooMSGS_b4_datset_change_mean	Mean of GOOSE messages before the change of the GOOSE dataset
numGooMSGS_b4_datset_change_max	The maximum number of GOOSE messages before the change of the GOOSE dataset
numGooMSGS_b4_datset_change_min	The minimum number of GOOSE messages before the change of the GOOSE dataset
numGooMSGS_b4_datset_change_std	The standard deviation of the GOOSE messages before the change of the GOOSE dataset
invalidAPPID_count	The number of packets with the invalid APP ID
GBlock_needs_configuration_count	Count of ndsComm changes
Label	Attack label

Appendix M

NF-IDPS Evaluation Results

The following tables present the ML/DL comparative analysis related to the intrusion and anomaly detection models of NF-IDPS.

Table M.1: Comparative Evaluation Results of Modbus/TCP Intrusion Detection Model

Classification Problem	Multi-Class Classification			
Dataset	Modbus/TCP Intrusion Detection Dataset (it will be published in IEEE Dataport and Zenodo)			
Features	Appendix E			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.943	0.603	0.030	0.603
LDA	0.943	0.604	0.030	0.604
Decision Tree Classifier	0.964	0.749	0.019	0.749
Naïve Bayes	0.928	0.497	0.038	0.497
SVM RBF	0.918	0.426	0.044	0.426
SVM Linear	0.921	0.453	0.042	0.453
Random Forest	0.947	0.633	0.028	0.633
MLP	0.938	0.570	0.033	0.570
Adaboost	0.887	0.214	0.060	0.214
Quadratic Discriminant Analysis	0.941	0.593	0.031	0.593
Dense DNN Relu	0.945	0.619	0.029	0.619
Dense DNN Tanh	0.945	0.619	0.029	0.619

Table M.2: Comparative Evaluation Results of Modbus/TCP Anomaly Detection Model

Classification Problem	Outlier/Novelty Detection			
Dataset	Modbus/TCP Intrusion Detection Dataset (it will be published in IEEE Dataport and Zenodo)			
Features	Appendix E			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.949	0.999	0.100	0.951
Isolation Forest	0.950	0.999	0.099	0.952

PCA	0.540	0.846	0.567	0.488
MCD	0.948	0.999	0.102	0.950
LOF	0.947	0.999	0.104	0.950
Autoencoder	0.950	0.999	0.099	0.952

Table M.3: Comparative Evaluation Results of DNP3 Intrusion Detection Model

Classification Problem	Multi-Class Classification			
Dataset	DNP3 Intrusion Detection Dataset (it will be published in IEEE Dataport and Zenodo)			
Features	Appendix F			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.756	7567	0.030	0.750
LDA	0.702	0.702	0.037	0.687
Decision Tree Classifier	0.959	0.959	0.005	0.959
Naïve Bayes	0.683	0.683	0.039	0.649
SVM RBF	0.690	0.690	0.038	0.651
SVM Linear	0.651	0.651	0.043	0.580
Random Forest	0.708	0.708	0.036	0.692
MLP	0.706	0.706	0.036	0.665
Adaboost	0.222	0.222	0.097	0.111
Quadratic Discriminant Analysis	0.716	0.716	0.035	0.660
Dense DNN Relu	0.755	0.755	0.030	0.737
Dense DNN Tanh	0.755	0.755	0.030	0.734

Table M.4: Comparative Evaluation Results of DNP3 TCP/IP Intrusion Detection Model

Classification Problem	Multi-Class Classification			
Dataset	DNP3 Intrusion Detection Dataset (it will be published in IEEE Dataport and Zenodo)			
Features	Appendix E			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.490	0.490	0.050	0.444
LDA	0.627	0.627	0.037	0.612
Decision Tree Classifier	0.797	0.797	0.020	0.782
Naïve Bayes	0.690	0.683	0.030	0.655
SVM RBF	0.554	0.554	0.044	0.500
SVM Linear	0.593	0.593	0.040	0.523
Random Forest	0.726	0.726	0.027	0.672
MLP	0.475	0.475	0.052	0.423
Adaboost	0.272	0.272	0.072	0.168
Quadratic Discriminant Analysis	0.090	0.090	0.090	0.015
Dense DNN Relu	0.584	0.584	0.041	0.539
Dense DNN Tanh	0.552	0.552	0.044	0.505

Table M.5: Evaluation Results of DNP3 TCP/IP Anomaly Detection Model

Classification Problem	Outlier/Novelty Detection			
Dataset	DNP3 Intrusion Detection Dataset (it will be published in IEEE Dataport and Zenodo)			
Features	Appendix E			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.951	0.999	0.097	0.953
Isolation Forest	0.950	0.999	0.098	0.953
PCA	0.500	0.000	0.000	0.000
LOF	0.942	0.999	0.114	0.945
MCD	0.946	0.999	0.107	0.949
Autoencoder	0.948	0.999	0.104	0.950

Table M.6: Comparative Evaluation Results of IEC 60870-5-104 Intrusion Detection Model

Classification Problem	Multi-Class Classification			
Dataset	IEC 60870-5-104 Intrusion Detection Dataset (Available in IEEE Dataport and Zenodo)			
Features	Appendix G			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.622	0.622	0.034	0.605
LDA	0.618	0.618	0.034	0.605
Decision Tree Classifier	0.831	0.831	0.015	0.825
Naïve Bayes	0.558	0.558	0.040	0.474
SVM RBF	0.553	0.553	0.040	0.480
SVM Linear	0.508	0.508	0.044	0.4144
Random Forest	0.664	0.664	0.030	0.647
MLP	0.590	0.590	0.037	0.570
Adaboost	0.250	0.250	0.068	0.181
Quadratic Discriminant Analysis	0.608	0.608	0.035	0.534
Dense DNN Relu	0.642	0.642	0.032	0.598
Dense DNN Tanh	0.576	0.576	0.038	0.517

Table M.7: Comparative Evaluation Results of IEC 60870-5-104 TCP/IP Intrusion Detection Model

Classification Problem	Multi-Class Classification			
Dataset	IEC 60870-5-104 Intrusion Detection Dataset (Available in IEEE Dataport and Zenodo)			
Features	Appendix E			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.900	0.602	0.056	0.602
LDA	0.904	0.619	0.054	0.619
Decision Tree Classifier	0.953	0.815	0.026	0.815
Naïve Bayes	0.855	0.421	0.082	0.421

SVM RBF	0.853	0.413	0.083	0.413
SVM Linear	0.843	0.375	0.089	0.375
Random Forest	0.918	0.672	0.046	0.672
MLP	0.904	0.619	0.054	0.619
Adaboost	0.843	0.375	0.089	0.375
Quadratic Discriminant Analysis	0.899	0.598	0.057	0.598
Dense DNN Relu	0.909	0.636	0.051	0.636
Dense DNN Tanh	0.916	0.664	0.047	0.664

Table M.8: Comparative Evaluation Results of IEC 60870-5-104 TCP/IP Anomaly Detection Model

Classification Problem	Outlier/Novelty Detection			
Dataset	IEC 60870-5-104 Intrusion Detection Dataset (Available in IEEE Dataport and Zenodo)			
Features	Appendix E			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.947	0.999	0.105	0.949
Isolation Forest	0.950	0.999	0.094	0.955
PCA	0.500	0.000	0.000	0.000
LOF	0.949	0.999	0.101	0.951
MCD	0.880	0.857	0.097	0.877
Autoencoder	0.881	0.852	0.089	0.877

Table M.9: Comparative Evaluation Results of GOOSE Intrusion Detection Model

Classification Problem	Multi-Class Classification			
Dataset	Synthesized Dataset: P.P. Biswas et al. in [25]			
Features	Appendix L			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.351	0.351	0.162	0.272
LDA	0.433	0.433	0.141	0.368
Decision Tree Classifier	0.406	0.406	0.148	0.389
Naïve Bayes	0.228	0.228	0.192	0.108
SVM RBF	0.371	0.371	0.157	0.279
SVM Linear	0.843	0.375	0.089	0.375
Random Forest	0.359	0.359	0.160	0.275
MLP	0.355	0.355	0.161	0.279
Adaboost	0.387	0.387	0.153	0.286
Quadratic Discriminant Analysis	0.203	0.203	0.199	0.187
Dense DNN Relu	0.375	0.375	0.156	0.301
Dense DNN Tanh	0.368	0.368	0.157	0.311

