AI-Driven Solutions for Safeguarding IoT Environments: An Intrusion Detection and Prevention Study

by

Poulmanogo ILLY

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULFILLMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, MARCH 18, 2024

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

M. Georges Kaddoum, Thesis supervisor
Department of Electrical Engineering, École de technologie supérieure

M. François Coallier, President, Board of Examiners
Department of Software and IT Engineering, École de technologie supérieure

M. Kim Khoa Nguyen, member of the jury
Department of Electrical Engineering, École de technologie supérieure

M. Azzam Mourad, external examiner
Lebanese American University

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON FEBRUARY 20, 2024

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

This dissertation is written based on the author's Ph.D. research outcomes under the supervision of Professor Georges Kaddoum from January 2018 to September 2023. This work is financially supported by the Natural Sciences and Engineering Research Council of Canada under the B-CITI CRDPJ 501617-16 grant. The main theme of this dissertation focuses on intrusion detection and prevention systems for cyber-security in future Internet of Things (IoT) based smart city applications. This dissertation is written as a monograph based on three published IEEE journal papers as the first author.

In this dissertation, the first two chapters present the introduction and the literature review of IoT applications and intrusion detection and prevention systems. The following three chapters are written based on my research journal papers. Finally, the conclusion and the recommendation for future work are given in the last chapter.

ACKNOWLEDGEMENTS

I feel so blessed to have special people who have generously supported me and given me opportunities to grow in different ways in my life.

First and foremost, I would like to express my gratitude to my supervisor, Professor Georges Kaddoum. *Dear Professor, I consider the day I meet you as one of the best days of my life. Thank you for the opportunity you gave me, the trust, the constant support, motivation, encouragement, and patience throughout my Ph.D*. Professor Kaddoum always showed endless faith in my abilities and gave me good conditions for doing the research. Especially, he knew how to pull me to reach my best performance without putting too much pressure. I would never have been able to complete this thesis without his guidance.

I am also appreciative to Prof François Coallier, Prof Kim Khoa Nguyen, and Prof Azzam Mourad for their agreement to serve as my jury members, and for reviewing and giving many constructive comments for my dissertation.

My sincere thanks go to my fellow labmates in LACIME. I am grateful to all of those with whom I have worked, especially Pr. Kuljeet Kaur and Dr. Sahil Garg. Many thanks also go to my friends for their help and for encouraging me with high enthusiasm.

Special thanks to my loving and supportive wife, Dr. Rose Blandine Dao, for encouraging me to complete this thesis successfully, taking care of me, and keeping me away from depression. I could not have made it without you.

I would like to wholeheartedly thank my adoptive parents Poussi Etienne Ily and Victorine Doulkom; my uncles, cousins, brothers and their wives for their invaluable support and love. Finally, I thank my departed biological parents.

Solutions basées sur l'IA pour la protection des systèmes de l'IdO : Une étude sur la détection et la prévention d'intrusion

Poulmanogo ILLY

RÉSUMÉ

Les progrès réalisés ces dernières années dans le domaine des technologies de l'information et de la communication (TIC) ont donné naissances à des nouveaux concepts révolutionnaires dont l'un des plus récents et les plus importants est l'internet des objets (IdO). L'IdO, à travers des objets connectés bas prix, permet une collection abondante en temps réel de données et des automatisations intelligentes des systèmes d'information et des systèmes d'opérations. Les énormes opportunités d'innovation ouvertes par l'IdO ont entrainé son adoption fulgurante dans les entreprises privées et les institutions publiques de multiples domaines d'activités. Cependant l'impact des cyber-attaques est devenu plus inquiétant pour trois raisons principales : (1) la faiblesse des mécanismes de sécurité implémentés dans l'IdO, (2) la valeur des données qu'une cyber-attaque pourraient permettre d'accéder et (3) le niveau de control que l'attaquant pourrait désormais se donner dans ces systèmes qui sont totalement automatisés. Dans ce contexte, les systèmes de détection et de prévention d'intrusions (SDPI), en tant que couche indispensable dans la cyber-sécurité, sont devenus une des parties les plus actives des travaux de recherche pour la sécurisation de l'IdO. Les systèmes de détection d'intrusions sont capables d'analyser les activités en temps réel afin de distinguer les évènements, puis détecter et signaler les cyber-attaques pour permettre des réactions manuelles ou automatisées pour bloquer ou atténuer les menaces. Cependant, vu la mutation et le développement des attaques, il est indispensable de concevoir et implémenter de nouveaux systèmes de détection intelligents, précis, rapides et évolutifs. Pour répondre à ce besoin, l'apprentissage machine, notamment l'apprentissage par les réseaux de neurones profonds (RNPs), se révèle être une approche adéquate.

Dans cette thèse, trois objectifs principaux pour la conception des systèmes intelligents de détection d'intrusion sont considérés. Ces objectifs sont la détection d'une large variété d'attaques ciblant l'IdO, y compris les attaques de type "jour zéro", l'amélioration de la précision de détection et la minimisation du temps de latence de la détection et de la réponse. Pour atteindre ces objectifs, nous analysons les cyber-attaques qui ciblent les systèmes IdO et proposons diverses primitives qui peuvent être utilisées dans les algorithmes d'apprentissage machine pour détecter efficacement chacune de ces attaques. Ensuite, nous implémentons et comparons différents algorithmes d'apprentissage, y compris des méthodes d'apprentissage superficiel, des RNPs et des méthodes d'ensemble, afin de proposer des modèles qui améliorent la précision de la détection. De plus, nous concevons un schéma d'apprentissage collaboratif qui permet une détection à faible latence et une réponse pour contrer les attaques détectées.

Le chapitre 2 étudie principalement les comportements des différentes attaques ciblant l'IdO en considérant un scénario de maison intelligente, puis analyse la qualité des primitives qui peuvent être extraites et employées dans les algorithmes d'apprentissage machine pour détecter efficacement chacune de ces attaques. Nous proposons différentes primitives qui peuvent

améliorer les performances des systèmes de détection basés sur l'apprentissage machine. Plus précisément, nous proposons des primitives issues des en-têtes de paquets TCP/IP (protocole de contrôle de transmission/protocole Internet), des primitives basées sur la fréquence et le nombre de connexions et des primitives basées sur la partie donnée des paquets TCP/IP. En outre, pour détecter les attaques qui exploitent le canal de communication sans fil, d'autres primitives sont examinées, notamment la distance par rapport à l'émetteur radio, l'empreinte de la fréquence radio, l'intensité de signal reçu, le rapport signal/bruit et le profil énergétique du système.

Le chapitre 3 propose un SDPI multi-étages hybride basé sur les RNPs, offrant une précision améliorée pour des systèmes de supervision industriel (SSI) à très haut risque qui ne peuvent se permettre de faire des compromis sur la sécurité pour gagner en latence. Les modèles d'apprentissage sont entraînés de manière séquentielle avec divers algorithmes de RNPs, et chaque modèle dans la séquence se focalise sur les limites des modèles précédents. Le réseau de neurones multi-étages qui en résulte utilise la décision de chaque étage dans une fonction de combinaison pour produire une décision finale avec une précision améliorée.

Contrairement au chapitre 3 qui étudie un scénario SSI à haut risque où une sécurité renforcée est préférable malgré le coût en latence, le chapitre 4 étudie un scénario SSI à mission critique où la latence est une exigence cruciale. Avant tout, nous effectuons une analyse de la complexité en temps des RNPs pour mettre en évidence l'impact que les structures de ces modèles ont sur la latence à l'apprentissage et à la prédiction. Ensuite, nous concevons un SDPI collaboratif à faible latence et robuste basé sur l'apprentissage profond qui utilise deux modèles de classifications. Le premier modèle effectue une détection d'anomalie basée sur un RNP léger déployé dans des serveurs locaux pour permettre une détection rapide et une réponse urgente aux attaques. Le deuxième modèle effectue la classification du trafic anormal dans des serveurs infonuagiques pour guider les tâches complémentaires de prévention d'intrusions. De plus, une architecture de déploiement basée sur le concept SDN (réseau défini par logiciel) du SDPI collaboratif proposé est fournie pour les réseaux SSI.

Mots-clés: internet des objets, système de détection d'intrusion, système de prévention d'intrusion et apprentissage machine

AI-Driven Solutions for Safeguarding IoT Environments: An Intrusion Detection and Prevention Study

Poulmanogo ILLY

ABSTRACT

The progress in information and communication technologies (ICT) made in recent years has led to new revolutionary concepts where one of the most important ones is the Internet of Things (IoT). IoT, through low-cost connected objects, enables abundant and real-time data collection and smart automation of information and operation systems. The tremendous innovation opportunity opened by IoT has triggered its massive adoption in multiple business domains. Meanwhile, the impact of cyber-attacks has become more alarming for three main reasons: (1) critical weaknesses in IoT security mechanisms, (2) valuable data that attract cyber-attacks, and (3) the level of control that successful attacks could open in IoT-based automated systems. In this context, intrusion detection and prevention, which is essential in cyber-security, has become one of the most active research areas for securing IoT applications. Intrusion detection systems (IDSs) can analyze real-time activities to detect and report cyber-attacks to security administrators or automated intrusion prevention systems (IPSs) that initiate response measures to block the threats or attenuate their impact. However, given the changing and expanding nature of cyber-attacks, it is essential to design and implement new IDSs that are intelligent, accurate, fast, and scalable. In this vein, machine learning (ML), and particularly deep learning (DL), has emerged as a suitable approach to meet these requirements.

In this thesis, three main objectives essential for the design of an intelligent intrusion detection system are considered. These objectives are the detection of a wide range of IoT attacks, including zero-day attacks, the enhancement of the detection accuracy, and the minimization of the detection and response latency. To achieve these objectives, we analyze the cyber-attacks that target IoT systems and propose diverse features that can be used in ML algorithms to detect each of these attacks efficiently. Then, we implement and compare different learning algorithms, including shallow, deep, and ensemble learning methods, to propose models that enhance the detection accuracy. Furthermore, we design a collaborative learning scheme that enables low-latency detection and response to mitigate detected attacks.

Chapter 2 mainly studies the behaviors of different IoT attacks in a smart home scenario, and analyzes the quality of the features that can be extracted and employed in ML algorithms to detect each of these attacks efficiently. We propose various features that can improve the performance of ML-based IDSs. Specifically, transmission control protocol/internet protocol (TCP/IP) packet headers, time-based statistics, connection-based statistics, and TCP/IP packet content features are proposed. Furthermore, to detect attacks that exploit the wireless communication channel, more features are discussed, including the distance from the radio transmitter, radio-frequency fingerprint, received signal strength, signal-to-noise ratio, and the system's energy profile.

Chapter 3 proposes a hybrid multistage deep neural networks (DNNs)-based intrusion detection and prevention system (IDPS) with improved accuracy for critical industrial control systems

(ICSs) that cannot afford to compromise the security to improve latency. The learning models are trained sequentially with diverse algorithms, and each model in the sequence focuses on the limitations of the previous models. The resulting multistage DNN uses each stage's decision in a combination function to produce a final decision with improved accuracy.

In contrast to Chapter 3 which considers a high-risk ICS scenario where enforced security is preferable even at the cost of latency, chapter 4 considers a mission-critical ICS scenario where latency is also a crucial requirement. In this context, first and foremost, we conduct a time complexity analysis of DNNs to illustrate how the structures of these models impact the training and prediction latency. Then, we design a low latency and robust deep learning-based collaborative IDPS that employs two levels of classifications. The first level performs a lightweight DNN-based anomaly detection in local servers to allow faster attack detection and emergency response measures. The second level performs attack classification of the anomalous traffic in cloud servers to guide complementary intrusion prevention tasks. Moreover, an SDN-based deployment architecture of the proposed collaborative IDPS in ICS networks is provided.

Keywords: internet of things, intrusion detection system, intrusion prevention system, and machine learning

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF ALGORITHMS

Page

LIST OF ABREVIATIONS

3GPP	3rd generation partnership project
5G	fifth generation
6LOWPAN	internet protocol version 6 over low power wireless personal area networks
AI	artificial intelligence
ANN	artificial neural network
API	application programming interface
ARP	address resolution protocol
BLE	bluetooth low energy
CNN	convolutional neural network
CPU	central processing unit
CSV	comma-separated values
DCS	dynamic channel selection
DDoS	distributed denial of service
DELM	deep extreme learning machine
DL	deep learning
DNN	deep neural network
DNS	domain name system
DoS	denial of service
DT	decision tree

ÉTS	école de technologie supérieure
FAR	false alarm rate
FN	false negative
FP	false positive
GAN	generative adversarial network
GRU	gated recurrent unit
GSM	global system for mobile communications
HIDS	host-based intrusion detection system
HMI	human-machine interface
HTTP	hypertext transfer protocol
ICMP	internet control message protocol
ICSs	industrial control system
ICT	information and communication technology
IDPS	intrusion detection and prevention system
IDS	intrusion detection system
IDWG	intrusion detection working group
IEEE	institute of electrical and electronics engineers
IETF	internet engineering task force
IIoT	industrial internet of things
IoT	internet of things

IoV	internet of vehicles
IP	internet protocol
IPS	intrusion prevention system
IPv6	internet protocol version 6
IRS	intrusion response system
IT	information technology
KNN	k-nearest neighbors
LAN	local area network
LPWAN	low-power wide-area network
LSTM	long short-term memory
MAC	media access control
MitM	man in the middle
ML	machine learning
MLP	multi-layer perceptron
NB-IoT	narrow band internet of things
NFV	network functions virtualization
NIDS	network-based intrusion detection system
NIST	national institute of standards and technology
OS	operating system
owasp	open web application security project

PAN	personal area network
PCAP	packet capture
PER	packet error rate
PKI	public key infrastructure
PLC	programmable logic controller
PUF	physically uncloneable function
QoS	quality of service
R2L	remote to local
RBFN	radial basis function network
RF	random forest
RFF	radio frequency fingerprinting
RFID	radio-frequency identification
RNN	recurrent neural network
RSS	received signal strength
RTU	remote terminal unit
SCADA	supervisory control and data acquisition
SDN	software-defined networking
SNR	signal to noise ratio
SOM	self Organizing Map
SSH	secure shell

SSL	secure sockets layer
SVM	Support vector machine
SYN	synchronization
TCP	transmission control protocol
TCP/IP	transmission control protocol / internet protocol
TN	true negative
TP	true positives
TTL	time to live
U2R	user to root
UDP	user datagram protocol
UML	unified modeling language
UR	undetected rate
USA	United States of America
VM	virtual machine
VOS	voltage over-scaling
WAN	wide area network
WSN	wireless sensor network

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

bps	bit per second
Gbit	gigabit
Gbps	gigabit per second
GHz	gigahertz
Kbit	kilobit
Kbps	kilobit per second
km	kilometre
km/h	kilometres per hour
m	metre
Mbit	megabit
Mbps	megabit per second
mn	minutes
ms	milliseconds
s	seconds

INTRODUCTION

Motivation, challenges, and research objective

With the massive deployment of internet of things (IoT) devices, ensuring cyber-security has become one of the biggest challenges for future networks. IoT devices are generally designed with operability in mind without much consideration on cyber-security. As a result, these devices come with alarming vulnerabilities, including weak passwords, encryption, and authentication methods that cyber-criminals can leverage to perform diverse attacks. According to the 2020 Thales data threat report [Thales Group (2019)], in a survey of security executives with respondents from over 16 countries, 99% of them have IoT security concerns related to multiple challenges and 33% have concerns regarding attacks on IoT devices that may impact critical operations (Figure 0.1).

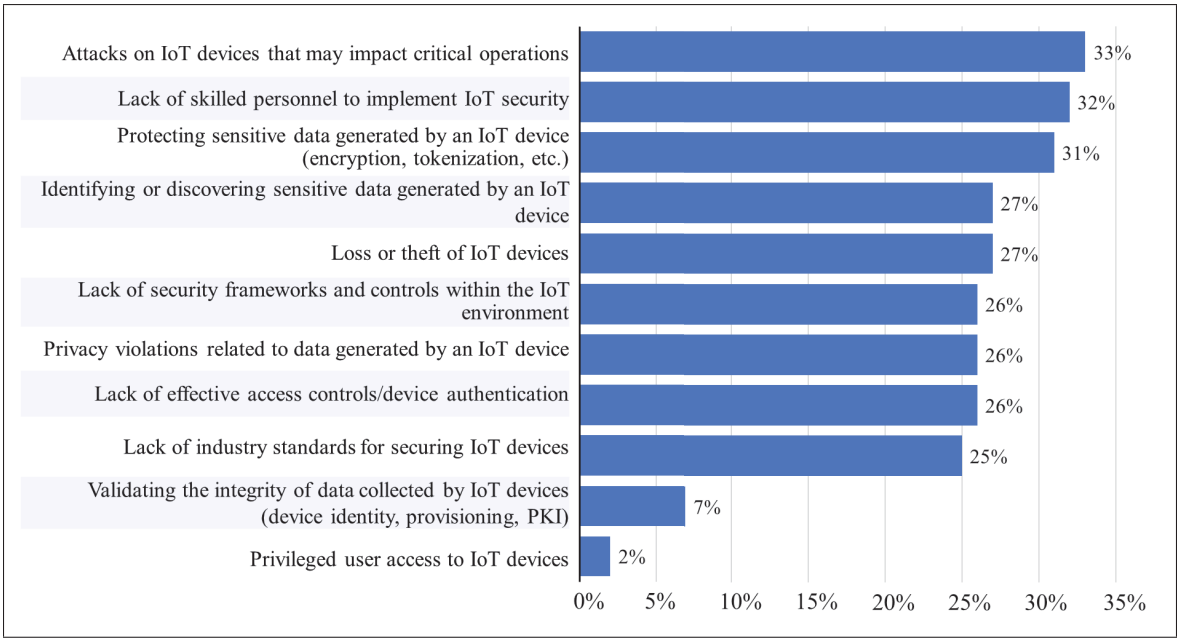

Figure 0.1 Global IoT cyber-security concerns
Taken from Thales Group (2019)

To address Information and communication technology (ICT) security concerns, over the years, complex encryption, authentication, authorization, and physical layer security mechanisms have been developed and have shown high efficiency. However, generally, these complex security mechanisms require heavy computation and communication load and are designed for traditional computer networks, where important computation resources are available. As a result, they cannot be directly deployed in IoT systems because of resource limitations in most IoT devices. These limitations include low computational power, small memory size, less radio bandwidth, and short battery lifetime [Haroon, Shah, Asim, Naeem, Kamran & Javaid (2016); Sinche, Raposo, Armando, Rodrigues, Boavida, Pereira & Silva (2020)]. Moreover, IoT introduces new significant security challenges, especially, the heterogeneity in terms of hardware, software, and protocols. For instance, in a network, a security measure that is appropriate for one IoT device may not be applicable to another. Another crucial aspect is that, given the enormous number of IoT devices, it is challenging to develop and deploy security mechanisms that can endure with the scale and range of devices. For instance, authentication and authorization through cryptographically pre-shared keys are not applicable. The rapidly growing number of objects will make key management become a problematic task [Chaudhary, Aujla, Kumar & Zeadally (2019); Zhang, Cho, Wang, Hsu, Chen & Shieh (2014)]. This implies that new security mechanisms that are adapted to IoT constraints and specifications are required.