Table M.10: Comparative Evaluation Results of MMS TCP/IP Anomaly Detection Model

Classification Problem	Outlier/Novelty Detection

Dataset	Synthetic data from the SPEAR project			
Features	Appendix E			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.968	0.999	0.064	0.968
Isolation Forest	0.971	0.999	0.058	0.971
PCA	0.500	0.000	0.000	0.000
LOF	0.955	0.999	0.090	0.956
MCD	0.977	0.999	0.045	0.977
Autoencoder	0.972	0.999	0.056	0.9727

TABLE M.11: Comparative Evaluation Results of HTTP TCP/IP Intrusion Detection Model

Classification Problem	Multi-Class Classification			
Dataset	CSE-CIC-IDS2018			
Features	Appendix E			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.937	0.844	0.038	0.844
LDA	0.946	0.866	0.033	0.866
Decision Tree Classifier	0.964	0.911	0.026	0.911
Naïve Bayes	0.878	0.696	0.075	0.696
SVM RBF	0.908	0.770	0.057	0.770
SVM Linear	0.928	0.822	0.044	0.822
Random Forest	0.922	0.807	0.048	0.807
MLP	0.940	0.851	0.037	0.851
Adaboost	0.760	0.400	0.150	0.400
Quadratic Discriminant Analysis	0.911	0.777	0.055	0.777
Dense DNN Relu	0.940	0.851	0.037	0.851
Dense DNN Tanh	0.940	0.851	0.0370	0.851

TABLE M.12: Comparative Evaluation Results of HTTP TCP/IP Anomaly Detection Model

Classification Problem	Outlier/Anomaly Detection			
Dataset	CSE-CIC-IDS2018			
Features	Appendix E			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.577	0.571	0.416	0.558
Isolation Forest	0.833	0.948	0.281	0.850
PCA	0.596	0.592	0.400	0.581
MCD	0.719	0.545	0.106	0.660
LOF	0.946	0.954	0.058	0.938
DIDEROT Autoencoder	0.934	0.927	0.061	0.902

Table M.13: Comparative Evaluation Results of SSH TCP/IP Intrusion Detection Model

Classification Problem	Multi-Class Classification			
Dataset	CSE-CIC-IDS2018			
Features	Appendix E			
Training Dataset Size	70%			
Testing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.859	0.750	0.058	0.821
LDA	0.945	0.920	0.038	0.928
Decision Tree Classifier	0.960	0.958	0.038	0.955
Naïve Bayes	0.823	0.741	0.154	0.640
SVM RBF	0.837	0.660	0.339	0.788
SVM Linear	0.799	0.845	0.307	0.307
Random Forest	0.955	0.903	0.009	0.942
MLP	0.903	0.841	0.010	0.910
Adaboost	0.950	0.890	0.010	0.934
Quadratic Discriminant Analysis	0.500	0.500	0.250	0.666
Dense DNN Relu	0.916	0.985	0.014	0.906
Dense DNN Tanh	0.916	0.836	0.011	0.904

Table M.14: Comparative Evaluation Results of SSH TCP/IP Anomaly Detection Model

Classification Problem	Outlier/Anoamly Detection			
Dataset	CSE-CIC-IDS2018			
Features	Appendix E			
Training Dataset Size	70%			
Testing Dataset Size	30%			
Classification Problem	Outlier/Novelty Detection			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.935	0.870	0.013	0.922
Isolation Forest	0.943	0.901	0.013	0.941
PCA	0.701	0.596	0.247	0.564
MCD	0.957	0.970	0.050	0.944
LOF	0.925	0.913	0.066	0.909
DIDEROT Autoencoder	0.946	0.954	0.058	0.938

Appendix N

H-IDPS Evaluation Results

The following tables present the ML/DL comparative analysis related to the anomaly detection models of H-IDPS.

TABLE N.1: Operational Data Based Anomaly Detection Model – Hydropower Plant Use Case.

Classification Problem	Outlier/Novelty Detection			
Data Type	Oprational Data - Hydropower Plant Use Case			
Features	Appendix H			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.581	0.993	0.522	0.487
Isolation Forest	0.716	0.948	0.341	0.572
PCA	0.745	0.978	0.312	0.606
MCD	0.733	0.210	0.135	0.240
LOF	0.579	0.996	0.525	0.486
ARIES GAN	0.746	0.978	0.311	0.607

TABLE N.2: Operational Data Based Anomaly Detection Model – Substation Use Case.

Classification Problem	Outlier/Novelty Detection			
Data Type	Operational Data - Substation Use Case			
Features	Appendix I			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.839	0.995	0.200	0.713
Isolation Forest	0.850	0.951	0.175	0.718
PCA	0.847	0.961	0.181	0.716
MCD	0.822	0.991	0.220	0.691
LOF	0.873	0.993	0.157	0.759
ARIES GAN	0.840	0.961	0.189	0.708

Table N.3: Operational Data Based Anomaly Detection Model – Power Plant Use Case.

Classification Problem	Outlier/Novelty Detection			
Data Type	Operational Data - Power Plant Use Case			
Features	Appendix J			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.692	0.989	0.397	0.600
Isolation Forest	0.813	0.960	0.231	0.705
PCA	0.851	0.982	0.187	0.755
MCD	0.715	0.299	0.158	0.329
LOF	0.829	0.992	0.220	0.730
ARIES GAN	0.851	0.982	0.188	0.755

Table N.4: Operational Data Based Anomaly Detection Model – Smart Home Use Case.

Classification Problem	Outlier/Novelty Detection			
Data Type	Operational Data - Smart Home Use Case			
Features	Appendix K			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.649	0.668	0.362	0.597
Isolation Forest	0.769	0.976	0.279	0.615
PCA	0.859	0.976	0.167	0.724
MCD	0.729	0.992	0.332	0.581
LOF	0.690	0.735	0.344	0.676
ARIES GAN	0.859	0.976	0.167	0.725

Bibliography

[1] Divesh Aggarwal, Gavin K Brennen, Troy Lee, Miklos Santha, and Marco Tomamichel. 2018. Quantum attacks on Bitcoin, and how to protect against them. *Ledger* 3 (2018), 1–21.

[2] Preeti Aggarwal and Sudhir Kumar Sharma. 2015. Analysis of KDD dataset attributes-class wise for intrusion detection. *Procedia Computer Science* 57 (2015), 842–851.

[3] Irfan Ahmed, Sebastian Obermeier, Sneha Sudhakaran, and Vassil Roussev. 2017. Programmable logic controller forensics. *IEEE Security & Privacy* 15, 6 (2017), 18–24.

[4] Kevser Ovaz Akpinar and Ibrahim Ozcelik. 2019. Analysis of machine learning methods in EtherCAT-based anomaly detection. *IEEE Access* 7 (2019), 184365–184374.

[5] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. 2015. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials* 17, 4 (2015), 2347–2376.

[6] Noor Al-Gharabally, Nosayba El-Sayed, Sara Al-Mulla, and Imtiaz Ahmad. 2009. Wireless honeypots: survey and assessment. In *Proceedings of the 2009 conference on Information Science, Technology and Applications*. ACM, Kuwait, Kuwait, 45–52.

[7] Rawan Al-Shaer, Jonathan M Spring, and Eliana Christou. 2020. Learning the associations of mitre att & ck adversarial techniques. In *2020 IEEE Conference on Communications and Network Security (CNS)*. IEEE, Avignon, France, 1–9.

[8] Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alotaibi. 2017. Internet of Things security: A survey. *Journal of Network and Computer Applications* 88 (2017), 10–28.

[9] Omar Alghushairy, Raed Alsini, Terence Soule, and Xiaogang Ma. 2020. A review of local outlier factor algorithms for outlier detection in big data streams. *Big Data and Cognitive Computing* 5, 1 (2020), 1.

[10] Mohammed H Almeshekah and Eugene H Spafford. 2016. Cyber security deception. , 23–50 pages.

[11] Fahad M Alotaibi and Vassilios G Vassilakis. 2021. SDN-Based Detection of Self-Propagating Ransomware: The Case of BadRabbit. *IEEE Access* 9 (2021), 28039–28058.

[12] Fadwa Abdul Aziz Alseiari and Zeyar Aung. 2015. Real-time anomaly-based distributed intrusion detection systems for advanced Metering Infrastructure utilizing stream data mining. In *2015 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE)*. IEEE, Offenburg, Germany, 148–153.

[13] Adel Alshamrani, Sowmya Myneni, Ankur Chowdhary, and Dijiang Huang. 2019. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials* 21, 2 (2019), 1851–1877.

[14] Simon Duque Anton, Suneetha Kanoor, Daniel Fraunholz, and Hans Dieter Schotten. 2018. Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/tcp data set. In *Proceedings of the 13th international conference on availability, reliability and security*. ACM, Hamburg, Germany, 1–9.

[15] Haripriya AP et al. 2019. Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things. *EURASIP Journal on Wireless Communications and Networking* 2019, 1 (2019), 1–15.

[16] Emekcan Aras, Gowri Sankar Ramachandran, Piers Lawrence, and Danny Hughes. 2017. Exploring the security vulnerabilities of LoRa. In *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*. IEEE, Exeter, UK, 1–6.

[17] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38.

[18] Mohamed Attia, Hichem Sedjelmaci, Sidi Mohammed Senouci, and El-Hassane Aglzim. 2015. A new intrusion detection approach against lethal attacks in the smart grid: temporal and spatial based detections. In *2015 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, Guadalajara, Mexico, 1–3.

[19] Dinesh Chowdary Attota, Viraaji Mothukuri, Reza M Parizi, and Seyedamin Pouriyeh. 2021. An ensemble multi-view federated learning intrusion detection for iot. *IEEE Access* 9 (2021), 117734–117745.

[20] Sana Aurangzeb, Muhammad Aleem, Muhammad Azhar Iqbal, Muhammad Arshad Islam, et al. 2017. Ransomware: a survey and trends. *Journal of Information Assurance & Security* 6, 2 (2017), 48–58.

[21] Suresh Balakrishnama and Aravind Ganapathiraju. 1998. Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing* 18, 1998 (1998), 1–8.

[22] Horace B Barlow. 1989. Unsupervised learning. *Neural computation* 1, 3 (1989), 295–311.

[23] Boldizsár Bencsáth, Gábor Pék, Levente Buttyán, and Mark Felegyhazi. 2012. The cousins of stuxnet: Duqu, flame, and gauss. *Future Internet* 4, 4 (2012), 971–1003.

[24] Robin Berthier and William H Sanders. 2011. Specification-based intrusion detection for advanced metering infrastructures. In *2011 IEEE 17th Pacific rim international symposium on dependable computing*. IEEE, Pasadena, CA, USA, 184–193.

[25] Partha P Biswas, Heng Chuan Tan, Qingbo Zhu, Yuan Li, Daisuke Mashima, and Binbin Chen. 2019. A synthesized dataset for cybersecurity study of IEC 61850 based substation. In *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, Beijing, China, 1–7.

[26] Nadia Boumkheld, Mounir Ghogho, and Mohammed El Koutbi. 2016. Intrusion detection system for the detection of blackhole attacks in a smart grid. In *2016 4th International Symposium on Computational and Business Intelligence (ISCBI)*. IEEE, Olten, Switzerland, 108–111.

[27] Enrique Mármol Campos, Pablo Fernández Saura, Aurora González-Vidal, José L Hernández-Ramos, Jorge Bernal Bernabe, Gianmarco Baldini, and Antonio Skarmeta. 2021. Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges. *Computer Networks* 203, - (2021), 108661.

[28] Nicola Capuano, Giuseppe Fenza, Vincenzo Loia, and Claudio Stanzione. 2022. Explainable Artificial Intelligence in CyberSecurity: A Survey. *IEEE Access* 10 (2022), 93575–93600.

[29] Daniel Celebucki, Maj Alan Lin, and Scott Graham. 2018. A security evaluation of popular internet of things protocols for manufacturers. In *2018 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, Las Vegas, NV, USA, 1–6.

[30] Christian Cervantes, Diego Poplade, Michele Nogueira, and Aldri Santos. 2015. Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, Ottawa, ON, Canada, 606–611.

[31] Jair Cervantes, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, and Asdrubal Lopez. 2020. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing* 408 (2020), 189–215.

[32] S Chakrabarti, Mohuya Chakraborty, and Indraneel Mukhopadhyay. 2010. Study of snort-based IDS. In *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*. Association for Computing Machinery, Mumbai, Maharashtra, India, 43–47.

[33] Sumit Chakravarty and Andrew Hopkins. 2020. LoRa mesh network with BeagleBone Black. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, London, UK, 306–311.

[34] Po-Yu Chen, Shusen Yang, Julie A McCann, Jie Lin, and Xinyu Yang. 2015. Detection of false data injection attacks in smart-grid systems. *IEEE Communications Magazine* 53, 2 (2015), 206–213.

[35] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Honolulu, HI, USA, 1251–1258.

[36] Mauro Conti, Nicola Dragoni, and Viktor Lesyk. 2016. A survey of man in the middle attacks. *IEEE Communications Surveys & Tutorials* 18, 3 (2016), 2027–2051.

[37] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. 2008. Supervised learning. , 21–49 pages.

[38] Padraig Cunningham and Sarah Jane Delany. 2021. K-nearest neighbour classifiers-a tutorial. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–25.

[39] Alfred DeMaris. 1995. A tutorial in logistic regression. *Journal of Marriage and the Family* 57, 4 (1995), 956–968.

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, Miami, FL, USA, 248–255.

[41] Dorothy E Denning. 1987. An intrusion-detection model. *IEEE Transactions on software engineering* SE-13, 2 (1987), 222–232.

[42] Yau Ti Duna, Mohd Faizal Ab Razaka, Mohamad Fadli Zolkiplib, Tan Fui Beea, and Ahmad Firdausa. 2021. Grasp on Next Generation Security Operation Centre (NGSOC): Comparative Study. *Int. J. Nonlinear Anal. Appl* 12, 2 (2021), 867–896.

[43] Mica R Endsley. 1995. Toward a theory of situation awareness in dynamic systems. *Human factors* 37, 1 (1995), 32–64.

[44] Mustafa Amir Faisal, Zeyar Aung, John R Williams, and Abel Sanchez. 2014. Data-stream-based intrusion detection system for advanced metering infrastructure in smart grid: A feasibility study. *IEEE Systems journal* 9, 1 (2014), 31–44.

[45] Zihao Feng, Sujuan Qin, Xuesong Huo, Pei Pei, Ye Liang, and Liming Wang. 2016. Snort improvement on PROFINET RT for industrial control system intrusion detection. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, Chengdu, China, 942–946.

[46] Tiago M Fernández-Caramés. 2019. From pre-quantum to post-quantum IoT security: A survey on quantum-resistant cryptosystems for the Internet of Things. *IEEE Internet of Things Journal* 7, 7 (2019), 6457–6480.

[47] Salvador García, Julián Luengo, and Francisco Herrera. 2015. Data preprocessing in data mining.

[48] Benyamin Ghojogh, Ali Ghodsi, Fakhri Karray, and Mark Crowley. 2021. Generative adversarial networks and adversarial autoencoders: Tutorial and survey. *arXiv preprint arXiv:2111.13282* 0, 0 (2021), 1–37.

[49] Sheetal Gokhale, Ashwini Dalvi, and Irfan Siddavatam. 2020. Industrial Control Systems Honeypot: A Formal Analysis of Conpot. *International Journal of Computer Network & Information Security* 12, 6 (2020), 1–13.