In this context, new lightweight encryption, authentication, and authorization mechanisms designed explicitly for IoT are emerging [Deebak, Memon, Khowaja, Dev, Wang, Qureshi & Su (2022); Gunathilake, Al-Dubai & Buchana (2020); Hammi, Livolant, Bellot, Serhrouchni & Minet (2017); Zhang, Shen, Su, Arafin & Qu (2022)]. However, considering the national institute of standards and technology (NIST) cyber-security framework, most of these IoT security mechanisms act on the protection layer, *i.e.*, the first cyber-security layer, which aims to avoid the violation of systems security policy, such as confidentiality, integrity, availability, etc. This protection layer alone is not enough to ensure complete cyber-security, because

some attacks, especially zero-day attacks and attacks that employ advanced techniques, high computational resources or leaked security information can still compromise the systems [Mylrea, Gourisetti & Nicholls (2017); Norris, Mateczun & Forno (2022)]. Therefore, as a second security layer, intrusion detection and prevention systems (IDPSs) are required so that malicious attempts (successful or failed) to compromise the systems' security policy can be detected beforehand.

Intrusion detection systems (IDSs) can monitor real-time system events and employ a predefined database of attack signatures to detect intrusions [Li, Tug, Meng & Wang (2019); Mughal, Luo, Ullah, Ullah & Mahmood (2018); Sheikh, Rahman, Al-Qahtani, Kumar Hazra & Sheikh (2019)]. This method, named signature-based detection, has the advantage of providing IDSs that have low false alarm rates. However, signature-based detection is limited when it comes to detecting zero-day attacks or attacks that are not yet included in the signature database. IDSs can also employ predefined systems specifications, such as the system's functional states and transitions, operational and protocol constraints, and statistical rules to differentiate between normal and malicious events [Choudhary, Astillo, You, Yim, Chen & Cho (2020); Sharma, You, Yim, Chen & Cho (2019); Yun, Astillo, Lee, Kim, Kim & You (2021)]. This method, named specification-based detection, can detect known attacks as well as zero-day attacks. However, specification-based detection is limited to human expertise and manual systems modelization, which causes problems in terms of precision and scalability.

In the context of rapidly evolving attacks and constantly growing networks, the need for intelligent, scalable, and accurate IDPSs become a central research topic in IoT security. Taking the aforementioned challenges into account, this thesis aims to design an intelligent, accurate, and evolutive IDS that can learn to detect existing IoT attacks as well as zero-day attacks, with an automatic response system to stop or attenuate detected attacks. Regarding the growing number of attacks and their mutations, learning-based approaches are expected to play a critical role

in the security of large and heterogeneous IoT networks. More specifically, future IDPSs will require high levels of intelligence to ensure efficient and robust security. Therefore, we propose anomaly-based IDSs that use machine learning detection models. In particular, we focus on collaborative machine learning models that can provide high accuracy and low latency IDPSs to ensure security for future IoT applications.

Contributions and Outline

The organization of this dissertation, which includes four chapters, is structured and detailed as follows. In Chapter 1, a comprehensive literature review of IoT technology and security challenges is provided. In this vein, IoT architecture, security threats, and vulnerabilities are presented. Additionally, recent works on IoT security approaches, especially intrusion detection and prevention are provided. In particular, we focus on learning-based approaches for building robust IDSs.

Chapter 2 presents the first article, which proposes machine learning-based IDPSs enhancement with complementary detection features. While appropriate machine learning algorithms can improve IDSs accuracy, some of the limitations in the detection of various attacks are often caused by a lack of appropriate detection features. Therefore, considering a smart home scenario, we study the behaviors of different cyber-attacks and determine features that can be extracted and employed in machine learning algorithms to detect each of these attacks efficiently. This paper identified various features that can improve the detection accuracy. Specifically, in addition to packet header features, we have proposed time-based and connection-based traffic features against scanning, probing, and denial of service (DoS) attacks. Then, different content features have been proposed to enhance the detection of unauthorized user access, such as user to root (U2R) and remote to local (R2L) attacks. Furthermore, to detect attacks that exploit the wireless communication channel, such as jamming, de-authentication, spoofing, and contamination attacks, more features are discussed, including the distance from the radio

transmitter, radio-frequency fingerprint, received signal strength, signal to noise ratio, and the system's energy profile.

Chapter 3 presents the second article, which proposes a hybrid multistage deep neural network (DNN) based IDPS for high-risk IoT applications. Single deep learning-based models are largely implemented for IDSs but have shown limitations when learning increasingly complex intrusion patterns. Therefore, we propose a multistage combination of diverse DNNs to provide improved detection accuracy. The DNNs are trained sequentially in a way that allows new stage DNNs to focus on the limitations of the previous stages. Specifically, each new stage is trained using the data that were misclassified or classified with low confidence by the previous stage DNN. The resulting multistage DNN uses each stage's decision in a combination function to produce a final decision with improved accuracy.

Chapter 4 presents the third article, which proposes a collaborative DNN-based low-latency IDPS for mission-critical smart factory networks. Considering a scenario where latency is a crucial requirement, we conduct a time complexity analysis of neural networks to illustrate how the structures of these models impact the training and prediction latency. Then, we design a low latency and robust deep learning-based collaborative IDPS that employs two levels of classifications. The first level performs a lightweight DNN-based anomaly detection in local servers to allow faster attack detection and response. The second level performs the attack classifications of the anomalous traffic in cloud servers to guide the intrusion prevention tasks. In the proposed approach, emergency response measures can be launched as soon as an anomaly is detected. Then, when the anomaly is classified in a specific attack type, suitable complementary response measures can be applied to mitigate the threat efficiently. Furthermore, an SDN-based deployment architecture of the proposed collaborative IDPS in industrial control system (ICS) networks is provided.

CHAPTER 1

LITERATURE REVIEW

1.1 Internet of Things (IoT) Applications

The IoT paradigm is impacting almost every business domain and is expected to play a crucial role in future information and operation technologies. The IoT provides simple and effective means to collect and analyze all kinds of data to provide information and enable real-time resource visibility, predictive and prescriptive insights, improve operational efficiency, reduce costs, etc. Organizations incorporate IoT technology to enable entirely new business models or radically enhance existing models [Maiti & Ghosh (2021); Ploennigs, Cohn & Stanford-Clark (2018); Salih, Rashid, Radovanovic & Bacanin (2022)]. As a result, they are all facing IoT cyber-security threats. This section describes four main IoT applications, *i.e*, smart home, smart factory, smart transportation, and smart city.

1.1.1 Definition of IoT

The IoT as a concept has been created in 1999. The term IoT was initially proposed to refer to uniquely identifiable interoperable connected objects with radio-frequency identification (RFID) technology [Ashton et al. (2009); Gershenfeld (1999)]. However, the term became popular in 2010/2011 and reached the mass market in early 2014. Today, different definitions have been proposed by many worldwide technical standards coordination and development organizations. For the international telecommunication union (ITU), IoT can be defined as *"a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, inter-operable information and communication technologies"*. According to the IEEE IoT journal, *"an IoT system is a network of networks where, typically, a massive number of objects/things/sensors/devices are connected through communications and information infrastructure to provide value-added services via intelligent data processing and management for different applications"*. However, from a basic

perspective, IoT can be defined as sensors and actuators embedded in physical objects and linked through wired and wireless computer networks like the internet. The term object here includes whatever comes to mind, ranging from wearables to home appliances to industrial equipment. These interconnected objects enable a wide range of innovative applications, including environmental monitoring, workplace and home support, inventory and production management, healthcare services, security, and surveillance.

IoT applications have been dramatically evolving in recent years. According to the survey released by Transforma Insights in May 2022, the number of connected IoT devices worldwide, estimated at 13 billion in 2022, is forecasted to double to more than 29 billion in 2030 [Transforma Insights (2022)]. Figure 1.1 presents the estimated number of connected IoT devices worldwide from 2019 to 2021, with forecasts to 2030.

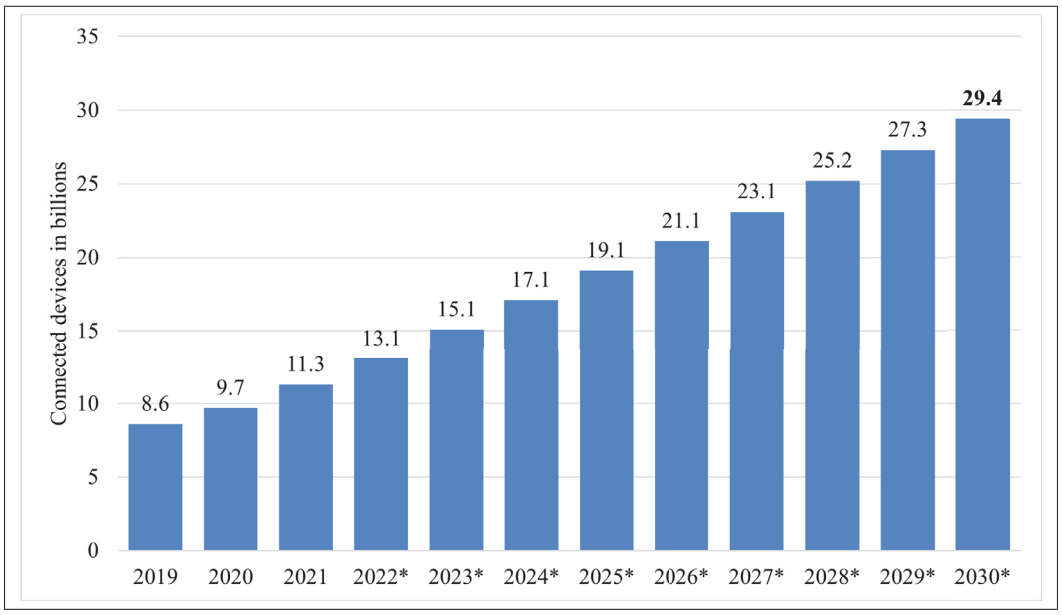

Figure 1.1 Number of connected IoT devices worldwide 2019-2021 with forecasts to 2030
Taken from Transforma Insights (2022)

1.1.2 IoT Application Domains

The growth of the IoT technology has created immense opportunities to build innovative solutions in many domains. In most applications, the solutions are designed to monitor and capture data in real-time from the sensing devices and send it through wired or wireless networks to the processing devices, such as local servers or data centers, for analytics to drive smart automation systems, *i.e.*, systems that are more reactive, correlative, flexible, and communicative. These approaches have created new concepts, such as smart home, smart healthcare, smart factory, smart agriculture, smart transportation, and smart city. Some of these IoT applications are briefly explained in the coming paragraphs.

1.1.2.1 Smart Home

A large number of heterogeneous IoT devices are manufactured for homes. These devices are deployed to upgrade traditional homes and build smart homes which enable advanced monitoring, automation, and remote control. Most smart home applications, for example, offer the residents features to remotely monitor and control their lights, windows, doors, heaters, air conditioners, etc., from their smartphones, tablets and computers [Aartech (2019)]. Figure 1.2 illustrates diverse use cases in smart homes. Such applications present many benefits, including more security and safety, optimized energy consumption, and maximized comfort.

An example of a smart home project is HOWZ, a smart system from a United Kingdom Start-Up that allows users to remotely monitor senior citizens' use of electrical devices (kitchen, kettle, coffee machine, toaster, room light) via sensors in their home [Howz (2019)]. It is a way of checking that everything is normal, that their home has the ideal temperature, and that their door is closed. Howz provides monitoring to help elderly people retain their independence while also reassuring their families. With a £200 initial price for a starter kit, it is an innovation designed to make smart homes available to all. Howz was the winner of the Électricité de France (EDF) Pulse Awards 2017 in the smart home category in France.

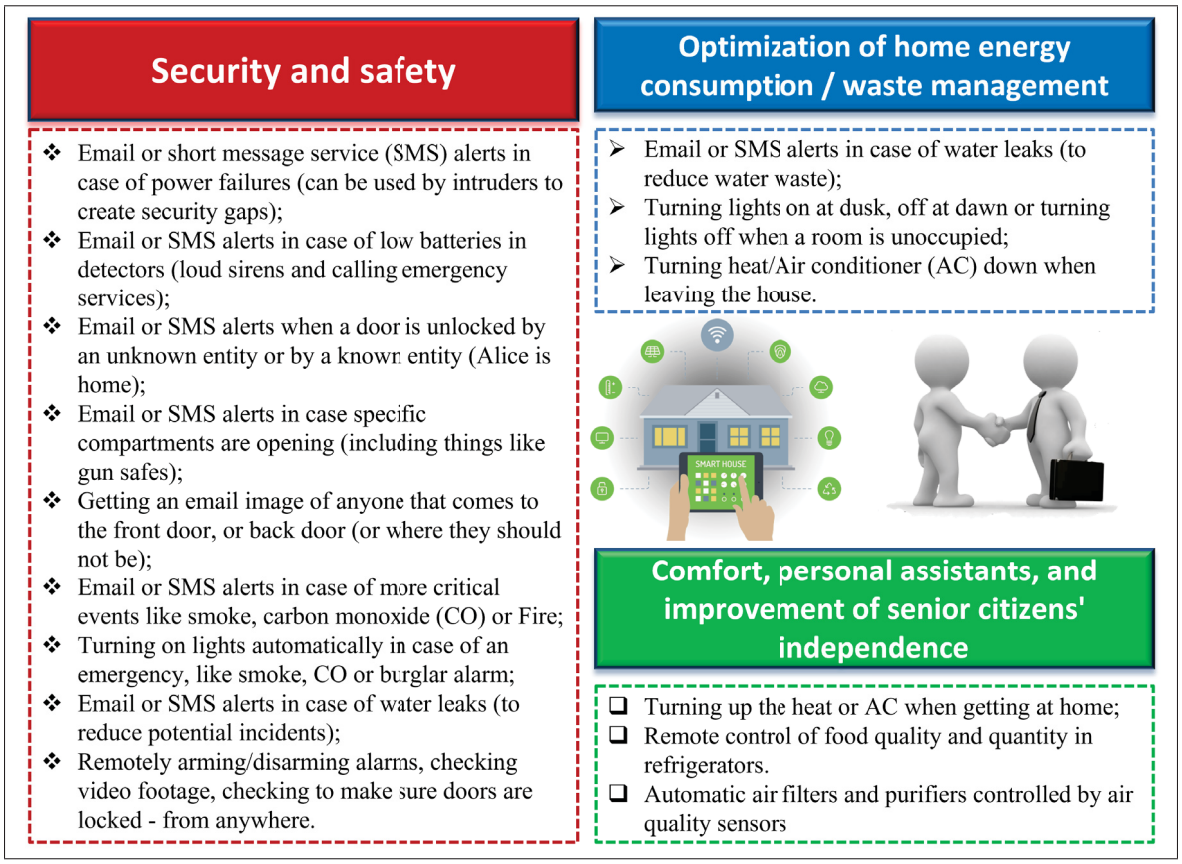

Figure 1.2 Example of smart home use cases

Inside the École de technologie supérieure (ÉTS), a project called Star ÉTS (Sustainable Smart ÉTS Residence) has been initiated as a living lab to support research on smart home applications and more generally, smart city applications. The Smart ÉTS Residence project uses the ÉTS residences to test prototypes of systems in the development or test phase that can be implemented in a smart home. Sensors of all kinds are placed in the apartments and data are collected to develop resource (electricity, water) optimization systems, to control appliances, such as smart heating, and to control the lights (for example the apartment can detect that there is nobody left and turn off the lights that have been left on). The Smart ÉTS Residence project was submitted to Canada's first open-air laboratory for smart living (LabVI), created by Videotron in collaboration with Ericsson, ÉTS, and Montréal's Quartier de l'Innovation [ÉTS (2019)].

1.1.2.2 Smart Factory

In the manufacturing sector, industrial IoT (IIoT) is one of the key emerging technologies that drive the fourth industrial revolution (Industry 4.0). In this major technological breakthrough, traditional manufacturing industries are being upgraded to smart manufacturing industries which provide solutions for handling the increasing production complexity through the establishment of intelligent products and production processes [Abate *et al.* (2022); Wan, sXia, Hong, Pang, Jayaraman & Shen (2019)]. Figure 1.3 depicts a typical ecosystem of a smart factory. IIoT devices are integrated into the manufacturing processes to enable smart factory applications that gather and analyze the manufacturing data for real-time equipment monitoring, service and quality control, flexible production, dynamic reconfiguration, manufacturing optimization, production scheduling, and adaptation to the changes of business models and consumer shopping behaviors.

Smart factories will play a crucial role in the economic growth of developed and developing countries. It is one of the top trending innovation areas that industrialized countries are paying close attention to. Important projects, programs, and strategies have been launched worldwide, such as Europe 2020 strategy [European Commission (2010)], Industry 4.0 strategy [Lasi, Fettke, Kemper, Feld & Hoffmann (2014)], China manufacturing 2025 [Zhou (2015)], and the United States (U.S.) national strategic plan [EOP (2012)]. In Canada, an example of initiative is the refined manufacturing acceleration process (REMAP), a program *"designed for future-focused leaders who want to leverage a competitive advantage and resiliency through digital technologies and smart manufacturing tools"* [REMAP (2022)]. REMAP is supported by the Government of Canada's Business-led Networks of Centres of Excellence program. REMAP is home to Canada's first advanced manufacturing network. They have built a supply chain of more than 70 of Canada's most sophisticated labs and manufacturing lines, in 11 cities across Canada, and aim to accelerate the growth and create new business opportunities in the adoption of smart manufacturing.

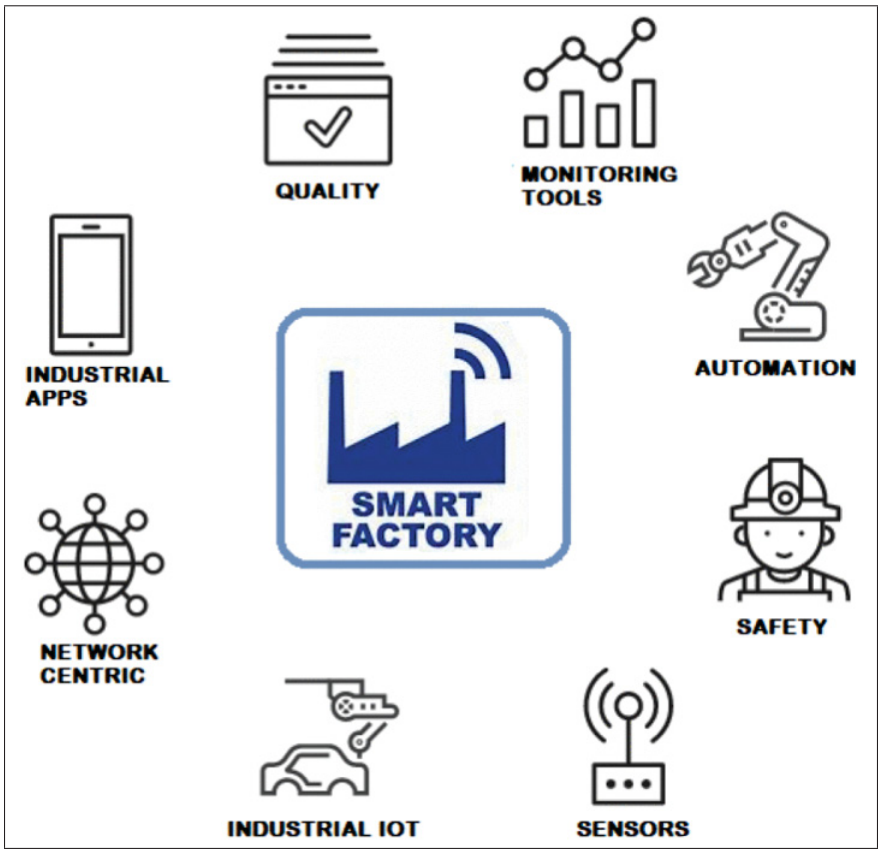

Figure 1.3 Ecosystem of a smart factory
Taken from Abate *et al.* (2022)

1.1.2.3 Smart Transportation

The transportation industry is one of the largest segments that are investing in IoT. Thanks to smart gadgets, many applications are possible, including smart fleet, public transit, and inventory management.

The smart fleet management systems are slowly and gradually being adopted, enabling improved operational efficiency, maintenance cost, fuel consumption, regulatory compliance, and accident response time. Global positioning system (GPS) tracking, geo-fencing, customized dashboards, and real-time business decisions are some of the key features fleet management offers [Verma (2018)]. The smart public transit systems offer many benefits to passengers. With real-time tracking of vehicles, the IoT has eradicated many challenges that were facing public transit

systems. It has enabled re-routing features to help people make alternate arrangements as real-time tracking of the vehicle is easily done. The smart inventory management provides real-time information across warehouse, distribution, and production centers, which reduces the cost of inventory management and improves predictive maintenance. Smart inventory management systems have also reduced inventory errors.

1.1.2.4 Smart City

The smart city vision aims to exploit the most advanced communication technologies to support added-value services for the administration of cities and citizens. This objective is pursued by the deployment of an urban IoT, *i.e.*, a communication infrastructure that provides unified, simple, and economical access to a plethora of public services, thus unleashing potential synergies and increasing transparency to the citizens. Urban IoT brings a number of benefits in the management and optimization of traditional public services, such as transport and parking, lighting, surveillance and maintenance of public areas, preservation of cultural heritage, garbage collection, and salubrity of public places (hospitals, schools, etc.). Furthermore, different types of data, collected by a pervasive urban IoT, may also be exploited to increase transparency and promote the actions of the local governments toward the citizens, enhance people's awareness about the status of their city, stimulate the active participation of the citizens in the management of public administrations, and also stimulate the creation of new services upon those provided by the IoT [Cuff, Hansen & Kang (2008); Zanella, Bui, Castellani, Vangelista & Zorzi (2014)].

An example of a smart city project is the European-wide smart cities project named bIoTope (Building an IoT open innovation ecosystem for connected smart objects) [Foundation Biotope (2017)], illustrated in Figure1.4. In Canada, Star ÉTS is also an example of a smart city project [ÉTS (2019)]. By transforming ÉTS apartments into smart houses, this project uses the residences to create a smaller-scale smart city. This is used to test prototypes that can be implemented in a smart city. An example of application is the intelligent dishwasher system that looks at greenhouse gas emissions to suggest the best time to perform the task.

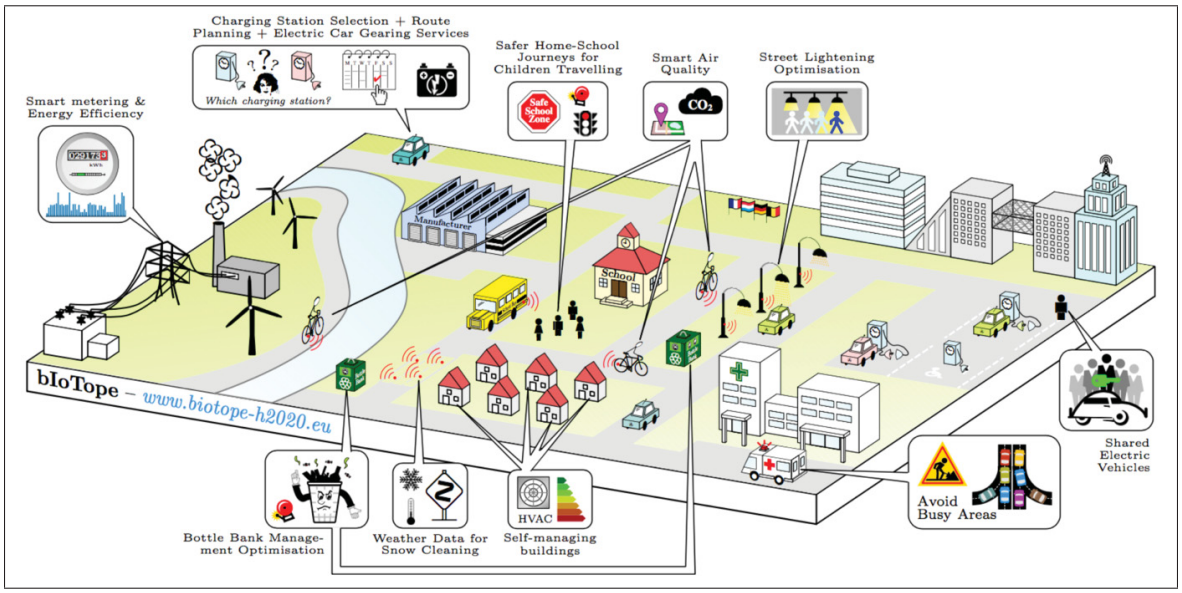

Figure 1.4 bIoTope: Building an IoT OPen innovation Ecosystem for connected smart objects
Taken from Foundation Biotope (2017)

1.2 IoT Architectures

A thorough analysis of IoT architectures not only clarifies the functional and technical requirements and constraints but also can provide an effective understanding of the attack surface, *i.e.*, the attack vectors and all possible points where an unauthorized user can access the systems' operations and data. Therefore, this section describes generic IoT architecture and emerging IoT architectures based on key future technologies, such as software-defined networking (SDN), and cloud and edge computing. For each architecture, the layers involved, and technologies used are presented. Then, potential benefits, and drawbacks are discussed.

1.2.1 Basic Three-layer IoT Architecture

IoT architectures are widely researched, and different implementation models have been extensively discussed [Atzori, Iera, Morabito & Nitti (2012); Leo, Battisti, Carli & Neri (2014); Mahmoud, Yousuf, Aloul & Zualkernan (2015b); Wan, Yang, Wang & Hua (2018); Xiaocong, Jiao & Shaohong (2015)]. Still, there is no universal implementation standard. As a result, a

number of architectures have been proposed, each designed to meet the technical and functional requirements of specific application domains. However, most propositions are based on common patterns. Figure 1.5 presents the basic architectural framework. This architecture consists of three key layers, namely the smart device layer, the network layer, and the application layer.

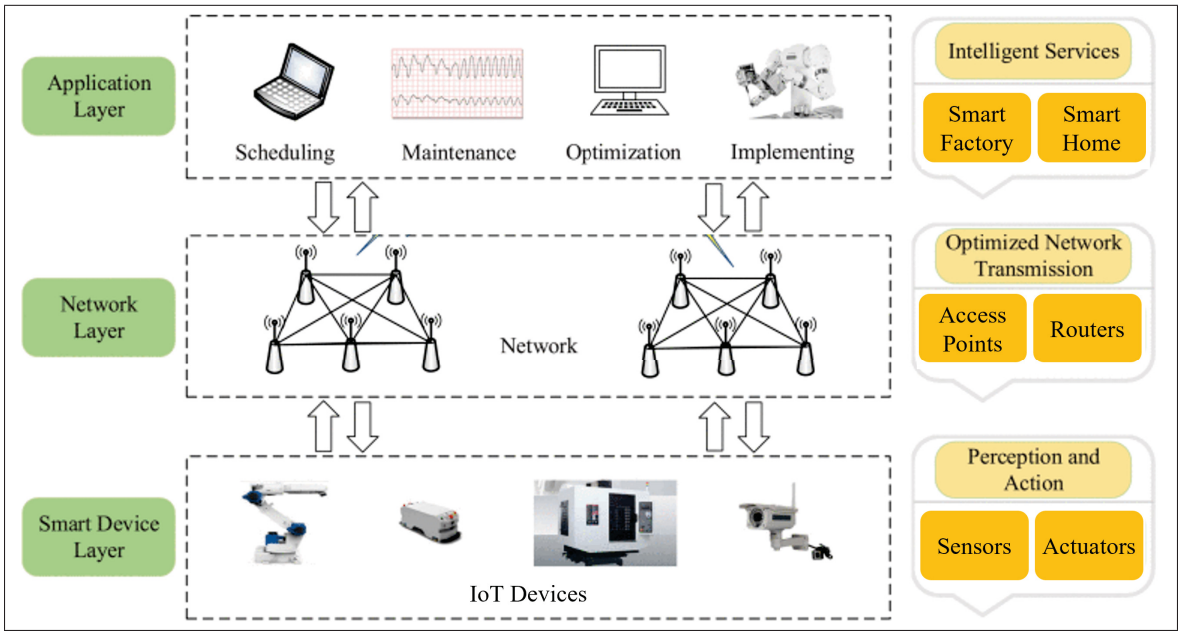

Figure 1.5 Three-layer IoT architecture

The smart device layer, also known as the sensors or perception layer, consists of IoT devices (*e.g.*, IIoT devices, home IoT devices, smart city devices, etc.). The function of this layer is to acquire data from the physical environment using the sensing devices. Then, the collected data can be processed locally or transmitted using the network layer. IoT devices can also collaborate in this layer by sharing data directly with each other using short-range communication technologies. Furthermore, control commands sent to the actuation devices are executed in this layer.

The network layer is the mediator between the smart device layer and the application layer. It serves the function of data transmission as well as the interconnection of systems and platforms. It also serves as the mediator between different IoT nodes by aggregating, filtering, and transmitting data to and from different IoT devices. This stage mostly relies on wireless communication technologies. It includes switching and routing devices, internet gateways, and cloud computing

platforms. The ecosystem of this layer benefits from important innovations driven by emerging technologies, such as network function virtualization (NFV), SDN, and network slicing.

The application layer is the top level and represents the final stage of the architecture. This layer provides the functions of storing, organizing, processing, and sharing the environment data and other information obtained from the IoT devices. This is the stage where the purpose of IoT is achieved, *i.e.*, the implementation of smart applications, such as smart homes, smart factories, smart transportation, smart cities, etc. As the monitoring tasks are becoming more and more complex and more data is coming from IoT devices, applications require increasing storage and computation resources. Efficient technologies and models for processing the data are required. Therefore, new technologies, such as cloud and fog computing are massively integrated at this stage for extensible data storage and processing. Moreover, to efficiently exploit the massive scales of data and improve smart systems, IoT applications make use of big data and artificial intelligence (AI) technologies [Misra, Dixit, Al-Mallahi, Bhullar, Upadhyay & Martynenko (2022); Mohammadi, Al-Fuqaha, Sorour & Guizani (2018); Zhu, Ota & Dong (2022)].

1.2.2 IoT Wireless Communication Protocols

Smart devices can have wired connectivity but they mostly use wireless communication for flexibility, mobility, and cost-effectiveness. Since wireless networks do not require physical media (wires or cables), the devices can communicate even when they are moving. Moreover, wireless communication provides low-cost accessibility and scalability as the network can be easily extended, even to places where wires are not accessible. Furthermore, improvements in wireless communication technologies enable faster information transfer.

Different wireless communication technologies and protocols are used to connect smart devices in personal area networks (PAN), local area networks (LAN), and wide area networks (WAN). Each application employs the protocol that best suits its functional and technical requirements, such as the communication range and data transmission rate. The protocol selection also

considers limitations of the IoT devices, including energy constraints, processing capability, and storage volume.

New wireless communication protocols, specially designed to meet IoT specifications and power constraints, have emerged. These protocols include low-power personal area network protocols and low-power wide-area network protocols [Al-Sarawi, Anbar, Alieyan & Alzubaidi (2017); Ballerini, Polonelli, Brunelli, Magno & Benini (2020); Iqbal & Lee (2019); Kim, Lim, Jeong, Choi & Koh (2022)]. Figure 1.6 presents the most common IoT communication protocols, categorized according to the communication range, data rate, and power consumption [Jiang *et al.* (2021); Wang (2021)]. Some of these main IoT communication protocols are briefly explained next.

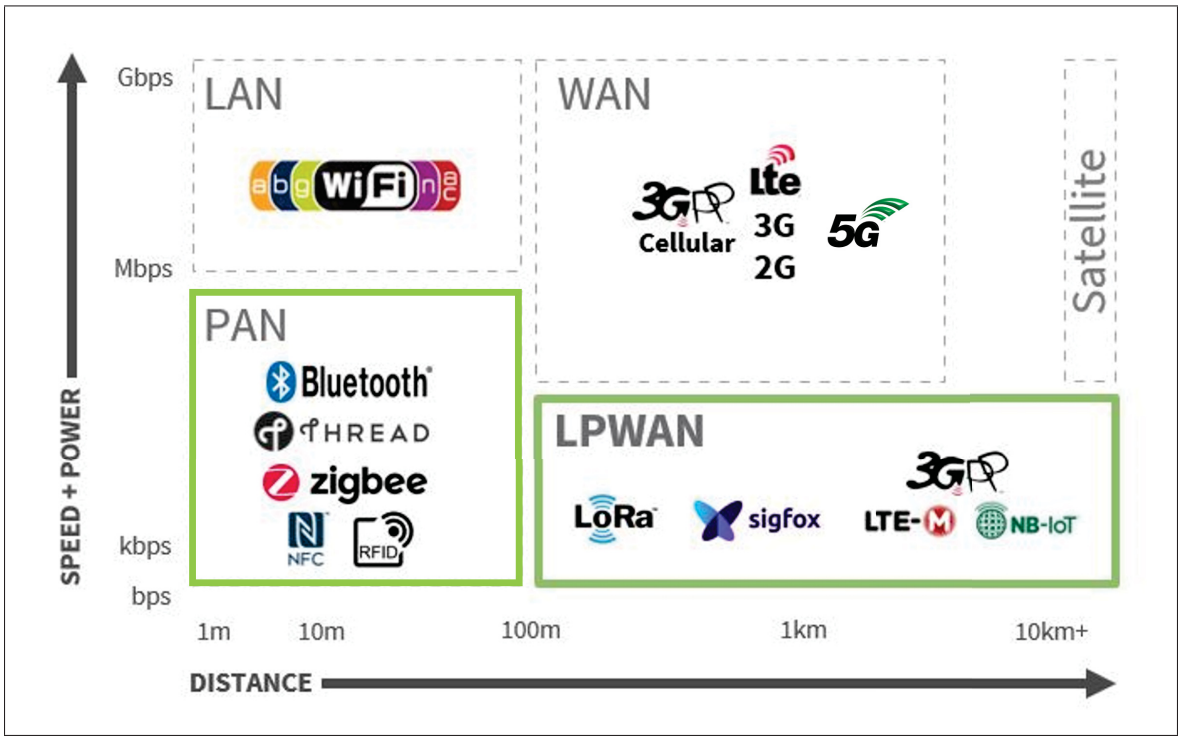

Figure 1.6 Most common IoT communication protocols
Taken from Jiang *et al.* (2021)

1.2.2.1 Low-Power Personal Area Networks

The most used wireless communication protocols to connect IoT devices in short-range networks include Bluetooth low energy (BLE), ZigBee, Z-Wave, and internet protocol version 6 (IPv6) over low power wireless personal area networks (6LoWPAN) [Al-Sarawi *et al.* (2017)].