[50] Niv Goldenberg and Avishai Wool. 2013. Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. *international journal of critical infrastructure protection* 6, 2 (2013), 63–75.

[51] Gustavo González-Granadillo, Susana González-Zarzosa, and Rodrigo Diaz. 2021. Security information and event management (SIEM): analysis, trends, and usage in critical infrastructures. *Sensors* 21, 14 (2021), 4759.

[52] Muthukrishnan Gowri and Balasubramanian Paramasivan. 2016. Rule-Based Anomaly Detection Technique Using Roaming Honeypots for Wireless Sensor Networks. *ETRI Journal* 38, 6 (2016), 1145–1152.

[53] Vincenzo Gulisano, Magnus Almgren, and Marina Papatriantafilou. 2014. Metis: a two-tier intrusion detection system for advanced metering infrastructures. In *International Conference on Security and Privacy in Communication Networks*. Springer, Cambridge, UK, 51–68.

[54] Dilara Gümüşbaş, Tulay Yıldırım, Angelo Genovese, and Fabio Scotti. 2020. A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. *IEEE Systems Journal* 15, 2 (2020), 1717–1731.

[55] Naman Gupta, Vinayak Naik, and Srishti Sengupta. 2017. A firewall for internet of things. In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*. IEEE, Bengaluru, India, 411–412.

[56] Hamed HaddadPajouh, Ali Dehghantanha, Reza M Parizi, Mohammed Aledhari, and Hadis Karimipour. 2019. A survey on internet of things security: Requirements, challenges, and solutions. *Internet of Things* 14 (2019), 100129.

[57] Oliver Hahm, Emmanuel Baccelli, Hauke Petersen, and Nicolas Tsiftes. 2015. Operating systems for low-end devices in the internet of things: a survey. *IEEE Internet of Things Journal* 3, 5 (2015), 720–734.

[58] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11, 1 (2009), 10–18.

[59] Botao Hao, Yasin Abbasi-Yadkori, Zheng Wen, and Guang Cheng. 2019. Bootstrapping upper confidence bound.

[60] Sahand Hariri, Matias Carrasco Kind, and Robert J Brunner. 2019. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering* 33, 4 (2019), 1479–1489.

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* IEEE, Las Vegas, NV, USA, 770–778.

[62] Ersi Hodo, Stepan Grebeniuk, Henri Ruotsalainen, and Paul Tavolato. 2017. Anomaly detection for simulated iec-60870-5-104 trafiic. In *Proceedings of the 12th International Conference on Availability, Reliability and Security.* ACM, Reggio Calabria, Italy, 1–7.

[63] Junho Hong, Chen-Ching Liu, and Manimaran Govindarasu. 2014. Detection of cyber intrusions using network-based multicast messages for substation automation. In *ISGT 2014.* IEEE, Washington, DC, USA, 1–5.

[64] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications.

[65] Marian Hristov, Maria Nenova, Georgi Iliev, and Dimiter Avresky. 2021. Integration of Splunk Enterprise SIEM for DDoS Attack Detection in IoT. In *2021 IEEE 20th International Symposium on Network Computing and Applications (NCA).* IEEE, Boston, MA, USA, 1–5.

[66] Yih-Chun Hu, Adrian Perrig, and David B Johnson. 2006. Wormhole attacks in wireless networks. *IEEE journal on selected areas in communications* 24, 2 (2006), 370–380.

[67] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* IEEE, Honolulu, HI, USA, 4700–4708.

[68] Mia Hubert and Michiel Debruyne. 2010. Minimum covariance determinant. *Wiley interdisciplinary reviews: Computational statistics* 2, 1 (2010), 36–43.

[69] René Hummen, Jens Hiller, Hanno Wirtz, Martin Henze, Hossein Shafagh, and Klaus Wehrle. 2013. 6LoWPAN fragmentation attacks and mitigation mechanisms. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks.* ACM, Budapest, Hungary, 55–66.

[70] SM Suhail Hussain, Shaik Mullapathi Farooq, and Taha Selim Ustun. 2021. A Security Mechanism for IEEE C37. 118.2 PMU Communication. *IEEE Transactions on Industrial Electronics* 69, 1 (2021), 1053–1061.

[71] SM Suhail Hussain, Taha Selim Ustun, and Akhtar Kalam. 2019. A review of IEC 62351 security mechanisms for IEC 61850 message exchanges. *IEEE Transactions on Industrial Informatics* 16, 9 (2019), 5643–5654.

[72] Muhammad Azhar Iqbal, Sajjad Hussain, Huanlai Xing, and Muhammad Ali Imran. 2021. *IoT Protocol Stack.* Wiley-IEEE Press, Hoboken, New Jersey, United States. 93–126 pages. https://doi.org/10.1002/9781119701460.ch5

[73] Waseem Iqbal, Haider Abbas, Mahmoud Daneshmand, Bilal Rauf, and Yawar Abbas Bangash. 2020. An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security. *IEEE Internet of Things Journal* 7, 10 (2020), 10250–10276.

[74] Celine Irvene, David Formby, Samuel Litchfield, and Raheem Beyah. 2017. HoneyBot: A honeypot for robotic systems. *Proc. IEEE* 106, 1 (2017), 61–70.

[75] Farhana Javed, Muhamamd Khalil Afzal, Muhammad Sharif, and Byung-Seo Kim. 2018. Internet of Things (IoT) operating systems support, networking technologies, applications, and challenges: A comparative review. *IEEE Communications Surveys & Tutorials* 20, 3 (2018), 2062–2100.

[76] Liangxiao Jiang, Harry Zhang, and Zhihua Cai. 2008. A novel bayes model: Hidden naive bayes. *IEEE Transactions on knowledge and data engineering* 21, 10 (2008), 1361–1371.

[77] Paria Jokar and Victor CM Leung. 2016. Intrusion detection and prevention for ZigBee-based home area networks in smart grids. *IEEE Transactions on Smart Grid* 9, 3 (2016), 1800–1811.

[78] Maelle Kabir-Querrec, Stephane Mocanu, Jean-Marc Thiriet, and Eric Savary. 2015. Power utility automation cybersecurity: IEC 61850 specification of an intrusion detection function. In *ESREL 2015-25th European Safety and Reliability Conference*. CRC Press, Zurich, Switzerland, 1–9.

[79] BooJoong Kang, Kieran McLaughlin, and Sakir Sezer. 2016. Towards a stateful analysis framework for smart grid network intrusion detection. In *4th International Symposium for ICS & SCADA Cyber Security Research 2016 4*. ScienceOpen, Belfast, UK, 124–131.

[80] Eirini Karapistoli, Panagiotis Sarigiannidis, and Anastasios A Economides. 2013. Srnet: a real-time, cross-based anomaly detection and visualization system for wireless sensor networks. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security*. Association for Computing Machinery, Atlanta, Georgia, USA, 49–56.

[81] Eirini Karapistoli, Panagiotis Sarigiannidis, and Anastasios A Economides. 2014. Visual-assisted wormhole attack detection for wireless sensor networks. In *International Conference on Security and Privacy in Communication Networks*. Springer, Beijing, China, 222–238.

[82] Nickson M Karie, Nor Masri Sahri, Wencheng Yang, Craig Valli, and Victor R Kebande. 2021. A Review of Security Standards and Frameworks for IoT-Based Smart Environments. *IEEE Access* 9 (2021), 121975–121995.

[83] Michael N Katehakis and Arthur F Veinott Jr. 1987. The multi-armed bandit problem: decomposition and computation. *Mathematics of Operations Research* 12, 2 (1987), 262–268.

[84] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. 2020. A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review* 53, 8 (2020), 5455–5516.

[85] Rafiullah Khan, Abdullah Albalushi, Kieran McLaughlin, David Laverty, and Sakir Sezer. 2017. Model based intrusion detection system for synchrophasor applications in smart grid. In *2017 IEEE Power & Energy Society General Meeting*. IEEE, Chicago, IL, USA, 1–5.