BLE, also known as Bluetooth smart, is an important protocol for IoT applications. It is optimized for short-range, low-bandwidth, and low-latency IoT applications. The benefits of BLE include lower power consumption, fast setup, and support for star network topology with an unlimited number of nodes [Salman (2015); Samie, Bauer & Henkel (2016)].

The ZigBee protocol was developed by the ZigBee Alliance and is based on the IEEE802.15.4 low-power wireless network standard. ZigBee was created to be a standard to suit high-level low-cost communication protocols creating personal area networks from small-size, low-power digital radios that transmit data over long distances. It is used in applications requiring a low data rate, long battery life, and security. Furthermore, ZigBee can support a variety of topologies, such as the mesh, star, and tree network topologies [Salman (2015); Samie *et al.* (2016)].

Z-Wave is a Zensys-developed low-power medium access control (MAC) protocol used in wireless home automation to connect 30-50 nodes and has been used for IoT communication, particularly in the smart home and small commercial domains. It is designed for small data packets at relatively low speeds of up to 100 kilobits per second (kbps) and point-to-point communication over a distance of 30 metres (m). As a result, it is appropriate for small messages in IoT applications, such as light, energy, and healthcare controls. Z-Wave is supported by two types of devices (controller and worker). Worker nodes are low-cost devices that cannot initiate messages. They can only respond to and execute commands sent by controlling devices that initiate network messages. Z-Wave supports the mesh network topology [Barker & Parsons (2022); Salman (2015)].

6LoWPAN can connect IoT devices directly to internet protocol (IP) networks without the use of intermediate network components, such as translation gateways or proxies. The internet

engineering task force (IETF) developed this technology as a standard IP communication over low-power and low-cost wireless IEEE802.15.4 networks using IPv6. 6loWPAN supports a variety of topologies, including the mesh and star topologies. To enable interoperability between IEEE 802.15.4 and IPv6, 6loWPAN proposes an adaptation layer between the MAC and network layers [Gomes, Salgado, Pinto, Cabral & Tavares (2018); Kim *et al.* (2022); Palma (2018); Yang & Chang (2019)].

1.2.2.2 Low-Power Wide-Area Networks

Cellular and low-power wide-area network (LPWAN) technologies, such as SigFox, LoRa, and narrow-band IoT (NB-IoT) are the most commonly used wireless communication protocols for connecting IoT devices in wide-area networks [Al-Sarawi *et al.* (2017); Iqbal, Abdullah & Shabnam (2020)].

Cellular technologies are an effective solution for IoT applications that require high throughput transmission and long-distance communication. The use of cellular technologies allows IoT applications to benefit from the global system for mobile communications (GSM), third-generation (3G), fourth-generation (4G), and fifth-generation (5G) cellular communication capabilities, which provide reliable high-speed internet connectivity. In this context, 5G is generating great enthusiasm for IoT applications. 5G differs from its predecessors through several technical advances that provide high throughput (20 Gbps), low latency (< 1 ms), high mobility (500 km/h), high connection density (1 million connections/km^2), and low power consumption (90% reduction) [Nokia (2014)]. The summary of the key requirements for 5G is illustrated in Figure 1.7. 5G can meet these requirements thanks to a combination of different complementary technologies, including new radio frequencies, small base stations comprising femtocells, picocells, millimeter wave (mm-wave) technologies, and multiple-input-multiple-output (MIMO) antennas with the beamformation technology, SDN, and NFV. All these key technology drivers in 5G have a direct positive impact on the implementation of IoT-based smart applications [Chettri & Bera (2020); Hong, Jiang, Yu, Zhou, Chen, Yu, Zhang, Yang, Pang, Jiang et al. (2017); Shafique, Khawaja, Sabir, Qazi & Mustaqim (2020)]. However,

cellular communication protocols may be unsuitable for battery-constrained IoT devices [Ejaz, Anpalagan, Imran, Jo, Naeem, Qaisar & Wang (2016); Samie *et al.* (2016)].

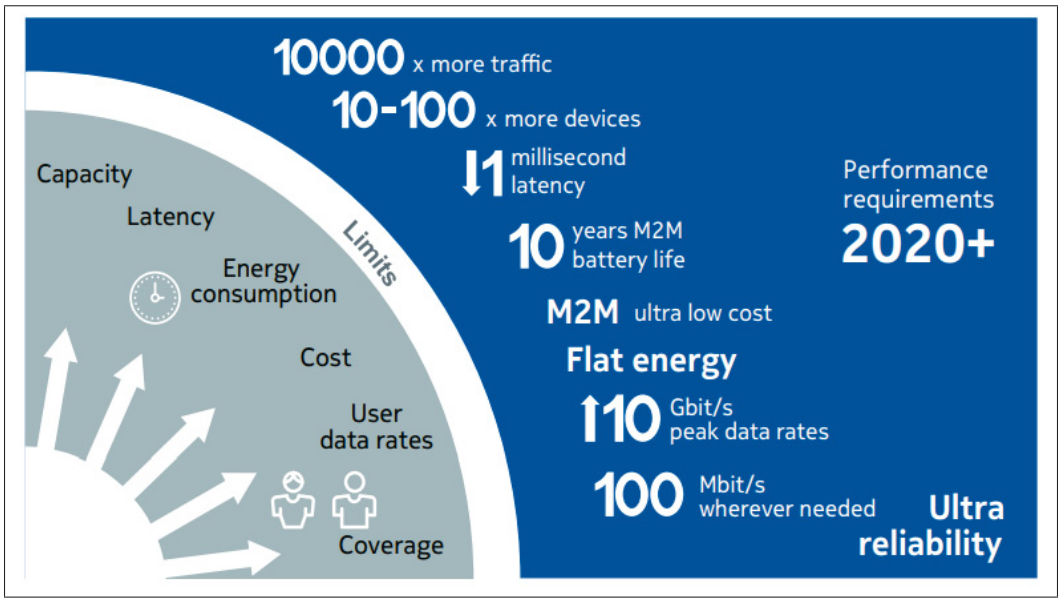

Figure 1.7 The summary of key requirements for 5G
Taken from Nokia (2014)

LoRa (acronym from "long-range") is the most widely used low-power wide-area technology. It adjusts and modulates signals by utilizing an exclusive spread spectrum technique in the sub-gigahertz (GHz) industrial, scientific, and medical (ISM) band [Vangelista (2017)]. LoRa technology operates in the unlicensed ISM band (868 MHz in Europe, 915 MHz in North America, and 433 MHz in Asia). It uses the chirp spread spectrum modulation where the output signals have a lower noise level, enabling high interference salience and making it difficult to identify or detect [Reynders, Meert & Pollin (2016)].

Similar to LoRa technology, Sigfox employs unlicensed ISM channels and effectively uses the frequency bandwidth with very low noise, low power consumption, high receiver sensitivity, and low-cost antenna design through the use of binary shift keying (BPSK) modulation. Therefore, this technology is suitable for low-energy devices. It enables the transmission of small amounts of data over distances of up to 50 km. SigFox uses ultra-narrow band (UNB) technology, only intended for low data transfer rates of 10 to 1000 bits per second (bps) and can be powered

by a small battery [Krupka, Vojtech & Neruda (2016); Mekki, Bajic, Chaxel & Meyer (2018); Osman & Abbas (2018); Rubio-Aparicio, Cerdan-Cartagena, Suardiaz-Muro & Ybarra-Moreno (2019)].

NB-IoT, one of the leading LPWAN technologies standardized by the third generation partnership project (3GPP), operates seamlessly in authorized frequencies on the current GSM and long-term evolution (LTE) networks. However, compared to LoRa and Sigfox, which provide efficient power consumption and long-term battery life, NB-IoT provides a higher quality of service (QoS), latency, reliability, and coverage [Beyene, Jantti, Tirkkonen, Ruttik, Iraji, Larmo, Tirronen, Torsner & Johan (2017); Mekki, Bajic, Chaxel & Meyer (2019)].

1.2.3 Software-Defined Networking (SDN)

The exponential growth of IoT creates new management challenges that traditional networking architectures are unable to address. In this context, SDN has emerged as a promising technology that can efficiently handle large-scale and complex IoT infrastructures due to its flexible management and programmability.

1.2.3.1 SDN Concept

SDN decouples the control plane from the data plane of the forwarding devices and logically centralizes the network intelligence and state in new devices called controllers. In general, the SDN's communication infrastructure operates according to standards designed by the open networking foundation (ONF). The controller connects to the switches through an OpenFlow channel and manages the switches via the OpenFlow protocol [Flauzac, González, Hachani & Nolot (2015); Liyanage, Abro, Ylianttila & Gurtov (2016); Mahmood, Chilwan, Østerbø & Jarschel (2015); Valdivieso Caraguay, Benito Peral, Barona Lopez & Garcia Villalba (2014)]. SDN has three key attributes that directly benefit IoT infrastructures :

1. **Logically centralized intelligence:** a controller with a global view that can manage the entire network;

2. **Programmability:** the ability to use advanced software programming techniques to modify network behavior and functions;

3. **Abstraction:** the ability to hide complex network infrastructure and protocols behind the network operating system (OS); business applications can abstract underlying network information with the help of SDN.

1.2.3.2 SDN-based IoT Architectures

The SDN-based architecture is divided into three planes, namely the data plane, the control plane, and application plane [Farris *et al.* (2019)], as illustrated in Figure 1.8.

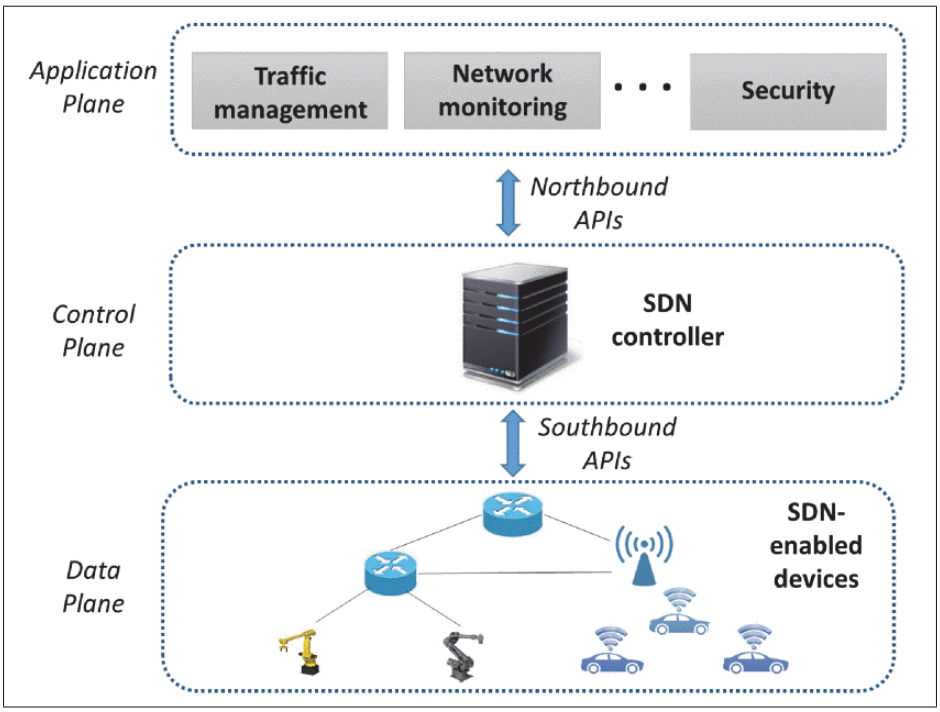

Figure 1.8 The three layers in SDN architectures
Taken from Farris *et al.* (2019)

The data plane is composed of the forwarding devices, such as physical and virtual switches, routers, and gateways [Jain & Paul (2013)]. These forwarding devices operate according to the controller's forwarding decisions in the SDN control plane. These decisions are organized

into a flow table in the delivery device using a data control plane interface. The flow tables operate according to the instructions added to the instruction set available on the controller [Aujla, Chaudhary, Kumar, Rodrigues & Vinel (2017)].

The control plane is the core of the SDN architecture and serves as the decision-making plane. This plane operates according to the centralized control logic provided to the controller. This plane's primary functions are to install control commands on forwarding devices, manage and maintain global information for all SDN applications running at the application plane, and collect feedback from the forwarding devices. Therefore, the SDN controller provides an abstract model of the underlying network for the SDN application layer. Moreover, the controller can use the network OS to create a virtual controller using a hypervisor. One of the key features of SDN is that the control logic can be programmed and reconfigured according to the environment [Li, Dong & Ota (2016)].

In the application plane, with the help of network virtualization, the controller can create multiple virtual networks on the physical network. The virtual machines (VMs) can run multiple SDN applications at the same time. This is an effective solution for managing large amounts of data. Virtualization also enables isolation between users, resource sharing, and physical resource aggregations into a unified virtual resource [Mahmood *et al.* (2015)]. SDN applications are software programs that run to manage the resources and networks efficiently. Using the interface between the application and the control planes, the control logic generated by the SDN controller handles internal decisions directly and maintains an abstract view of the network.

There is no universal SDN implementation standard for IoT systems. Each IoT application introduces specific requirements and optimization in its SDN adoption [Bera, Misra & Vasilakos (2017)]. As a result, a number of implementations are proposed at different levels, such as data center, core, and access network, as illustrated in Figure 1.9, thus covering the IoT traffic management from the devices, which generate the data, up to the cloud services, where data processing is performed [Farris *et al.* (2019)].

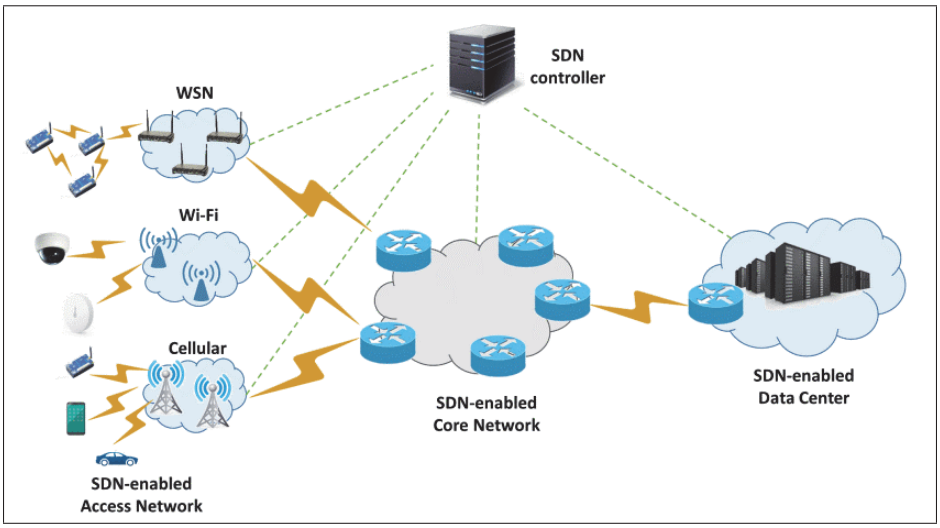

Figure 1.9 Deployment scenario of SDN paradigm for IoT systems
Taken from Farris *et al.* (2019)

1.2.4 Cloud Computing and Edge Computing

The high volume, heterogeneity, and velocity of IoT-generated data lead to important demands in data storage and processing resources. Regarding the cost of these resources (acquisition, deployment, energy, space, management, and maintenance), it is too expensive or economically inefficient for most businesses to locally store and process all their data. This has created a strong need to integrate cloud computing into IoT architectures.

1.2.4.1 Cloud Computing-based IoT Architectures

Cloud computing provides on-demand self-service (automatic provisioning of computing capabilities), broad network access (capabilities are available over a networked infrastructure), resource pooling (resources are pooled together to serve multiple consumers using a multi-tenant model), rapid elasticity (rapid and elastic provisioning of capabilities to quickly scale up or down as required), and measured service (automatic control and optimization of resources utilizing a pay-per-use model) [Mell, Grance et al. (2011)]. Different integration architectures of cloud and IoT have been proposed to obtain benefits in specific application scenarios [Aitken, Chandra,

Myers, Sandhu, Shifren & Yeric (2014); Alhakbani, Hassan, Hossain & Alnuem (2014); Botta, De Donato, Persico & Pescapé (2016); Gomes, Righi & da Costa (2014); Sarkar *et al.* (2018)]. Figure 1.10 illustrates a generic cloud computing-based IoT architecture [Eclipse IoT Working Group (2016)].

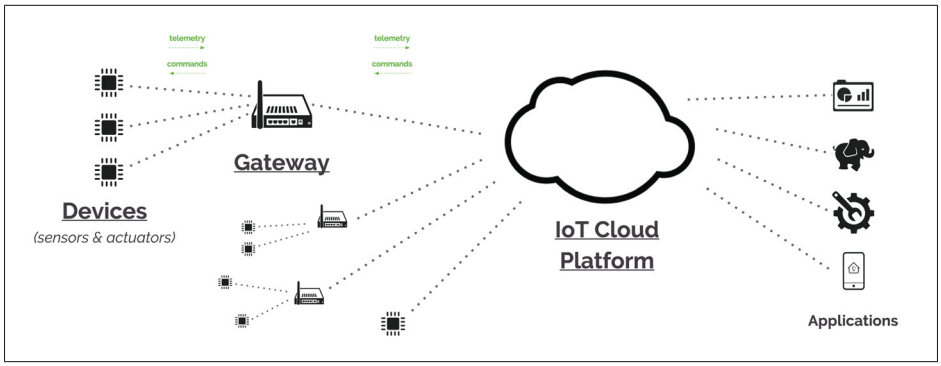

Figure 1.10 Cloud computing-based IoT architecture
Taken from Eclipse IoT Working Group (2016)

A wide variety of IoT applications are latency sensitive and require quasi-real-time data processing [Wei (2014)]. However, IoT architectures that rely only on cloud computing suffer from higher latency and bandwidth cost due to processing servers' location being generally far from the endpoints. This has created a need to build adaptive and decentralized computational architectures that complement the centralized cloud computing model. As a result, edge computing has been introduced to bridge these gaps and enable ultra-low latency applications.

1.2.4.2 Edge Computing-based IoT Architectures

Edge computing is a distributed computing paradigm that selectively moves some functionalities of the cloud (*e.g.*, computation, control, and decision-making) to the vicinity of end-users [Baek (2022); Kaur, Dhand, Kumar & Zeadally (2017); Mouradian, Naboulsi, Yangui, Glitho, Morrow & Polakos (2018)]. The most popular edge computing technologies include fog computing, multi-access edge computing (MEC), and cloudlet. Figure 1.11 illustrates a fog computing architecture in the context of IoT. This approach enables low latency and location awareness, wide-spread geographical distribution, mobility, support for a large number of nodes,

strong presence of streaming and real-time applications, and heterogeneity (wide variety of environments) [Bonomi, Milito, Zhu & Addepalli (2012)]. These characteristics make edge computing an appropriate architecture for a number of mission-critical IoT applications, such as connected vehicles, smart healthcare, smart grid, and smart cities.

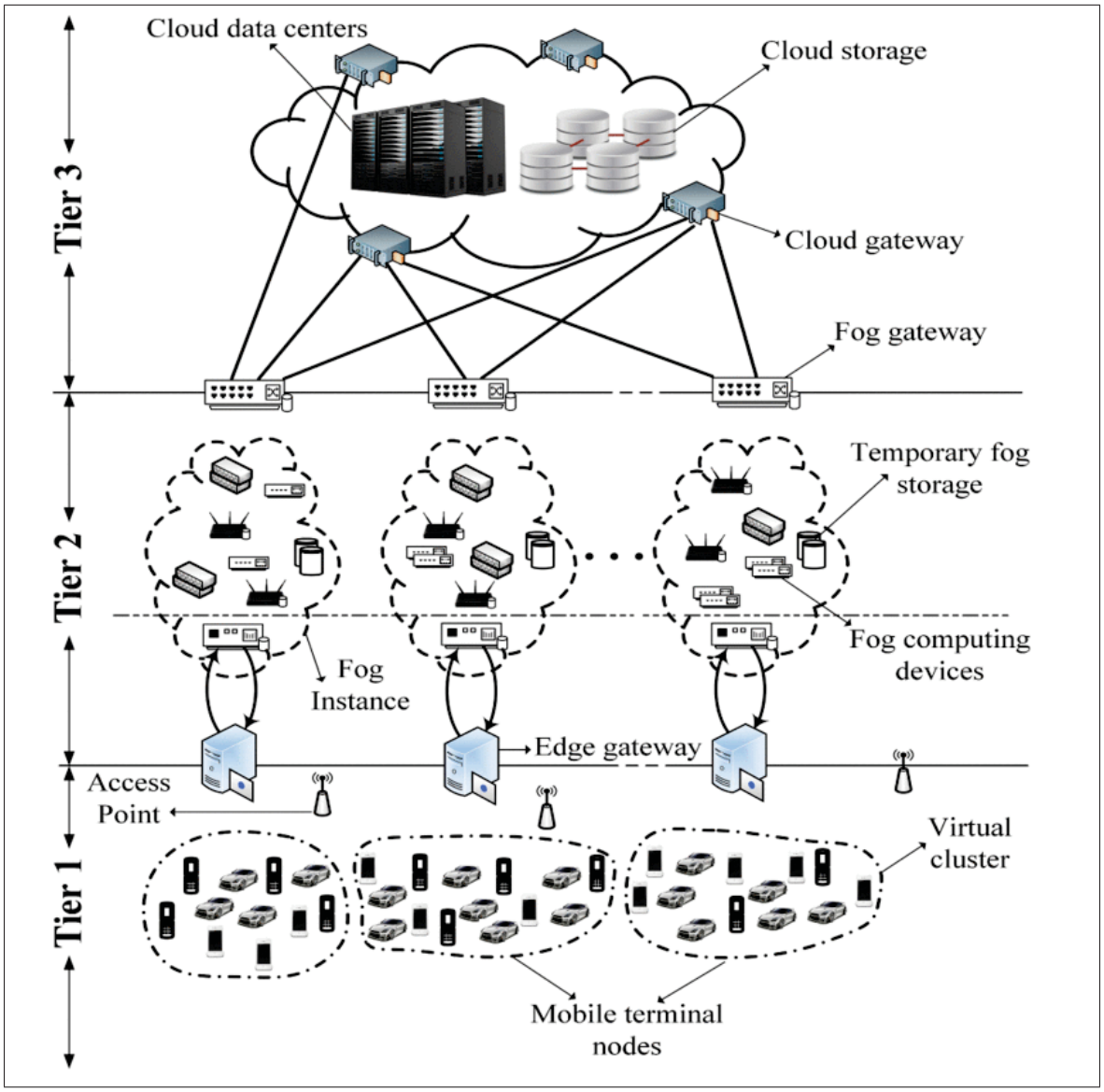

Figure 1.11 Cloud and edge computing-based IoT architecture
Taken from Sarkar *et al.* (2018)

1.3 IoT Security

The main objectives of IoT security are to preserve systems' privacy, confidentiality, integrity, and authenticity, ensure users' and devices' security, and guarantee the availability of the services. These objectives have many challenges and are the subject of many research works. Firstly, this section discusses critical IoT security requirements and reviews major IoT cyber-threats and vulnerabilities. Secondly, we review IoT protection layer security solutions presented in recent works.

1.3.1 IoT Security Requirements

A security requirement is a statement of needed security functionality that ensures one of many different security properties is being satisfied. Security requirements are generally derived from standards, history of past vulnerabilities, and applicable laws to solve a specific security problem and prevent the repetition of past security failures. IoT security requirements include the three fundamental information security attributes, namely confidentiality, integrity, and availability. However, IoT security can also include specific requirements that can often be complex, interwoven, and formulated on different abstraction levels [Mahmoud, Yousuf, Aloul & Zualkernan (2015a); Oh & Kim (2017); Strielkina, Kharchenko & Uzun (2018); Varshney & Gupta (2017)]. The main IoT security requirements presented in the literature are defined as follows:

- **Confidentiality:** This important attribute ensures that the data is secure and only available to authorized users. These users can be humans, machines, services, internal objects (devices that are part of the network), and external objects (devices that are not part of the network). For example, it can be crucial to make sure that sensors don't reveal the collected data to neighboring nodes [Strielkina *et al.* (2018)]. The confidentiality policy defines as well how data must be managed. For example, it is important for IoT users to be informed of the data management mechanisms, the process or the person responsible for the management, and the protection measures implemented to ensure data privacy.

- **Integrity:** This attribute ensures that the data are correct and trustworthy. It involves maintaining the accuracy, trustworthiness and consistency of information over the complete life-cycle of the data. In the transit of information, data should not be altered, and measures should be taken to make sure that the information is not breached by unauthorized participants.

- **Availability:** This requirement ensures that the data is accessible to authorized users whenever they need it. Devices and services must also be reachable and available when needed in a timely fashion in order to satisfy IoT applications' technical and functional requirements.

- **Identification and authentication:** Each object in the IoT system must be able to clearly identify and authenticate other objects. However, this process can be very challenging because of the nature of IoT systems; many entities are involved (devices, people, services, service providers and, processing units) and sometimes objects may need to interact with other objects for the first time (objects they do not know). Therefore, mechanisms to mutually authenticate entities in every interaction in the IoT are needed.

- **Access control:** This requirement ensures that each user, device, and platform can only access specific resources according to predefined rules. These rules can be a fundamental part of IoT security as they dictate who is allowed to access and use IoT resources. By combining identification, authentication, and authorization, access control policies make sure entities are who they say they are and that they have access to the appropriate resources.

- **Accountability:** This requirement ensures the traceability of actions performed by a user, process, or device. It also ensures that any malicious act performed by an intruder can be detected. Accountability is crucial for trust, as it relates to the responsibilities, incentives, and means for recourse regarding those building, deploying, managing, and using IoT systems and services [Singh, Millard, Reed, Cobbe & Crowcroft (2018)].

- **Non-repudiation:** This requirement ensures that an entity cannot declare to not be the owner of its generated information or deny the actions made by it. With this feature, it is easy to trace the actions performed by any communicating party [Alsamani & Lahza (2018)].

- **Policies:** There must be policies and standards to ensure that data is managed, protected, and transmitted in an efficient way. More importantly, a mechanism to enforce such policies is needed to ensure that every entity is applying the standards. Service level agreements (SLAs) must be clearly identified in every service involved. Current policies that are used in computer network security may not be applicable to IoT due to its heterogeneous and dynamic nature. The enforcement of such policies will reinforce trust by human users in the IoT paradigm which will eventually result in its growth and scalability.

IoT security requirements can vary according to the application domain [Chaudhary (2018)]. For example, mission-critical IoT applications require high availability and very low latency to support applications like remote healthcare and other 5G use cases. On the other hand, in most massive IoT use cases, some applications are tolerant in terms of network latency and availability (*e.g.*, remote health monitoring).

1.3.2 Major IoT Threats and Vulnerabilities

IoT security is one of the major current information and communications technology (ICT) security challenges. The abundance of valuable data produced by the IoT attracts not only legitimate economic and governmental actors, but also alarming numbers of malicious users. The targeted data might contain industrial, economic, financial information (as in smart enterprises), or very sensitive personal information (as in smart homes or smart healthcare systems). For example, according to a recent study by Cynerio, a healthcare cyber-security provider, 56% of hospitals have had their IoT devices attacked in the past two years [Sadhu, Yanambaka & Abdelgawad (2022)]. Secondly, IoT devices generally suffer from significant security vulnerabilities and are seen by hackers as easy entry points into more important systems. As a result, the integration of IoT into traditional systems, such as homes, factories, healthcare, and transportation systems increases their exposure to attacks. For example, according to the recent study by the healthcare cyber-security provider Cynerio, 53% of medical IoT devices have at least one critical vulnerability, and 88% of data breaches involved IoT devices. In what follows, some of the main IoT attacks and vulnerabilities are discussed.

1.3.2.1 IoT Security Threats

Cyber-attacks against IoT systems can target diverse elements in the infrastructure, including IoT devices, communication channels, and applications. The most popular attacks include brute-force attacks, firmware hijacking, eavesdropping, spoofing, malicious node injection, man-in-the-middle (MitM), physical tampering, jamming, denial of service (DoS), malware attacks, and privacy leakage [Andrea, Chrysostomou & Hadjichristofi (2015a); Chen, Zhang, Li, Zhang, Deng, Ray & Jin (2018); Deogirikar & Vidhate (2017)]. These attacks are generally interrelated in their operation modes or in their objectives. For example, some attacks are launched specifically to open up access to more important attacks. In addition, there is a constant evolution in the level of sophistication of attacks, which increasingly use advanced technologies and benefit from more resources in terms of computing power.

In the state of the art, IoT attacks are generally classified according to various criteria. Figure 1.12 proposes a classification into physical, network, software and encryption attacks [Deogirikar & Vidhate (2017)]. Moreover, Table 1.1 provides a classification of diverse attacks based on the transmission control protocol/internet protocol (TCP/IP) communication protocol stack [Nawir *et al.* (2016)]. In what follows, some of these main IoT attacks are briefly explained.

Brute-force attacks: In these attacks, unauthorized users try to log into legitimate users' accounts using guessed passwords, and continue trying different passwords until they get the correct one. Usually, the attackers have automated software that generates different combinations of passwords. Regardless of the computational burden on the attacker, brute-force attacks were shown to be one of the major threats to network security in general, and IoT in particular [Najafabadi, Khoshgoftaar, Kemp, Seliya & Zuech (2014); Salamatian, Huleihel, Beirami, Cohen & Médard (2019)]. Generally, the solution to this threat is to set up a system that detects and prevents too many queries from a user, as determined by IP addresses. As a result, an attacker who uses a single IP address would be limited to a predefined number of guesses. However, this defence is circumvented by using massive botnets, each bot querying potential passwords. In this situation, it is more challenging to prevent brute-force attacks.

Firmware hijacking: These attacks take place when IoT devices download firmware updates. The attackers hijack the device to download malicious software from illegitimate sources. With so many IoT devices, brands, and products, firmware hijacking is a major concern. Moreover, most hardware makers don't cryptographically sign their firmware.

Eavesdropping attacks: These attacks are also known as sniffing or snooping attacks. Eavesdropping is typically done by listening to digital or analog voice communications or via the interception of sniffed data. The malicious users intercept the traffic between two end devices or between an IoT device and a server. If non-encrypted or weakly encrypted communications are found, the malicious users access confidential information, which can be used for more important attacks.

Spoofing attacks: A spoofing node impersonates a legal IoT device with its identities, such as the MAC address and RFID tag, to gain illegal access to the IoT system. This attack is used to access confidential information and can lead to MitM attacks [Xiao, Li, Han, Liu & Zhuang (2016)].

Malicious nodes injection and MitM attacks: In these attacks, the hackers breach the communications by injecting malicious nodes between the legitimate nodes or by targeting the communication protocols in IoT networks. Through the MitM concept, the hackers can alter the traffic flow, reconfigure the network topology, create fake identities, and generate malicious and false information to compromise the IoT system. The variants of MitM attack are Sybil, Wormhole, identity replication, and node replication attacks [Andrea, Chrysostomou & Hadjichristofi (2015b)].

DoS attacks: DoS attackers flood a targeted server with superfluous requests to prevent IoT users from obtaining services [Andrea *et al.* (2015b); Xiao, Wan, Lu, Zhang & Wu (2018)]. One of the most challenging types of DoS attacks is distributed denial of service (DDoS) attacks, where attackers use thousands of IP addresses to request IoT services, making it difficult for the server to distinguish the legitimate IoT devices from attackers [Zhang & Jetter (2016)].

Physical tampering: IoT devices usually suffer from hardware-level vulnerabilities [Hernandez, Arias, Buentello & Jin (2014); Wurm, Hoang, Arias, Sadeghi & Jin (2016); Zhao (2017)]. Malicious users that can obtain physical access to these devices can disassemble the hardware and perform different types of physical attacks. Physical attacks include different types of probing (passive/active injectors or pico-probes), machining methods (manual material removal, mechanical, water, laser, chemical, or shaped charge), and electrical (radiation, temperature, or high voltage imprinting, power supply fluctuations, clock glitching, circuit disruption, or electron beam and infrared laser read/write) [Grand (2004); Weingart (2000)]. With these methods, the attackers can modify the memory/computation, unveil employed cryptographic schemes, replicate firmware using malicious nodes, or simply corrupt the data.

Jamming attacks: These attacks consist in sending fake radio signals to interrupt the ongoing radio transmissions of IoT devices and further deplete their bandwidth, energy, central processing units (CPUs), and memory through failed communication attempts [Han, Xiao & Poor (2017)].

Malware attacks: The security problem for the IoT is sometimes aggravated by vulnerabilities produced by careless program designs, which create opportunities for malware or backdoor installation. Mobile malware, such as Trojans, worms, and viruses can result in IoT systems privacy leakage, power depletion, and network performance degradation [Xiao, Li, Huang & Du (2017)].

Privacy leakage: IoT systems have to protect user privacy during data caching and exchange. Some caching owners are curious about the data content stored on their devices and analyze and sell such private information. Wearable devices that collect users' personal information, such as location and health information, witness an increased risk of privacy leakage [Han *et al.* (2017)].

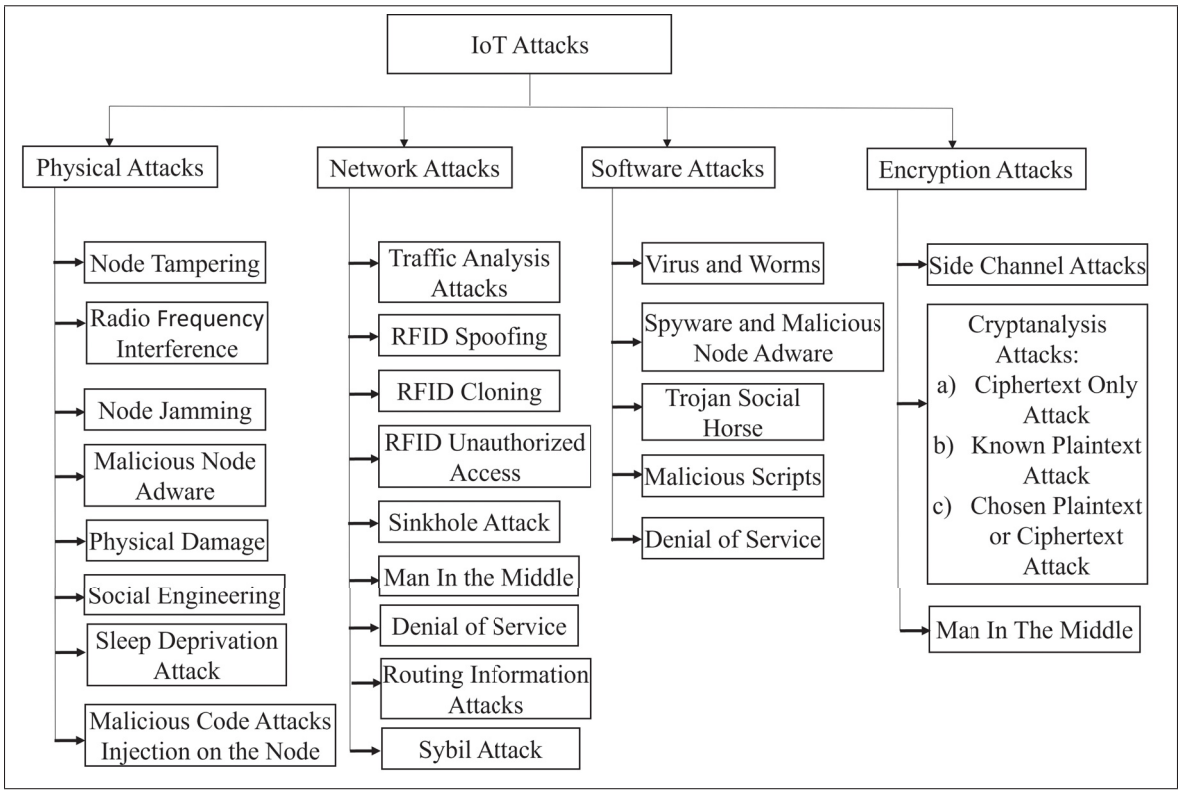

Figure 1.12 IoT attacks
Adapted from Sarkar *et al.* (2018)

1.3.2.2 IoT Security Vulnerabilities

Most IoT devices and applications have been designed with operationality and flexibility in mind instead of security. This has resulted in a significant number of vulnerabilities in every layer of IoT architectures. In general, IoT vulnerabilities come from a lack of cyber-security awareness, resource constraints, wireless communication channels, systems heterogeneity, and vulnerabilities related to IoT enabling technologies, such as cloud computing and SDN [Heer, Garcia-Morchon, Hummen, Keoh, Kumar & Wehrle (2011); Neshenko, Bou-Harb, Crichigno, Kaddoum & Ghani (2019)].