[86] HyunGon Kim. 2008. Protection against packet fragmentation attacks at 6LoWPAN adaptation layer. In *2008 International Conference on Convergence and Hybrid Information Technology*. IEEE, Daejeon, Korea (South), 796–801.

[87] Taehong Kim et al. 2018. A study of the Z-wave protocol: implementing your own smart home gateway. In *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*. IEEE, Nagoya, Japan, 411–415.

[88] Nikos Komninos, Eleni Philippou, and Andreas Pitsillides. 2014. Survey in smart grid and smart home security: Issues, challenges and countermeasures. *IEEE Communications Surveys & Tutorials* 16, 4 (2014), 1933–1954.

[89] Ioannis Koniaris, Georgios Papadimitriou, and Petros Nicopolitidis. 2013. Analysis and visualization of SSH attacks using honeypots. In *Eurocon 2013*. IEEE, Zagreb, Croatia, 65–72.

[90] Barbara Kordy, Sjouke Mauw, Saša Radomirović, and Patrick Schweitzer. 2014. Attack–defense trees. *Journal of Logic and Computation* 24, 1 (2014), 55–87.

[91] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. 2008. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, Las Vegas, Nevada, USA, 444–452.

[92] Volodymyr Kuleshov and Doina Precup. 2014. Algorithms for multi-armed bandit problems.

[93] Durgesh Kumar and Vivek Vashishtha. 2012. Snort based h-ids with kf sensor and weka.

[94] Punit Kumar and Atul Gupta. 2020. Active learning query strategies for classification, regression, and clustering: a survey. *Journal of Computer Science and Technology* 35, 4 (2020), 913–945.

[95] Aditya Kuppa and Nhien-An Le-Khac. 2020. Black box attacks on explainable artificial intelligence (XAI) methods in cyber security. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Glasgow, UK, 1–8.

[96] Aditya Kuppa and Nhien-An Le-Khac. 2021. Adversarial xai methods in cybersecurity. *IEEE Transactions on Information Forensics and Security* 16 (2021), 4924–4938.

[97] Giwon Kwon, Jeehyeong Kim, Jaewon Noh, and Sunghyun Cho. 2016. Bluetooth low energy security vulnerability and improvement method. In *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, Seoul, Korea (South), 1–4.

[98] Sungmoon Kwon, Hyunguk Yoo, and Taeshik Shon. 2020. IEEE 1815.1-based power system security with bidirectional RNN-based network anomalous attack detection for cyber-physical system. *IEEE Access* 8 (2020), 77572–77586.

[99] YooJin Kwon, Huy Kang Kim, Yong Hun Lim, and Jong In Lim. 2015. A behavior-based intrusion detection technique for smart grid infrastructure. In *2015 IEEE Eindhoven PowerTech*. IEEE, Eindhoven, Netherlands, 1–6.

[100] Thi-Thu-Huong Le, Yustus Eko Oktian, and Howon Kim. 2022. XGBoost for imbalanced multiclass classification-based industrial internet of things intrusion detection systems. *Sustainability* 14, 14 (2022), 8707.

[101] Gyu-min Lee, Cheol-woong Lee, and Byeong-hee Roh. 2022. QoS Support Path Selection for Inter-Domain Flows Using Effective Delay and Directed Acyclic Graph in Multi-Domain SDN. *Electronics* 11, 14 (2022), 2245.

[102] Corrado Leita, VH Pham, Olivier Thonnard, E Ramirez-Silva, Fabian Pouget, Engin Kirda, and Marc Dacier. 2008. The leurre. com project: collecting internet threats information using a worldwide distributed honeynet. In *2008 WOMBAT Workshop on Information Security Threats Data Collection and Sharing*. IEEE, Amsterdam, Netherlands, 40–57.

[103] Antoine Lemay, Joan Calvet, François Menet, and José M Fernandez. 2018. Survey of publicly available reports on advanced persistent threat actors. *Computers & Security* 72 (2018), 26–59.

[104] Antoine Lemay and Jose M. Fernandez. 2016. Providing SCADA Network Data Sets for Intrusion Detection Research. In *9th Workshop on Cyber Security Experimentation and Test (CSET 16)*. USENIX Association, Austin, TX, 1–8. `https://www.usenix.org/conference/cset16/workshop-program/presentation/lemay`

[105] Erxia Li, Chaoqun Kang, Deyu Huang, Modi Hu, Fangyuan Chang, Lianjie He, and Xiaoyong Li. 2019. Quantitative Model of Attacks on Distribution Automation Systems Based on CVSS and Attack Trees. *Information* 10, 8 (2019), 251.

[106] Hao Li, Guangjie Liu, Weiwei Jiang, and Yuewei Dai. 2015. Designing snort rules to detect abnormal DNP3 network data. In *2015 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, Changshu, 343–348.

[107] Kun-Lun Li, Hou-Kuan Huang, Sheng-Feng Tian, and Wei Xu. 2003. Improving one-class SVM for anomaly detection. In *Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693)*, Vol. 5. IEEE, Xian, 3077–3081.

[108] Szu-Chuang Li, Yennun Huang, Bo-Chen Tai, and Chi-Ta Lin. 2017. Using data mining methods to detect simulated intrusions on a modbus network. In *2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2)*. IEEE, Kanazawa, Japan, 143–148.

[109] Yuancheng Li, Rixuan Qiu, and Sitong Jing. 2018. Intrusion detection system using Online Sequence Extreme Learning Machine (OS-ELM) in advanced metering infrastructure of smart grid. *PloS one* 13, 2 (2018), e0192216.

[110] Hui Lin, Chen Chen, Jianhui Wang, Junjian Qi, Dong Jin, Zbigniew T Kalbarczyk, and Ravishankar K Iyer. 2016. Self-healing attack-resilient PMU network for power system operation. *IEEE Transactions on Smart Grid* 9, 3 (2016), 1551–1565.

[111] Fernando A Aires Lins and Marco Vieira. 2020. Security requirements and solutions for iot gateways: A comprehensive study. *IEEE Internet of Things Journal* 8, 11 (2020), 8667–8679.

[112] Xiaoxue Liu, Peidong Zhu, Yan Zhang, and Kan Chen. 2015. A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure. *IEEE Transactions on Smart Grid* 6, 5 (2015), 2435–2443.

[113] Yuxin Liu, Ming Ma, Xiao Liu, Neal N Xiong, Anfeng Liu, and Ying Zhu. 2018. Design and analysis of probing route to defense sink-hole attacks for Internet of Things security. *IEEE Transactions on Network Science and Engineering* 7, 1 (2018), 356–372.

[114] Susan Lomax and Sunil Vadera. 2013. A survey of cost-sensitive decision tree induction algorithms. *ACM Computing Surveys (CSUR)* 45, 2 (2013), 1–35.

[115] Sushant Manchekar, Makarand Kadam, and Krantee Jamdaade. 2018. Application of honeypot in cloud security: A review. *International Journal on Future Revolution in Computer Science & Communication Engineering* 4, 6 (2018), 63–65.

[116] Pedro Manso, José Moura, and Carlos Serrão. 2019. SDN-based intrusion detection system for early detection and mitigation of DDoS attacks. *Information* 10, 3 (2019), 106.

[117] Stefan Marksteiner, Víctor Juan Expósito Jiménez, Heribert Valiant, and Herwig Zeiner. 2017. An overview of wireless IoT protocol security in the smart home domain. In *2017 Internet of Things Business Models, Users, and Networks*. IEEE, Copenhagen, Denmark, 1–8.

[118] Daniele Midi, Antonino Rullo, Anand Mudgerikar, and Elisa Bertino. 2017. Kalis—A system for knowledge-driven adaptable intrusion detection for the Internet of Things. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, Atlanta, GA, USA, 656–666.

[119] Robert Mitchell and Ray Chen. 2013. Behavior-rule based intrusion detection systems for safety critical smart grid applications. *IEEE Transactions on Smart Grid* 4, 3 (2013), 1254–1263.