Lack of cyber-security awareness: An increasing number of industries are providing or adopting IoT technologies without putting enough effort on raising cyber-threat awareness. This is resulting in a lot of security malpractice that escalates the risks. An example of such

Table 1.1 Layer-based attacks with their attack strategies in IoT systems
Taken from Nawir *et al.* (2016)

Layer	**Attacks**	**Methods/Strategies attacks**
Physical	Jamming	Creates radio interference and exhaustion on IoT devices.
	Tampering	Creates compromised nodes.
Data Link	Collision	Simultaneous transmissions from two nodes on the same frequency.
	Exhaustion	By repetitive collisions.
	Unfairness	Using above link layer attacks.
Network	Spoofed, altered or replayed routing information	Creates routing loops, extend or shorten sources routes, attracting or repelling network from selected nodes.
	Selective forwarding	Choose what information to discard or gather before transmission.
	Sinkhole	Monitoring, Redundancy, Authentication
	Sybil	Single node duplicates its node to be in multiple locations.
	Wormholes	Selectively tunneling or retransmiting information to the IoT devices.
	HELLO flood	Uses HELLO packets as weapon to launch the attack on IoT systems.
	Acknowledgement spoofing	Spoof the link layer acknowledgments for overhead packets.
Transport	Flooding	Repeat the request of a new connection until the IoT system reaches maximum level.
	De-synchronization	Disruption of an existing connection.
Application	Attacks on reliability and Clone attack: - Clock skewing, - Selective message forwarding, - Data aggregation distortion	The adversaries usually masquerade like normal users in the IoT system, but they can launch malicious activities.

malpractice is the use of default and weak passwords. An enormous number of deployed IoT devices in conjunction with their cloud management solutions do not force a password of sufficient complexity [Hewlett Packard (2015)]. Moreover, after installation, numerous devices

do not request to change the default user credentials. Such malpractice on both the provider and consumer sides increases the success rate of brute-force attacks. In addition, countless firmware are released with known vulnerabilities, such as backdoors, no secure socket layer (SSL) usage, and users with elevated permissions [Elmiligi, Gebali & El-Kharashi (2016); Tekeoglu & Tosun (2016)]. For example, most of the users have root access, which can allow adversaries who gain unauthorized access to the device to perform user to root (U2R) attacks and take control of the entire system [Markowsky & Markowsky (2015); Neshenko *et al.* (2019); Siboni, Shabtai, Tippenhauer, Lee & Elovici (2016)]. Furthermore, among the millions of organizations that are deploying smart devices, very few organizations have deployed proper security measures.

IoT resource constraints: Unlike traditional computer networks that have considerable resources, IoT is mainly composed of resource-constrained devices due to their small or limited size. These devices have some limitations in terms of computational power, memory size, radio bandwidth, and energy access. These constrained devices do not provide enough resources to deploy complex security mechanisms that generate heavy computation and communication load. Therefore, the majority of IoT devices don't have the capabilities to incorporate traditional antiviruses, firewalls, or encryption and authentication mechanisms. For example, many IoT applications cannot afford traditional key generation and authentication techniques based on classical cryptography as it requires expensive secret key storage and high-complexity cryptographic algorithms. Under such circumstances, attackers can perform successful intrusions, access confidential information, append spoofed malicious nodes, and violate data integrity [Furfaro, Argento, Parise & Piccolo (2017); Vidgren, Haataja, Patiño-Andres, Ramírez-Sanchis & Toivanen (2013)]. Moreover, these resource-constrained devices can be easily overloaded. For example, for devices with no energy access, DoS attackers might drain the stored energy by generating a flood of legitimate or corrupted messages [Trappe, Howard & Moore (2015); Vasserman & Hopper (2013)].

Physical exposure: A considerable number of IoT devices are usually deployed (or move autonomously) in external, outdoor, non-restricted, and non-secure environments. With little effort, attackers can obtain unauthorized physical access to such devices and execute physical tampering attacks. For example, IoT devices installed on smart cars can be accessed from

the outside because there is no control over who can touch them in an open environment. So, attackers can target these systems.

Wireless communication vulnerabilities: Wireless communication is an essential component for IoT infrastructures, used for data collection from the perception layer (smart device layer) and control message delivery. However, due to the open nature of the wireless medium, data exchange may suffer from various attacks, including eavesdropping, DoS by de-authentication attacks or jamming the used frequency band, unauthorized user access through the wireless gateways, identity spoofing of IoT nodes or wireless access points, MITM attacks, and message falsification/injection. Wireless attacks are significantly dynamic in nature, usually hard to predict, and relatively simple to carry out [Krejčí, Hujňák & Švepeš (2017); Mpitziopoulos, Gavalas, Konstantopoulos & Pantziou (2009); Zou, Zhu, Wang & Hanzo (2016)]. For example, jamming equipment for different frequencies can easily be acquired by malicious users.

IoT heterogeneity and scale: IoT is about billions of heterogeneous devices, in terms of hardware and software, connected through heterogeneous wireless communications protocols and network topologies. This makes the networks and applications management extremely complex [Bedhief, Kassar & Aguili (2016); Restuccia, D'Oro & Melodia (2018)]. Firstly, in an infrastructure built with components from diverse companies with different approaches, standards, and use cases, it is not easy to define who will be responsible for the global security orchestration. Secondly, with devices that have different resource constraints and applications with different quality-of-service requirements, a security solution that is appropriate for some IoT devices and applications may not be applicable to others. Moreover, due to the heterogeneity and complexity of the objects and networks, traditional authentication and authorization methods may not be applicable. For instance, authentication and authorization through cryptographically pre-shared keys is not applicable. The rapidly growing number of objects will greatly complicate the key management task [Zhang *et al.* (2014)]. Furthermore, it may be infeasible to issue a certificate to IoT objects since the total number of objects is often huge. Given the number of these objects, it is also challenging to develop and deploy security mechanisms that can endure with the scale and range of devices.

Vulnerabilities in IoT enabling technologies: Emerging technologies, such as 5G wireless communication and its fundamental building blocks, such as SDN, NFV, and cloud computing enable new security solutions. However, there are several new threats that arise as a result of their implementation [Dutta & Hammad (2020)].

In SDN, critical security vulnerabilities exist in different modules, such as controllers, VMs, and OpenFlow switches. For example, the SDN control plane is vulnerable to attacks like host location hijacking, link fabrication, port amnesia, and port probing [Rawat & Reddy (2017)]. In the data plane, attackers can use fraudulent flow rules, flooding attacks, traffic diversion, and address resolution protocol (ARP) spoofing attacks. The application layer is vulnerable to application programming interface (API) exploitation attacks [Rahouti, Xiong, Xin, Jagatheesaperumal, Ayyash & Shaheed (2022)]. Since SDN-based architectures are mostly dependent on the programs (software), SDN security vulnerabilities are particularly alarming as they can jeopardize entire networks. Successful attacks in software-defined networks can cause more control loss and the intruders may take full control of the system and put the entire infrastructure at stake.

In cloud computing, multiple security vulnerabilities are yet to be addressed. For example, a lack of strict isolation at multiple levels in multi-tenant shared cloud infrastructures among multiple virtual network operators exposes users to information confidentiality and integrity risks. Moreover, according to the 5G infrastructure public private partnership (5G PPP) phase 1 security landscape [Bisson & Waryet (2017); Mishra, Pilli, Varadharajan & Tupakula (2017)], network slicing has several open security challenges, such as security isolation of network slices and security of inter-slice communications [Ahmad, Kumar, Liyanage, Okwuibe, Ylianttila & Gurtov (2018)].

1.3.3 Recent Approaches to IoT Security

To overcome the aforementioned IoT security threats and vulnerabilities, the research community has investigated innovative security methods that accommodate IoT constraints and specifica-

tions. For example, multiple research works have come up with new lightweight encryption, authentication, and authorization methods designed explicitly for resource-constrained IoT systems [Dutta, Ghosh & Bayoumi (2019); Surendran, Nassef & Beheshti (2018)].

In [Tiburski, Moratelli, Johann, Neves, Matos, Amaral & Hessel (2019)], the authors proposed a lightweight security architecture based on embedded virtualization and trust mechanisms for IoT edge devices. The proposed scheme is composed of four security mechanisms that ensure the authenticity of the executed code, the integrity of the runtime states, and the confidentiality of elements stored in the persistent memory. These security mechanisms include secure boot, secure key storage, security by separation, and secure inter-domain communication mechanisms. The secure boot authenticates the stored programs with asymmetric keys so that eventual modifications can be detected during the boot process. The secure key storage protects the integrity of keys using write-once memory. The security by separation uses spatial and temporal separation to avoid software defects propagation from part of the devices to adjacent parts. With the proposed mechanisms, on the one hand, a compromised VM code will not pass the boot-up process. On the other hand, attackers that exploit vulnerabilities based on malfunctions can be detected, enabling developers to investigate the causes.

The work in [Gope & Sikdar (2019)] demonstrated a lightweight and privacy-preserving two-factor authentication scheme, where each IoT device uses a secret key and physically unclonable functions (PUFs) as factors for proving its legitimacy to the servers. PUFs are the result of the manufacturing process of integrated circuits (ICs), which introduces random physical variations into the micro-structure of an IC, making it unique. It is impossible to control these variations in the micro-structure of an IC during the manufacturing process. In addition, the outputs are derived from intrinsic characteristics of the PUF's physical elements and are therefore difficult to predict and almost impossible to clone. In this regard, PUFs use their internal structure to provide a one-way function that cannot be duplicated. Thus, PUFs have emerged as a promising cryptographic primitive and already gained popularity in the security domain. The fact that PUFs are hard to predict but easy to construct and evaluate makes them a good choice for use as a security primitive for IoT devices. However, noise and sensitivity to environmental factors are

still important factors in PUF design, which may result in one or several of the output bits of the PUF being incorrect for any challenge. The authors addressed this issue using the concept of reverse fuzzy extractor. The proposed scheme remains secure even if an adversary has physical access to an IoT device. This approach provides protection against physical attacks, cloning, impersonation, message tampering, and replay attacks.

The authors in [Arafin, Gao & Qu (2017); Zhang *et al.* (2022)] proposed a hardware-oriented lightweight authentication protocol that uses the errors generated by the computing units of digital circuits in voltage over-scaling (VOS). Like PUFs, these errors are also related to the manufacturing process variation and hence serve as hardware fingerprints. The authors used such errors as device signatures and designed a lightweight two-factor authentication scheme that uses passwords as something known and hardware properties as something possessed by legitimate users. VOS-based authentication was shown suitable for resource-constrained IoT applications. Compared to PUFs, VOS-based authentication benefits from a lower power consumption and no additional hardware is required for its implementation. The evaluation of the protocol proved its effectiveness against many potential attacks, such as random guessing, eavesdropping, MitM, compromised key, and side-channel attacks. However, in simulations performed in the HSpice platform using the FreePDK 45 nm libraries [Stine, Castellanos, Wood, Henson, Love, Davis, Franzon, Bucher, Basavarajaiah, Oh & Jenkal (2007)], the solution proposed has shown vulnerabilities against machine learning attacks, such as artificial neural network (ANN), recurrent neural network (RNN), and covariance matrix adaptation evolution strategy (CMA-ES).

The study in [Lam, Mitra, Gondesen & Yi (2022)] proposed a secure-by-design activity-network-things (ANT)-centric reference architecture based on the three IoT architectural perspectives. Security-by-design is an emerging paradigm that aims to deal with security concerns from the early phases of system development [Salnitri, Alizadeh, Giovanella, Zannone & Giorgini (2018)]. Secure-by-design approaches are widely accepted as the safest, most economical, and most resourceful way to secure complex systems. The proposed architecture was designed to secure a space-air-ground-integrated network (SAGIN)-enabled internet of vehicles (IoV), and

more globally, for any IoT application. SAGIN is an ideal alternative to traditional networking technologies which enables anytime anywhere IoV connectivity. However, significant security challenges are yet to be addressed as SAGIN-enabled IoV applications are vulnerable to diverse cyber-attacks due to their open and heterogeneous nature. The proposed architecture includes an organized process to understand the security requirements and select specific parameters for tailored security controls. The authors discussed some features for different security mechanisms, including public key, implicit hardware, and symmetric cryptography with delayed key disclosure-based authentications. They also conducted a scalability impact analysis of the security schemes. The proposed security reference architecture is built in a flexible way so that the practitioners can add or replace security features suiting their own requirements and based on the latest standards and best practices.

The paper [Cao, Ding, Wang, Lv, Tian, Wei & Gong (2022)] introduced a non-orthogonal multiple access (NOMA) assisted semi-grant-free transmission to enhance next generation IoT physical layer security. Unlike the conventional orthogonal multiple access (OMA) wireless communications systems, NOMA encourages spectrum sharing among wireless devices and has emerged as a spectrally efficient solution for supporting massive connectivity of IoT devices, which is a key step towards next generation IoT [[Ding (2021); Ding, Liu, Choi, Sun, Elkashlan, Chih-Lin & Poor (2017)]. However, physical layer security of NOMA is facing great challenges. The authors investigated the security of semi-grant-free NOMA transmission in the presence of passive and active eavesdropping attacks. In particular, for a first scenario (scenario-I) with strong grant-based user and weak grant-free users, scenario-I based maximal user scheduling (IbMUS) and scenario-I based optimal user scheduling (IbOUS) schemes are proposed to combat passive and active eavesdropping, respectively. For a second scenario (scenario-II) with weak grant-based user and strong grant-free users, two parallel schemes, namely the scenario-II based maximal user scheduling (IIbMUS) and scenario-II based optimal user scheduling (IIbOUS) schemes, are proposed to combat the passive and active eavesdropping, respectively. These proposed schemes enhance the security by scheduling a grant-free user with maximal main channel capacity/maximal secrecy capacity to access the NOMA channel on the premise of

ensuring the grant-based user's quality of service. Based on these proposed schemes, the exact secrecy outage probability (SOP) was analyzed to evaluate the system's performance.

1.4 Intrusion Detection and Prevention Systems (IDPSs) for IoT

Most of the security solutions presented above, including hardware and firmware integrity, lightweight authentication, user data confidentiality, and wireless communication security methods can all be classified as protection layer security, *i.e.*, the first layer of the global cyber-security framework. This section describes the second cyber-security layer, *i.e.*, IDPSs, explains its role, and provides a comprehensive review of intrusion detection methods and techniques. Specifically, we focus on anomaly-based detection methods that use machine learning algorithms, especially deep learning.

1.4.1 Role of IDPSs

In the global cyber-security framework, the protection layer represents the first security layer, which defines and implements protective safeguards to avoid the violation of the systems' security requirements (*i.e.*, confidentiality, integrity, availability, etc.). However, a protection layer alone is not enough to ensure complete cyber-security. Despite the deployment of strong protection measures, a system can still be compromised by an enduring adversary using zero-day attacks, advanced techniques, high computational resources, or leaked security information. Therefore, as defined by the national institute of standards and technology (NIST), under every protection layer, must be implemented a detection, response (also known as prevention), and recovery layers [Mylrea *et al.* (2017); Norris *et al.* (2022)]. Figure 1.13 illustrates the NIST's cyber-security framework and highlights our research focus. This research work focuses on intrusion detection systems (IDSs) and intrusion prevention systems (IPSs).

The function of an IDS is to monitor relevant activities from a security perspective, such as system logs and network packets, to analyze and detect activities that are potentially worthy of some type of action or response (*e.g.*, multiple failed logins or a ping flood) [Betser, Walther,

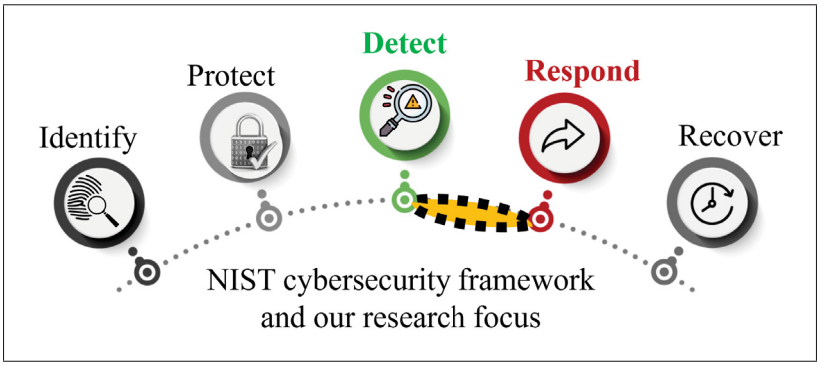

Figure 1.13 NIST's cyber-security framework and our research focus

Erlinger, Buchheim, Feinstein, Matthews, Pollock & Levitt (2001)]. The degree to which an event is worthy of a response is dictated by the security policy in place and is typically implemented in the configuration of the IDS. Once an event has been detected and deemed worthy of further action, it is necessary to communicate the existence of the event, and possibly its details, to a party that is capable of further action. The role of an IPS, also known as an intrusion response system (IRS) is to automatize the function of the IDS receiving party with programmed defensive measures. Compared to manual response, an IPS reduces the window of vulnerability between when an intrusion is detected and when the defensive action is taken. This solution has a significant impact as the success of many attacks mainly depends on the time gap between the detection and the defensive response.