[120] Thomas H Morris, Bryan A Jones, Rayford B Vaughn, and Yoginder S Dandass. 2013. Deterministic intrusion detection rules for MODBUS protocols. In *2013 46th Hawaii International Conference on System Sciences*. IEEE, Wailea, HI, USA, 1773–1781.

[121] Marcin Nawrocki, Matthias Wählisch, Thomas C Schmidt, Christian Keil, and Jochen Schönfelder. 2016. A survey on honeypot software and data analysis. *arXiv preprint arXiv:1608.06249* 0, 0 (2016), 1–38.

[122] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. 2020. What is being transferred in transfer learning?

[123] Quoc-Dung Ngo, Huy-Trung Nguyen, Van-Hoang Le, and Doan-Hieu Nguyen. 2020. A survey of IoT malware and detection methods based on static features. *ICT Express* 6, 4 (2020), 280–286.

[124] Mike O Ojo, Stefano Giordano, Gregorio Procissi, and Ilias N Seitanidis. 2018. A review of low-end, middle-end, and high-end IoT devices. *IEEE Access* 6 (2018), 70528–70554.

[125] Seiichi Ozawa, Tao Ban, Naoki Hashimoto, Junji Nakazato, and Jumpei Shimamura. 2020. A study of IoT malware activities using association rule learning for darknet sensor data. *International Journal of Information Security* 19, 1 (2020), 83–92.

[126] Shantanu Pal, Michael Hitchens, Tahiry Rabehaja, and Subhas Mukhopadhyay. 2020. Security requirements for the internet of things: A systematic approach. *Sensors* 20, 20 (2020), 5897.

[127] BM Pampapathi, Nageswara Guptha, and MS Hema. 2022. Towards an effective deep learning-based intrusion detection system in the internet of things. *Telematics and Informatics Reports* 7 (2022), 100009.

[128] Shengyi Pan, Thomas Morris, and Uttam Adhikari. 2015. Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid* 6, 6 (2015), 3104–3113.

[129] Ahmed Patel, Hitham Alhussian, Jens Myrup Pedersen, Bouchaib Bounabat, Joaquim Celestino Júnior, and Sokratis Katsikas. 2017. A nifty collaborative intrusion detection and prevention architecture for smart grid ecosystems. *Computers & Security* 64 (2017), 92–109.

[130] Jesus Arturo Perez-Diaz, Ismael Amezcua Valdovinos, Kim-Kwang Raymond Choo, and Dakai Zhu. 2020. A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. *IEEE Access* 8 (2020), 155859–155872.

[131] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. 2018. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)* 51, 5 (2018), 1–36.

[132] Upeka Kanchana Premaratne, Jagath Samarabandu, Tarlochan S Sidhu, Robert Beresh, and Jian-Cheng Tan. 2010. An intrusion detection system for IEC61850 automated substations. *IEEE Transactions on Power Delivery* 25, 4 (2010), 2376–2383.

[133] Adnan Qayyum, Muhammad Usama, Junaid Qadir, and Ala Al-Fuqaha. 2020. Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward. *IEEE Communications Surveys & Tutorials* 22, 2 (2020), 998–1026.

[134] Panagiotis Radoglou-Grammatikis, Athanasios Liatifis, Elisavet Grigoriou, Theocharis Saoulidis, Antonios Sarigiannidis, Thomas Lagkas, and Panagiotis Sarigiannidis. 2021. Trusty: A solution for threat hunting using data analysis in critical infrastructures. In *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, Rhodes, Greece, 485–490.

[135] Panagiotis Radoglou-Grammatikis and Panagiotis Sarigiannidis. 2021. *Chapter 5. Network Attacks - Cyber-Security Threats, Actors, and Dynamic Mitigation.* CRC Press, Boca Raton, Florida, United States.

[136] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, Christos Dalamagkas, Yannis Spyridis, Thomas Lagkas, Georgios Efstathopoulos, Achilleas Sesis, Ignacio Labrador Pavon, Ruben Trapero Burgos, Rodrigo Diaz, et al. 2021. SDN-based resilient smart grid: the SDN-microSENSE architecture. *Digital* 1, 4 (2021), 173–187.

[137] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, Panagiotis Diamantoulakis, Thomas Lagkas, Theocharis Saoulidis, Eleftherios Fountoukidis, and George Karagiannidis. 2022. Strategic honeypot deployment in ultra-dense beyond 5g networks: A reinforcement learning approach. *IEEE Transactions on Emerging Topics in Computing* -, - (2022), 1–12.

[138] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, George Efstathopoulos, Paris-Alexandros Karypidis, and Antonios Sarigiannidis. 2020. Diderot: an intrusion detection and prevention system for dnp3-based scada systems. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*. ACM, Virtual Event, Ireland, 1–8.

[139] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, Georgios Efstathopoulos, Thomas Lagkas, George Fragulis, and Antonios Sarigiannidis. 2021. A self-learning approach for detecting intrusions in healthcare systems. In *ICC 2021-IEEE International Conference on Communications*. IEEE, Montreal, QC, Canada, 1–6.

[140] Panagiotis Radoglou Grammatikis, Panagiotis Sarigiannidis, Georgios Efstathopoulos, and Emmanouil Panaousis. 2020. ARIES: A novel multivariate intrusion detection system for smart grid. *Sensors* 20, 18 (2020), 5305.

[141] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, Ioannis Giannoulakis, Emmanouil Kafetzakis, and Emmanouil Panaousis. 2019. Attacking iec-60870-5-104 scada systems. In *2019 IEEE World Congress on Services (SERVICES)*, Vol. 2642. IEEE, Milan, Italy, 41–46.

[142] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, Eider Iturbe, Erkuden Rios, Antonios Sarigiannidis, Odysseas Nikolis, Dimosthenis Ioannidis, Vasileios Machamint, Michalis Tzifas, Alkiviadis Giannakoulias, et al. 2020. Secure and private smart grid: The spear architecture. In *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. IEEE, Ghent, Belgium, 450–456.

[143] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, Thanasis Liatifis, Tryfon Apostolakos, and Spyridon Oikonomou. 2018. An overview of the firewall systems in the smart grid paradigm. In *2018 Global information infrastructure and networking symposium (GIIS)*. IEEE, Thessaloniki, Greece, 1–4.

[144] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, Antonios Sarigiannidis, Dimitrios Margounakis, Apostolos Tsiakalos, and Georgios Efstathopoulos. 2020. An anomaly detection mechanism for IEC 60870-5-104. In *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE, Bremen, Germany, 1–4.

[145] Panagiotis Radoglou-Grammatikis, Panagiotis G Sarigiannidis, and Ioannis D Moscholios. 2019. Securing the Internet of Things: Challenges, threats and solutions. *Internet of Things* 5 (2019), 41–70.

[146] Panagiotis Radoglou-Grammatikis, Ilias Siniosoglou, Thanasis Liatifis, Anastasios Kourouniadis, Konstantinos Rompolos, and Panagiotis Sarigiannidis. 2020. Implementation and detection of modbus cyberattacks. In *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE, Bremen, Germany, 1–4.

[147] Panagiotis I Radoglou-Grammatikis and Panagiotis G Sarigiannidis. 2018. An anomaly-based intrusion detection system for the smart grid based on cart decision tree. In *2018 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, Thessaloniki, Greece, 1–5.

[148] Panagiotis I Radoglou-Grammatikis and Panagiotis G Sarigiannidis. 2019. Securing the smart grid: A comprehensive compilation of intrusion detection and prevention systems. *IEEE Access* 7 (2019), 46595–46620.

[149] Sawsan Abdul Rahman, Hanine Tout, Chamseddine Talhi, and Azzam Mourad. 2020. Internet of things intrusion detection: Centralized, on-device, or federated learning? *IEEE Network* 34, 6 (2020), 310–317.

[150] Atul Rawal, James McCoy, Danda B Rawat, Brian M Sadler, and Robert St Amant. 2021. Recent Advances in Trustworthy Explainable Artificial Intelligence: Status, Challenges, and Perspectives. *IEEE Transactions on Artificial Intelligence* 3, 6 (2021), 852–866.

[151] Shahid Raza, Linus Wallgren, and Thiemo Voigt. 2013. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad hoc networks* 11, 8 (2013), 2661–2674.

[152] Mubashir Husain Rehmani, Fayaz Akhtar, Alan Davy, and Brendan Jennings. 2018. Achieving resilience in sdn-based smart grid: A multi-armed bandit approach. In *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. IEEE, Montreal, QC, Canada, 366–371.

[153] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. 2020. A survey of deep active learning.

[154] Paulo Angelo Alves Resende and André Costa Drummond. 2018. A survey of random forest based methods for intrusion detection systems. *ACM Computing Surveys (CSUR)* 51, 3 (2018), 1–36.

[155] S Revathi and A Malathi. 2013. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *International Journal of Engineering Research & Technology (IJERT)* 2, 12 (2013), 1848–1853.

[156] Brecht Reynders, Wannes Meert, and Sofie Pollin. 2016. Range and coexistence analysis of long range unlicensed communication. In *2016 23rd International Conference on Telecommunications (ICT)*. IEEE, Thessaloniki, Greece, 1–6.

[157] Markus Ringnér. 2008. What is principal component analysis? *Nature biotechnology* 26, 3 (2008), 303–304.

[158] Erkuden Rios, Angel Rego, Eider Iturbe, Marivi Higuero, and Xabier Larrucea. 2020. Continuous Quantitative Risk Management in Smart Grids Using Attack Defense Trees. *Sensors* 20, 16 (2020), 4404.

[159] Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. 2004. Are loss functions all the same? *Neural computation* 16, 5 (2004), 1063–1076.

[160] Krzysztof Rusek, José Suárez-Varela, Paul Almasan, Pere Barlet-Ros, and Albert Cabellos-Aparicio. 2020. RouteNet: Leveraging Graph Neural Networks for network modeling and optimization in SDN. *IEEE Journal on Selected Areas in Communications* 38, 10 (2020), 2260–2270.

[161] Hans Sagan. 2012. Space-filling curves.

[162] Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1249.

[163] Fatih Sakiz and Sevil Sen. 2017. A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV. *Ad Hoc Networks* 61 (2017), 33–50.

[164] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Salt Lake City, UT, USA, 4510–4520.

[165] Panagiotis Sarigiannidis, Eirini Karapistoli, and Anastasios A Economides. 2015. Detecting Sybil attacks in wireless sensor networks using UWB ranging-based information. *expert systems with applications* 42, 21 (2015), 7560–7572.

[166] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. 2017. A review of clustering techniques and developments. *Neurocomputing* 267 (2017), 664–681.

[167] Alexander K Seewald and Wilfried N Gansterer. 2010. On the detection and identification of botnets. *Computers & Security* 29, 1 (2010), 45–58.

[168] Guilherme Serpa Sestito, Afonso Celso Turcato, Andre Luis Dias, Murilo Silveira Rocha, Maira Martins da Silva, Paolo Ferrari, and Dennis Brandao. 2017. A method for anomalies detection in real-time ethernet data traffic applied to profinet. *IEEE Transactions on Industrial Informatics* 14, 5 (2017), 2171–2180.

[169] Vasu Sethia and A Jeyasekar. 2019. Malware capturing and analysis using dionaea honeypot. In *2019 International Carnahan Conference on Security Technology (ICCST)*. IEEE, Chennai, India, 1–4.

[170] Burr Settles. 2009. Active learning literature survey.

[171] Mridula Sharma, Haytham Elmiligi, Fayez Gebali, and Abhishek Verma. 2019. Simulating attacks for rpl and generating multi-class dataset for supervised machine learning. In *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, Vancouver, BC, Canada, 0020–0026.

[172] Devu Manikantan Shila, Yu Cheng, and Tricha Anjali. 2010. Mitigating selective forwarding attacks with a channel-aware approach in WMNs. *IEEE transactions on wireless communications* 9, 5 (2010), 1661–1675.

[173] Adam Shostack. 2014. Threat modeling: Designing for security.

[174] Rajesh Kumar Shrivastava, Bazila Bashir, and Chittaranjan Hota. 2019. Attack detection and forensics using honeypot in IoT environment. In *International Conference on Distributed Computing and Internet Technology*. Springer, Bhubaneswar, India, 402–409.

[175] Raúl Siles. 2007. Honeyspot: The wireless honeypot.

[176] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition.

[177] Kiran Jot Singh and Divneet Singh Kapoor. 2017. Create Your Own Internet of Things: A survey of IoT platforms. *IEEE Consumer Electronics Magazine* 6, 2 (2017), 57–68.

[178] T Aditya Sai Srinivas and SS Manivannan. 2020. Prevention of hello flood attack in IoT using combination of deep learning with improved rider optimization algorithm. *Computer Communications* 163 (2020), 162–175.

[179] William Stallings, Lawrie Brown, Michael D Bauer, and Arup Kumar Bhattacharjee. 2012. *Computer security: principles and practice*. Pearson Education, 221 River St, Hoboken, NJ 07030, US.

[180] Julia E Sullivan and Dmitriy Kamensky. 2017. How cyber-attacks in Ukraine show the vulnerability of the US power grid. *The Electricity Journal* 30, 3 (2017), 30–35.

[181] M Surendar and A Umamakeswari. 2016. Indres: An intrusion detection and response system for internet of things with 6lowpan. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, Chennai, India, 1903–1908.

[182] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. , 6105–6114 pages.

[183] Song Tan, Debraj De, Wen-Zhan Song, Junjie Yang, and Sajal K Das. 2016. Survey of security advances in smart grid: A data driven approach. *IEEE Communications Surveys & Tutorials* 19, 1 (2016), 397–422.

[184] Jiexiong Tang, Chenwei Deng, and Guang-Bin Huang. 2015. Extreme learning machine for multilayer perceptron. *IEEE transactions on neural networks and learning systems* 27, 4 (2015), 809–821.

[185] Koen Tange, Michele De Donno, Xenofon Fafoutis, and Nicola Dragoni. 2020. A systematic survey of industrial Internet of Things security: Requirements and fog computing opportunities. *IEEE Communications Surveys & Tutorials* 22, 4 (2020), 2489–2520.

[186] Diogo Teixeira, Leonardo Assunção, Teresa Pereira, Silvestre Malta, and Pedro Pinto. 2019. OSSEC IDS Extension to Improve Log Analysis and Override False Positive or Negative Detections. *Journal of Sensor and Actuator Networks* 8, 3 (2019), 46.

[187] Alaa Tharwat. 2016. Linear vs. quadratic discriminant analysis classifier: a tutorial. *International Journal of Applied Pattern Recognition* 3, 2 (2016), 145–180.

[188] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. 2017. Linear discriminant analysis: A detailed tutorial. *AI communications* 30, 2 (2017), 169–190.

[189] Aparna Tiwari and Dinesh Kumar. 2020. Comparitive study of various honeypot tools on the basis of their classification & features. In *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*. SSRN, Delhi, India, 1–6.

[190] Samaneh Torkzadeh, Hadi Soltanizadeh, and Ali A Orouji. 2021. Energy-aware routing considering load balancing for SDN: a minimum graph-based Ant Colony Optimization. *Cluster Computing* 24, 3 (2021), 2293–2312.

[191] Meenakshi Tripathi, Manoj Singh Gaur, and Vijay Laxmi. 2013. Comparing the impact of black hole and gray hole attack on LEACH in WSN. *Procedia Computer Science* 19 (2013), 1101–1107.

[192] Fan-Hsun Tseng, Li-Der Chou, and Han-Chieh Chao. 2011. A survey of black hole attacks in wireless mobile ad hoc networks. *Human-centric Computing and Information Sciences* 1, 1 (2011), 1–16.

[193] Michail Tsikerdekis, Sherali Zeadally, Amy Schlesener, and Nicolas Sklavos. 2018. Approaches for preventing honeypot detection and compromise. In *2018 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, Thessaloniki, Greece, 1–6.