IDSs communicate with IPSs or other receiving parties through alerts that contain, as a minimum, an information stating that an event occurred and ideally much more information about the event in order to facilitate an informed action or response by the receiving party. Figure 1.14 illustrates a generic IDS architecture as defined by the IETF intrusion detection working group (IDWG) [Brandao *et al.* (2006)]. In this architecture, the sensor is the entity that collects data (relevant activities) to be used in the detection process. The analyzer detects events based on activity data, and according to the security policy in place possibly generates alerts. Alerts are formatted and transferred to managers. The manager can inform the operator through different types of notification that important alerts have occurred, as per the security policy.

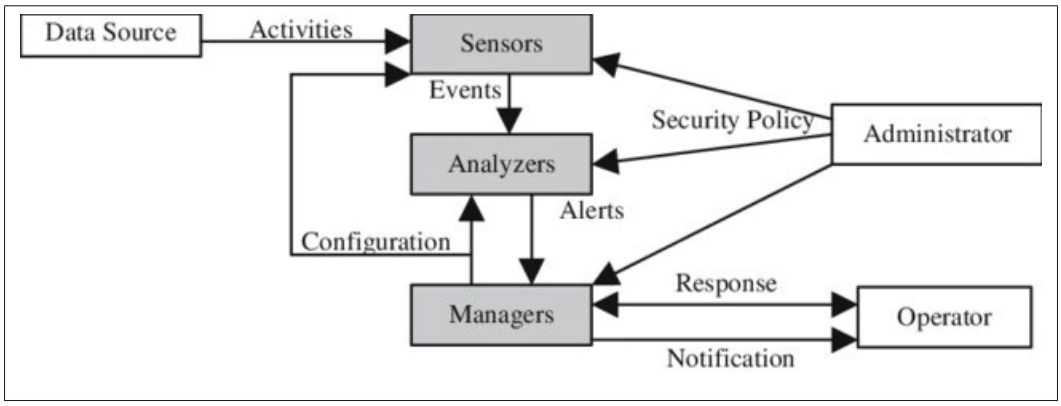

Figure 1.14 Basic elements of the IETF IDWG IDS model
Taken from Brandao *et al.* (2006)

IDSs can be categorized using different classification criteria, such as the detection phase, which includes online and offline IDSs; the detection surface, which includes network-based (NIDS), host-based (HIDS), and hybrid intrusion detection systems; and the detection method [Bhuyan, Bhattacharyya & Kalita (2014); Sabahi & Movaghar (2008); Vaigandla, Azmi & Karne (2022)].

1.4.2 Intrusion Detection Methods

The fundamental idea behind intrusion detection methods is that an intrusive activity always has some elements of differences compared to normal activities. Therefore, based on those differences, intrusive activities can be detected. A variety of methods for intrusion detection have been proposed in the literature. The most used methods include signature-based, specification-based, and anomaly-based detections.

1.4.2.1 Signature-based Detection

Signature-based detection, also known as misuse detection, relies on a signature database to discover attacks [Kemmerer & Vigna (2002)]. The signature database contains predefined signatures, which are created from rules that match exploits and patterns of known intrusive activities. Figure 1.15 presents the structure (1) and an example (2) of a detection rule used on SNORT, one of the most popular commercial IDSs. In a deployed signature-based IDS, the

detection mechanism consists of extracting the signatures from the new traffics and matching them with all the defined signatures in the database. Once a signature is matched, the corresponding traffic is reported as an intrusion.

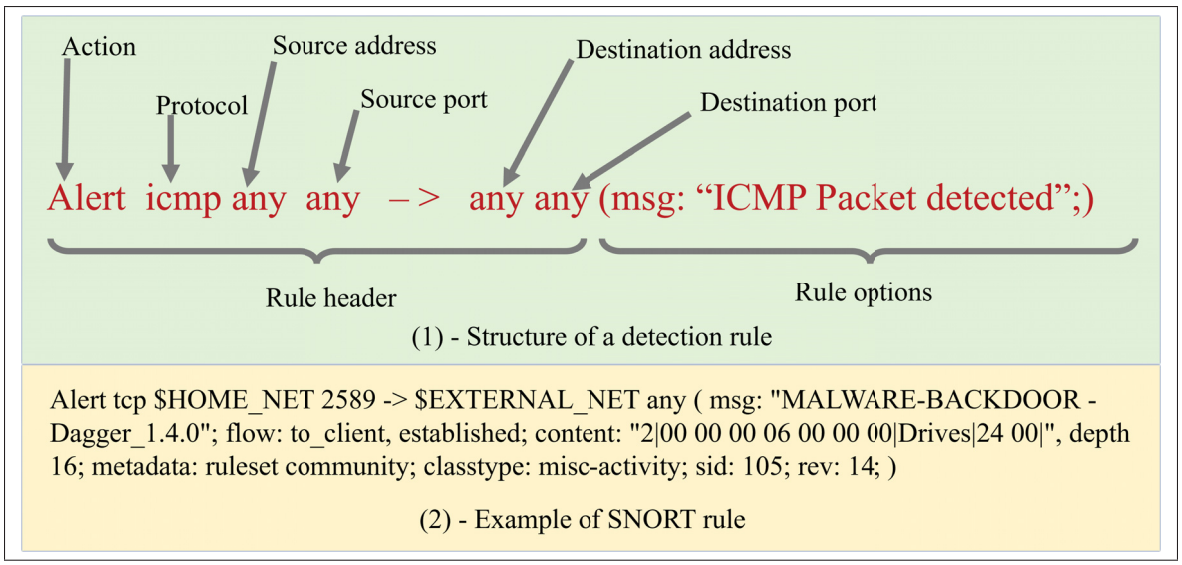

Figure 1.15 Structure and example of a detection rule

The main advantage of IDSs that use this method is that they typically produce few false positives. However, their major drawback is their inability to detect zero-day-attacks, *i.e.*, previously unseen attacks which exploit unaware security flaws. Another major drawback is their inability to detect attacks for which signatures are not yet added to the database. Therefore, keeping the signature database up to date is mandatory. Thus, for every newly discovered attack, the corresponding signature must be created and added to the signature database. Moreover, signature-based IDSs require a deep understanding of how the vulnerabilities actually work in order to develop the detection rules.

1.4.2.2 Specification-based Detection

IDSs that use this method rely on sets of predetermined specifications, such as the system's functional states and transitions, operational and protocol constraints, network maximum capacity of links, packet size, and statistical rules that define the system's legitimate behavior model

[Jokar & Leung (2018)]. During the monitoring process, the IDS analyzes the system's events and when an activity deviates from a predefined specification, an alert is issued.

With such a method, it is easy to detect known attacks. Another advantage is that, since specification-based IDSs detect deviations from normal behaviors, they can detect previously unseen attacks as well. However, most specifications are manually defined by human experts and are strongly dependent on the expertise of the system administrators. This causes significant shortcomings. Firstly, manually defining such rules can be task-intensive in large networks, which poses a scalability issue in IoT applications. Secondly, manual tasks increase the likelihood of human errors, which can lead to abundant false positives. Moreover, incomplete specifications can cause IDSs to miss benign-looking attacks and cause false negatives, which represent a considerable risk to IoT security. Furthermore, manually defined specifications may not adapt to different environments and may be incompatible with some proprietary operating systems, applications, or protocols.

1.4.2.3 Anomaly-based Detection

In general, anomaly detection aims to find patterns in events that do not conform to expected behavior [Zhou & Guo (2018)]. Therefore, an anomaly-based IDS attempts to model the normal behavior of a targeted system beforehand. Then, it compares the current behaviors of the targeted system with the modelled behavior for notable deviations. Finally, it identifies as an anomaly any observation with a behavior deviation that exceeds a predefined threshold. Another approach in anomaly-based IDSs is to model the abnormal behavior of the targeted system and to raise an alarm when the difference between an observed behavior and the modelled behavior falls below a given limit [Garcia-Teodoro, Diaz-Verdejo, Maciá-Fernández & Vázquez (2009)]. Anomaly detection methods can be classified based on their underlying computation techniques. The most common techniques include statistical-based, knowledge-based (rule, expert system, and logic), and machine learning-based techniques.

Anomaly detection is capable of detecting well-known attacks as well as zero-day attacks. However, the drawback of this method is its high false alarm rate due to the difficulties to model complex systems' behavior and setting up the right thresholds for identifying the anomalies. Based on the work presented by [Wang & Jones (2017)], we have built a comparison table of the presented intrusion detection methods (Table 1.2).

Table 1.2 Comparison between intrusion detection methods
Taken and adapted from Wang & Jones (2017)

IDS Methods	**Characteristics**	**Advantages**	**Disadvantages**
Signature-base detection	Uses the patterns of known attacks (signatures) to identify intrusions.	- Low alarm rate; - High speed; - Low resource consumption; - High reliability.	- Known attacks have to be hand-coded; - Unable to detect new attacks; - Need signatures update; - False negatives; - Low flexibility; - Low scalability; - Low robustness.
Anomaly-based detection	Uses the deviation from normal usage patterns to identify intrusions.	- Can detect new attacks; - High flexibility; - High scalability; - High robustness.	- Has to study sequential interrelation between; transactions; - False positives; - High alarm rate; - Low speed; - High resource; consumption; - Moderate reliability.
Specification-based detection	- Depends on vendor-developed generic profiles to specific protocols; - Protocols based on standards from international standard organizations.	- Know and trace protocol states; - Distinguish unexpected sequences of commands.	- Resource consuming to protocol state tracing and examination; - Unable to inspect attacks looking like benign protocol behaviours; - Might be incompatible with dedicated OSs or applications.

1.4.3 Recent Works on IDPSs for IoT

Many research works have come up with different IDPSs to provide more security in IoT applications. Out of all the implemented methods, anomaly-based IDSs that employ machine learning, especially deep learning techniques, have emerged as the most promising approaches. Supervised machine learning techniques can automatically learn from datasets of previous malicious and benign system events to build detection models that can identify future malicious events. However, generating a realistic learning dataset is often a challenging task. Therefore, many research works use available datasets generated from traditional computer systems because of the lack of challenging public datasets created specifically for IoT security. Some other research works opt to build their own datasets from private IoT networks. Overall, the most used datasets in the literature for NIDS include KDDCUP99 (KDD99), NSL-KDD, UNSW-NB15, CAIDA, DEFCON, ADFA IDS, KYOTO, and ISCX 2012 [Chaabouni *et al.* (2019); Moustafa & Slay (2015b)]. Table 1.3 presents a comparison between some of these popular datasets.

Diverse learning algorithms to build efficient intrusion detection models in IoT are proposed in the literature and the performance of the presented solutions are evaluated using different metrics. These metrics are mostly derived from the confusion matrix, as shown in Table 1.4. In this table, the parameter true negatives (TN) represents the number of normal events correctly classified as normal; True positives (TP) represents the number of abnormal events (attacks) correctly classified as attacks; False positive (FP) represents the number of normal events incorrectly classified as attacks; and False negative (FN) represents the number of abnormal events (attacks) incorrectly classified as normal events.

The most used evaluation metrics in the literature include accuracy, precision, false alarm rate (FAR), undetected rate (UR), and sensitivity. According to the confusion matrix, these metrics are defined as follows:

Table 1.3 Comparison between datasets
Taken from Chaabouni *et al.* (2019)

Datasets	Advantages	Drawbacks
KDD99	- KDD99 is popular and the most used. - Labeled data. - It is based on 41 features for each connection along with the class label. - Implements DoS, Remote to Local, U2R, and Probing attacks. - Provides network traffic (PCAP).	- KDD99 suffers from unbalanced classification methods. - The dataset is out of date. - Not for IoT systems.
NSL-KDD	- It is a better version of KDD99. - It overcomes KDD99 limitations. - No duplicated records in the training and test sets.	- Lack of modern low foot print attack scenarios. - Not for IoT systems.
UNSW-NB15	- It provides hybrid real modern normal activities and synthetic contemporary attack behaviors. - Provides network traffic (PCAP) and comma-separated values (CSV) files. - It has nine types of attacks, namely Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.	- It is more complex than the KDD99 dataset due to the similar behaviors of the modern attack and normal network traffic.
Sivanathan et al. Dataset	- Network traffic IoT dataset. - It reflects real world IoT systems. - Provides PCAP and CSV files.	- Unlabeled data. - For IoT devices proliferation and traffic characterization. - No attack data.
CICIDS	- Labeled network flows. - For machine and deep learning purposes. - Provides PCAP and CSV files. - Implements attacks, such as brute-force file transfer protocol (FTP), brute-force secure shell (SSH), DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS.	- Not public. - Not for IoT systems.
CSE-CIC-IDS2018	- Labeled network flows. - For machine and deep learning purposes. - Provides PCAP, CSV and log files. - Implements Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and local network infiltration attacks. - Dynamically generated dataset. - It is modifiable, extensible, and reproducible.	- Not public. - Not for IoT systems.

Table 1.4 Confusion matrix

		Predicted Class	
		Classified as Normal	**Classified as Attack**
Actual Class	**Normal Data**	True Negative (TN)	False Positive (FP)
	Attack Data	False Negative (FN)	True Positive (TP)

- Accuracy: Shows the percentage of the correctly predicted samples considering the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{1.1}$$

- Precision: From all the classes predicted as positive, shows the percentage of classes that are actually positive.

$$Precision = \frac{TP}{TP + FP} \times 100 \tag{1.2}$$

- False alarm rate (FAR): Also known as false positive rate (FPR). It represents the percentage of regular samples misclassified as attacks.

$$FAR = \frac{FP}{FP + TN} \times 100 \tag{1.3}$$

- Undetected rate (UR): The percentage of anomaly samples (attacks) misclassified as normal.

$$UR = \frac{FN}{FN + TP} \times 100 \tag{1.4}$$

- Sensitivity: Also known as the true positive rate or recall. It shows from all the positive classes, the percentage of those who are predicted correctly.

$$Sensitivity = \frac{TP}{TP + FN} \times 100 \tag{1.5}$$

- F-measure: Also known as F1 score. It calculates the harmonic mean of the precision and recall and provides a single weighted metric to evaluate the overall classification performance.

$$F - measure = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} \tag{1.6}$$

Numerous learning-based IDSs presented in the literature implemented traditional machine learning models, *i.e.*, shallow learning models, which are typically built with less than three computational layers. These models may be suitable for resource-constrained IoT and other edge devices as they do not require high computational power and have a relatively short training time. However, shallow learning models may encounter difficulties in discovering useful intrusion patterns from an increasing number of training samples. Therefore, deep learning models, which are built with a large number of neural layers, have emerged as much more promising models to build robust IDSs and are subject of more and more research works.

1.4.3.1 Traditional Shallow Machine Learning-based Detection

The research in [Sedjelmaci, Senouci & Taleb (2017)] designed a lightweight IDS that combines the advantages of anomaly and signature-based detection techniques. The anomaly-based detection technique relies on a learning algorithm to model the system's normal behavior and detect attacks. When a new attack pattern is detected, its signature is extracted and used by the signature-based detection technique. The combination of these two detection methods (anomaly and signature) generally exhibits high detection and low false positive rates. However, with low-resource IoT devices, this could lead to high-energy consumption, increased overhead, and degraded network performance. To mitigate this issue, the authors proposed to activate the anomaly detection only when a new attack pattern (*i.e.*, signature) is expected to occur. Hence, a balance between detection accuracy, false positive rates, and energy consumption can be achieved. The activation of the anomaly detection technique is done thanks to a proposed security game theoretic model, where they modelled the security strategy as a game formulation between the intruder attack and the IDS agent embedded in IoT devices. With the help of Nash Equilibrium, they determined the equilibrium state that allows the IDS agent to activate its

anomaly detection technique in order to detect new attack patterns. To further decrease the false positive rate, a reputation model based on game theory is proposed. This model aims to rank the monitored IoT devices into Legitimate, Suspect, and Malicious nodes according to their reputation scores. The performance analysis demonstrated the viability of the proposed approach in wireless sensor networks (WSNs) using TOSSIM (TinyOS library simulator). The simulation results showed that the proposed solution requires low energy consumption to detect the attacks with high detection and low false positive rates, almost 93% and 2%, respectively.

The work in [Teixeira, Salman, Zolanvari, Jain, Meskin & Samaka (2018)] created an IIoT testbed where they simulated attacks, built a dataset, and trained and evaluated machine learning-based detection models. The testbed consists of a water storage tank's control system, which is a stage in the process of water treatment and distribution. The authors simulated reconnaissance attacks against the testbed and captured the network traffic. They used TCP/IP packet headers and computed statistical data as detection features. The features in the dataset include the total packet count of each network transaction, the total transaction bytes, the source-to-destination packet count, the destination-to-source packet count, the source-to-destination transaction bytes, and the port number of the source. Five machine learning algorithms were trained to detect the attacks, including random forest, decision tree, logistic regression, naïve Bayes, and k-nearest neighbors (KNN). Considering the offline evaluations, all the machine learning algorithms performed well in terms of accuracy. The decision tree, random forest, naïve Bayes, and logistic regression models showed similar performance during online evaluations, compared to the offline evaluations. The same did not apply to the KNN model. There was a significant difference between the online and offline phases, which indicates that in practice, the KNN did not provide good accuracy.