[194] Niki Tsitsiroudi, Panagiotis Sarigiannidis, Eirini Karapistoli, and Anastasios A Economides. 2016. EyeSim: A mobile application for visual-assisted wormhole attack detection in IoT-enabled WSNs. In *2016 9th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, Colmar, France, 103–109.

[195] Robert Udd, Mikael Asplund, Simin Nadjm-Tehrani, Mehrdad Kazemtabrizi, and Mathias Ekstedt. 2016. Exploiting bro for intrusion detection in a SCADA system. In *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*. Association for Computing Machinery, New York, NY, USA, 44–51.

[196] Imtiaz Ullah and Qusay H Mahmoud. 2017. An intrusion detection framework for the smart grid. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, Windsor, ON, Canada, 1–5.

[197] Jesper E Van Engelen and Holger H Hoos. 2020. A survey on semi-supervised learning. *Machine Learning* 109, 2 (2020), 373–440.

[198] Niko Vidgren, Keijo Haataja, Jose Luis Patino-Andres, Juan Jose Ramirez-Sanchis, and Pekka Toivanen. 2013. Security threats in ZigBee-enabled systems: vulnerability evaluation, practical experiments, countermeasures, and lessons learned. In *2013 46th Hawaii International Conference on System Sciences*. IEEE, Wailea, HI, USA, 5132–5138.

[199] R Vijayanand, D Devaraj, and B Kannapiran. 2017. Support vector machine based intrusion detection system with reduced input features for advanced metering infrastructure of smart grid. In *2017 4th International conference on advanced computing and communication systems (ICACCS)*. IEEE, Coimbatore, India, 1–7.

[200] Linus Wallgren, Shahid Raza, and Thiemo Voigt. 2013. Routing attacks and countermeasures in the RPL-based internet of things. *International Journal of Distributed Sensor Networks* 9, 8 (2013), 794326.

[201] Pin-Han Wang, I-En Liao, Kuo-Fong Kao, and Jyun-Yao Huang. 2018. An intrusion detection method based on log sequence clustering of honeypot for modbus tcp protocol. In *2018 IEEE International Conference on Applied System Invention (ICASI)*. IEEE, Chiba, Japan, 255–258.

[202] Martin Wattenberg. 2005. A note on space-filling visualizations and space-filling curves. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* IEEE, Minneapolis, MN, USA, 181–186.

[203] Kevin Wong, Craig Dillabaugh, Nabil Seddigh, and Biswajit Nandy. 2017. Enhancing Suricata intrusion detection system for cyber security in SCADA networks. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, Windsor, ON, Canada, 1–5.

[204] Junfeng Xie, F Richard Yu, Tao Huang, Renchao Xie, Jiang Liu, Chenmeng Wang, and Yunjie Liu. 2018. A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. *IEEE Communications Surveys & Tutorials* 21, 1 (2018), 393–430.

[205] Yanghao Xie, Lin Huang, Yuyang Kong, Sheng Wang, Shizhong Xu, Xiong Wang, and Jing Ren. 2021. Virtualized Network Function Forwarding Graph Placing in SDN and NFV-Enabled IoT Networks: A Graph Neural Network Assisted Deep Reinforcement Learning Method. *IEEE Transactions on Network and Service Management* 19, 1 (2021), 524–537.

[206] Tianyi Xing, Zhengyang Xiong, Dijiang Huang, and Deep Medhi. 2014. SDNIPS: Enabling software-defined networking based intrusion prevention system in clouds. In *10th International Conference on Network and Service Management (CNSM) and Workshop*. IEEE, Rio de Janeiro, Brazil, 308–311.

[207] Leilei Xiong and Santiago Grijalva. 2019. N-1 RTU Cyber-Physical Security Assessment Using State Estimation. In *2019 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, Atlanta, GA, USA, 1–5.

[208] Hua Yang, Tao Li, Xinlei Hu, Feng Wang, and Yang Zou. 2014. A survey of artificial immune system based intrusion detection. *The Scientific World Journal* 2014 (2014), 1–12.

[209] Yi Yang, Kieran McLaughlin, Lei Gao, Sakir Sezer, Yubo Yuan, and Yanfeng Gong. 2016. Intrusion detection system for IEC 61850 based smart substations. In *2016 IEEE Power and Energy Society General Meeting (PESGM)*. IEEE, Boston, MA, 1–5.

[210] Yi Yang, Kieran McLaughlin, Tim Littler, Sakir Sezer, Bernardi Pranggono, and HF Wang. 2013. Intrusion detection system for IEC 60870-5-104 based SCADA networks. In *2013 IEEE power & energy society general meeting*. IEEE, Vancouver, BC, Canada, 1–5.

[211] Yu Yang, Kieran McLaughlin, Sakir Sezer, Timothy Littler, Bernardi Pranggono, Paul Brogan, and HF Wang. 2013. Intrusion detection system for network security in synchrophasor systems. , 246-252 pages.

[212] Y Yang, K McLaughlin, S Sezer, YB Yuan, and W Huang. 2014. Stateful intrusion detection for IEC 60870-5-104 SCADA security. In *2014 IEEE PES General Meeting— Conference & Exposition*. IEEE, National Harbor, MD, USA, 1–5.

[213] Yi Yang, Hai-Qing Xu, Lei Gao, Yu-Bo Yuan, Kieran McLaughlin, and Sakir Sezer. 2016. Multidimensional intrusion detection system for IEC 61850-based SCADA networks. *IEEE Transactions on Power Delivery* 32, 2 (2016), 1068–1078.

[214] Tahreem Yaqoob, Haider Abbas, and Mohammed Atiquzzaman. 2019. Security vulnerabilities, attacks, countermeasures, and regulations of networked medical devices—A review. *IEEE Communications Surveys & Tutorials* 21, 4 (2019), 3723–3768.

[215] Hyunguk Yoo and Taeshik Shon. 2015. Novel approach for detecting network anomalies for substation automation based on IEC 61850. *Multimedia Tools and Applications* 74, 1 (2015), 303–318.

[216] Seunghyun Yoon, Jin-Hee Cho, Dong Seong Kim, Terrence J Moore, Frederica Free-Nelson, and Hyuk Lim. 2020. Attack graph-based moving target defense in software-defined networks. *IEEE Transactions on Network and Service Management* 17, 3 (2020), 1653–1668.

[217] Bruno Bogaz Zarpelão, Rodrigo Sanches Miani, Cláudio Toshio Kawakani, and Sean Carlisto de Alvarenga. 2017. A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications* 84 (2017), 25–37.

[218] Dianbo Zhang, Jianfei Wang, and Hua Zhang. 2015. Peach improvement on profinet-DCP for industrial control system vulnerability detection. In *2015 2nd International Conference on Electrical, Computer Engineering and Electronics*. Atlantis Press, Jinan, P.R. China, 1622–1627.

[219] Guijuan Zhang, Yang Liu, and Xiaoning Jin. 2020. A survey of autoencoder-based recommender systems. *Frontiers of Computer Science* 14, 2 (2020), 430–450.

[220] Kuan Zhang, Xiaohui Liang, Rongxing Lu, and Xuemin Shen. 2014. Sybil attacks and their defenses in the internet of things. *IEEE Internet of Things Journal* 1, 5 (2014), 372–383.

[221] Kuan Zhang, Jianbing Ni, Kan Yang, Xiaohui Liang, Ju Ren, and Xuemin Sherman Shen. 2017. Security and privacy in smart city applications: Challenges and solutions. *IEEE Communications Magazine* 55, 1 (2017), 122–129.

[222] Yichi Zhang, Lingfeng Wang, Weiqing Sun, Robert C Green, and Mansoor Alam. 2011. Artificial immune system based intrusion detection in a distributed hierarchical network architecture of smart grid. In *2011 IEEE Power and Energy Society General Meeting*. IEEE, Detroit, MI, USA, 1–8.

[223] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Salt Lake City, UT, USA, 8697–8710.