In [Anthi, Williams, Słowińska, Theodorakopoulos & Burnap (2019)], the authors proposed a three-layer IDS that uses a machine learning approach to detect a range of popular network-based cyber-attacks on IoT networks. The first layer classifies the type and profiles the normal behaviour of each IoT device connected to the network so that unusual behaviour can be detected, and subsequently, so can cyber-attacks. The second layer identifies malicious packets on the

network when an attack is occurring. The third layer classifies the type of attack that has been detected so that crucial information is provided to help determine the severity of the attack, and subsequently accelerate the launch of countermeasures to defend against it without significant human efforts. The system is evaluated within a realistic smart home testbed consisting of 8 popular commercially available IoT devices connected to an access point via WiFi and Ethernet communications, and a laptop to record the network traffic and perform various network-based attacks. Three weeks worth of benign data and two weeks of malicious data were collected from the wireless access point using the tcpdump tool. The effectiveness of the proposed IDS architecture was evaluated against 12 popular attacks from 6 categories found within the IoT domain, but also against four scenarios of scripted multi-stage attacks with complex chains of events. The attacks considered include various reconnaissance attacks, like quick scan and intense scan, iot-scanner, various DoS, including TCP, user datagram protocol (UDP), and hello floods, various MitM (ettercap, ARP), replay attack, ARP and domain name system (DNS) spoofing, and 4 multi-stage scripts. Nine classifiers were tested and the J48 decision tree method with pruning achieved the best performance, resulting in an F-measure of 99.7%, 97.0%, and 99.0% for device identification, attack detection, and attack classification, respectively.

The authors in [Eskandari, Janjua, Vecchio & Antonelli (2020)] presented an anomaly-based IDS, named Passban, purposely designed to be directly hosted and executed by very cheap, commercially available IoT gateways (*e.g.*, single-board personal computer) or other typical edge devices. First, they built an IoT testbed able to resemble a typical smart home automation environment. Then, exploiting legitimate traffic flows of the IoT target network, they implemented two one-class classification techniques, namely isolation forest (i forest) and local outlier factor (LOF). The models were evaluated against four common attacks, namely port scanning, hypertext transfer protocol (HTTP) brute-force, SSH brute-force, and synchronization (SYN) flood attacks. In a first scenario, the IDS is directly deployed and executed on the IoT gateway, where Passban was able to protect the latter and all the IoT devices directly connected to it. In a second scenario, Passban is provided as a separate add-on device independently connected to the network it has to protect. In terms of threat detection accuracy, the experimental evaluation revealed that using

i Forest, Passban IDS can achieve F1 scores greater than 90% on some attacks (99% in the best case and 79% in the worst case). In terms of resource utilization, they proved that Passban can be executed even on cheap IoT gateway boards, like a Raspberry Pi 3 model B, especially when the data rate on its network interface is not expected to exceed 40-50 megabits per second (Mbps).

1.4.3.2 Deep Learning-based Detection

The study in [Yao, Gao, Zhang, Wang, Jiang & Lu (2019)] proposed a hybrid intrusion detection architecture for IIoT using a traditional machine learning algorithm, a light gradient-boosting machine (LightGBM), and a deep learning algorithm, a convolutional neural network (CNN) in the lower level and upper level of the network, respectively. LightGBM can be viewed as a fast, distributed, high-performance decision-tree-based gradient lifting framework. In the proposed architecture, devices with strong computing power and sufficient resources, such as edge routers, are regarded as the master nodes, while the industrial equipment of the edge part is regarded as edge nodes. They apply the lightweight LightGBM algorithm on the edge nodes and perform the first intrusion detection task at these nodes to ensure security. At the same time, they utilize the LightGBM algorithm to extract more advanced features to represent the traffic without increasing the resource consumption, so as to perform further detection on the master nodes. On the master nodes, they employ the deep learning algorithm with higher accuracy to perform the second intrusion detection task, further improving the detection accuracy of the overall network.

The work in [Ullah & Mahmoud (2021)] designed and developed an anomaly-based IDS for IoT networks using a deep learning-based model for multiclass and binary class traffic classification. The proposed system uses a CNN architecture in the multiclass classifier to categorize 15 types of attacks and regular network traffic. The model consists of an input layer, four blocks of convolutional layers, a flatten layer, a fully connected layer, and an output layer. The input layer receives inputs from a reshaping system, which transforms the network data into a format compatible with CNN one-dimension (CNN1D), CNN two-dimensions (CNN2D), and CNN three-dimensions (CNN3D) models. The proposed models were validated using many subsets

of the IoT-DS-2 dataset, including BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, and IoT-23. The authors used the transfer learning principle to deploy a pre-train multiclass CNN model for the multiclass and binary models. According to the authors, transfer learning from a multiclass CNN model to a binary class CNN model was effective because the binary CNN model was trained using a subset of data used by the multiclass classification model. The experimental results showed a minimum detection rate for the CNN1D model at 99.74%, CNN2D model at 99.42%, and CNN3D at 99.03% for BoT-IoT, MQTT-IoT-IDS2020, IoT-23, and IoT-DS-2 datasets.

In [Sun, Lai, Wang, Liu, Mao & Gu (2022)], the authors proposed a logic understanding IDS (LU-IDS) which uses a specially designed deep learning-based model to capture features automatically and carry out attack classification. The proposed LU-IDS analyzes the knowledge learned from the classification of attacks to understand an abnormal industrial control logic and generate rules to implement a rule-based IDS with in-depth understanding of industrial control logic, rather than manually designing rules based on the experience of domain experts. They designed three modules for the proposed LU-IDS. The first module is a tensor construction module, which builds two-dimensional measured element tensors by embedding all the sensor and actuator values in each moment. The second is the logic learning module, which takes the built tensors as input and uses convolution layers with a kernel size of $n \times 1$ to extract features from them automatically. The third module, the rule generation module, analyzes the knowledge learned by the deep learning-based model and generates interpretable rules with high confidence to detect attacks. The performance of the proposed solution is evaluated using two datasets. The first dataset, named SWaT, was collected in a six-stage safe water treatment testbed, which represents a scaled-down version of a real-world industrial water treatment plant. The second dataset, named WADI, was collected in a down-scaled water distribution testbed.

The authors in [Houda, Brik & Khoukhi (2022)] proposed an explainable artificial intelligence (XAI)-based framework for real-time intrusion detection in IoT networks. XAI is an emerging paradigm of AI, that provides a set of techniques to help interpret and understand predictions made by deep learning models. Once the deep neural network (DNN) makes decisions, the

proposed framework leverages three different approaches of XAI, including local interpretable model-agnostic explanations (LIME), Shapley additive explanations (SHAP), and RuleFit to add more explainability, transparency, and trust to the model's decisions. The proposed framework targets two different users, *i.e.*, users of the deep learning model that aim to understand and trust the model's outputs, in order to be able to optimize their decisions, and cyber-security experts that also aim to understand the model's outputs, in order to make the suitable recommendations, especially when an intrusion is detected. In the experiment, this work used both NSL-KDD and UNSW-NB15 datasets to demonstrate the feasibility and performance of the proposed solution.

CHAPTER 2

ML-BASED IDPS ENHANCEMENT WITH COMPLEMENTARY FEATURES FOR HOME IOT NETWORKS

Poulmanogo Illy [a], Georges Kaddoum [a], Kuljeet Kaur [a] and Sahil Garg [a],

[a] Département de Génie Électrique, École de Technologie Supérieure,
1100 Rue Notre-Dame Ouest, Montréal, Québec, H3C 1K3, Canada

Paper published in *IEEE Transactions on Network and Service Management*, Vol. 19, Issue. 2, pp. 772 - 783, January 2022

2.1 Introduction

Home is one place where everyone needs absolute privacy and security to relax, take care of their families, and keep valuable objects with complete peace of mind. To guarantee these vital necessities, it is important to prevent intrusions. Several efforts, including locksmithing, home surveillance, alarms, and law enforcement operations, are employed to ensure security. However, multiple intrusions in residential properties are perpetrated on a daily basis. According to the crime statistics released by the Federal Bureau of Investigation (FBI) in September 2019, up to 1.2 million burglaries were reported in the USA in 2018, *i.e.*, 1 burglary every 25.7 seconds; making it the second-most frequent crime. The victims of these burglaries suffered an estimated property loss of $3.4 billion. Almost 66% of all the theft offenses were accounted as burglaries of residential properties [U.S. Department of Justice—FBI (2019)]. Furthermore, an intrusion inside a house may often result in more critical consequences, including kidnapping, terrorism, and assassination for political, economic, and other reasons.

Meanwhile, homes are entering modernization enabled by the emerging Internet of Things (IoT). The IoT paradigm offers immense opportunities in many domains and has created new concepts such as smart homes, smart cities, and smart industries [Perera, Liu & Jayawardena (2015)]. These technologies are expected to be even more widespread with the emerging Fifth-Generation (5G) of mobile communications [Palattella, Dohler, Grieco, Rizzo, Torsner, Engel & Ladid

(2016)]. Various scenarios can motivate one to transform a traditional home into a smart home. These reasons include increased home comfort, optimized energy consumption, automated kitchen, senior citizens' independence, and most importantly, enhanced home safety and security (remote surveillance of security cameras and sensors, access control, central locking for all perimeter doors and windows, person identification, personalized alerts, etc.). However, the IoT network can become a major susceptibility in home security. This can be attributed to the security vulnerabilities in smart devices that could be a leverage for hackers to carry out nefarious activities. These inexpensive devices are generally not designed with any security mechanisms. Usually, IoT device manufacturers compromise on security measures to keep pace with market needs; for example, by reducing the time-to-market for their devices and minimizing the design and development costs. These products, thus, end up being attractive targets for adversaries [Alladi, Chamola, Sikdar & Choo (2020)]. For example, researchers have found critical weaknesses in IoT-enabled baby monitoring systems [Forbes (2019)], which provide access to malicious users to view and control our monitoring systems remotely. The authors in [Wurm *et al.* (2016)] survey the security of popular consumer and industrial IoT devices and highlight the associated security vulnerabilities. A successful cyber-attack in a smart home network may result in more critical consequences, including leakage of highly sensitive data that later can be used to blackmail the owner for ransom or reputation damage. Network attacks can also be employed to remotely control access points and security devices, such as doors, windows, surveillance cameras, occupancy detection sensors, etc., to enable physical intrusion.

Smart home IoT networks can be attacked through wireless or wired channels. Wireless networks are vulnerable to diverse security threats, including unauthorized user access through the wireless gateways, Denial of Service (DoS) by de-authentication attacks or jamming the wireless communication, eavesdropping, identity spoofing of an IoT node or the wireless access point (dummy gateway, rogue access point), man-in-the-middle (MITM) attack, and message falsification/injection, to name a few [Zou *et al.* (2016)]. Traditional cyber-attacks through the wired networks are also used against IoT networks.

To deal with these threats, it is essential to implement effective security solutions. The research work in this field is facing several issues. Unlike traditional computer network security challenges, IoT infrastructures introduce new significant challenges. Firstly, IoT nodes are characterized by limited resources due to the limited or small size of the devices. These limitations include low computational power, memory size, radio bandwidth, and energy budget [Heer *et al.* (2011)]. These constrained devices do not provide enough resources to execute computationally and latency-sensitive security tasks that generate heavy computation and communication load. Therefore, it is not feasible to directly implement complex security mechanisms. This implies that IoT security technologies must accommodate the constraints of IoT devices. The second main challenge in IoT network security is the heterogeneity in terms of hardware, software, and protocols. A security measure that is appropriate for one IoT device may not be applicable to another. Moreover, each device generally has different security requirements and implementation. In a network built with components from diverse companies (device manufacturers, software developers, network operators, deployment companies) with various technologies, it is not easy to define who will be responsible of the global security orchestration [Restuccia *et al.* (2018)]. Another crucial aspect is that, given the number of these devices, it is challenging to develop and deploy security mechanisms that can endure with the scale and range of devices. For instance, authentication and authorization through cryptographically pre-shared keys are not applicable. The rapidly growing number of objects will make key management become a problematic task [Zhang *et al.* (2014)]. It is also important to point out that due to the distributed nature of the IoT, most of the generated traffic will be wireless in nature [Xu, He & Li (2014)]. In this context, wireless attacks are significantly dynamic in nature, usually hard to predict and relatively simple to carry out [Mpitziopoulos *et al.* (2009)].

2.1.1 Related Work

To overcome the aforementioned challenges, the research communities have been investigating innovative security methods that accommodate with the IoT constraints. Indeed, some research works have come up with new light weight cryptographic algorithms designed explicitly for

encryption, hashing, and authentication on the constrained IoT resources [Dutta *et al.* (2019); Surendran *et al.* (2018)]. However, cryptography is only meant to prevent attacks. On the other hand, some important research work focuses on Intrusion Detection Systems (IDSs), so that malicious attempts (successful or failed) can be detected beforehand.

The most used intrusion detection methods in the state-of-the-art include signature-based [Kemmerer & Vigna (2002)], anomaly-based [Chandola, Banerjee & Kumar (2009); Zhang, Yang & Geng (2009)], and Specification-based [Zarpelão, Miani, Kawakani & de Alvarenga (2017)] detections. Table 2.1 presents a comparison of these methods in accordance with a survey given in [Wang & Jones (2017)].

Table 2.1 Comparison between different intrusion detection methods

Method	**Advantages**	**Disadvantages**
Signature-based	Low false alarm rate; High speed; Low resource consumption; High reliability.	Known attacks must be hand-coded; Fails to detect new attacks; Requires signature update; Generates false negatives; Low flexibility; Low scalability; Low robustness.
Anomaly-based	Detects new attacks; High flexibility; High scalability; High robustness.	Requires the study of sequential interrelation between transactions; Produces false positives; High alarm rate; Low speed; High resource consuming; Moderate reliability.
Specification-based	Knows and traces the normal protocol states; Distinguishes unexpected sequences of commands.	Consumes significant resources for protocol state tracing and examination; Fails to inspect benign looking like attacks; Might be incompatible to some proprietary Operating Systems (OSs) or applications; Protocol dependant.

Recent studies that investigated intrusion detection for IoT networks employed anomaly-based detection and used different machine learning (ML) techniques with various detection features. The typical approach consists of designing a realistic IoT testbed [Anthi *et al.* (2019); Nobakht, Sivaraman & Boreli (2016)] or a simulated environment [Amouri, Alaparthy & Morgera (2018); Facchini, Giorgi, Saracino & Dini (2020)], and launching multiple attacks in the experimental network in order to collect and label the data (normal and attack traffic). Then, the generated data is used in supervised learning algorithms to train the intrusion detection models. To improve the detection accuracy, most researchers employ hyper-parameters tuning or implement various learning algorithms [Khan, Abbas, Rehman, Saeed, Zeb, Uddin, Nasser & Ali (2020); Siniosoglou, Radoglou-Grammatikis, Efstathopoulos, Fouliras & Sarigiannidis (2021)]. For example, Anthi *et al.* [Anthi *et al.* (2019)] proposed a three-layered IDS that works with a supervised ML approach to detect a range of popular network-based cyber-attacks on IoT networks. The first layer classifies the type and profiles the normal behavior of each IoT device. The second layer identifies malicious packets on the network during the attack. The third layer classifies the type of the attack. The designed system was evaluated within a realistic smart home testbed consisting of 8 popular commercially available IoT devices. The effectiveness of the proposed IDS architecture was evaluated against 12 popular attacks.

The main shortcoming of supervised learning is the need for a large number of labelled samples, which requires important efforts from domain experts. To address this issue, semi-supervised and unsupervised learning approaches for IDS are prioritized by the research community [Abdel-Basset, Hawash, Chakrabortty & Ryan (2021); Cheng, Xu, Zhong & Liu (2019); Heartfield, Loukas, Bezemskij & Panaousis (2021); Ibrahim, Basheer & Mahmod (2013)]. For example, Ibrahim *et al.* [Ibrahim *et al.* (2013)] employed an unsupervised learning method (Self-Organization Map Artificial Neural Network) to classify the system's input data into normal and abnormal/intrusive instances. Ferdowsi and Saad [Ferdowsi & Saad (2019)] proposed a distributed and privacy-preserving IDS for IoT using a Generative Adversarial Network (GAN). In the considered system model, an intrusion is defined as any activity that causes an IoT device to send data points (packet payload) that do not follow its normal data distribution. They showed

analytically that the proposed distributed GAN provides a higher accuracy, higher precision, and lower false-positive rate.

New techniques, including reinforcement learning, were also proposed in the literature [Apruzzese, Andreolini, Marchetti, Venturi & Colajanni (2020); Heartfield *et al.* (2021); Khan *et al.* (2020); Otoum, Kantarci & Mouftah (2019)]. For example, Heartfield *et al.* [Heartfield *et al.* (2021)] proposed a self-configurable IDS that can autonomously adjust the decision function of the anomaly classification models to a smart home's changing conditions (*e.g.*, new devices, new automation rules, and user's interactions with them). To realize that, the authors employed reinforcement learning on the unsupervised learning methods and model selection based on human presence inferences. Specifically, they applied a probabilistic cluster-based reward mechanism on a non-stationary multi-armed bandit reinforcement learning. The detection model takes cyber and physical sources of data into account. The cyber sources include communication and computation features, such as the frame and packet type, size, IP addresses, and ports; packet time to live (TTL), the mean and standard deviation, the min and max of the sample frequency; content type, size, length, latency, and flow direction. Meanwhile, the physical sources include the audio and radio-frequency signal strength. Khan *et al.* [Khan *et al.* (2020)] proposed a lightweight intrusion identification and prediction scheme for smart home networks using blockchain and a Deep Extreme Learning Machine (DELM) model. In this approach, the data collected from the smart home are first organized in a blockchain to minimize information-related problems such as data repetition, incomplete data values, defects, and noise. Then, the DELM models analyze each chain fragment to spot intrusions and attack patterns. Simulation results showed that the overhead created by the proposed method in terms of distribution, processing time, and energy consumption is marginal compared to the protection and privacy benefits. The proposed method was evaluated using the NSL-KDD and KDD-CUP-99 datasets, where performance improvements were demonstrated compared to Artificial neural network (ANN), Support vector machine (SVM), and Decision tree models.

Regarding the features used in the datasets for training the detection methods, no study has shown that there is a category of features that always offers the best detection accuracy. Therefore,

there is a wide variety of features used in the literature. For example, Jokar and Leung [Jokar & Leung (2016)] presented an IDPS for ZigBee-based home area networks in smart grids, which used six features from the network traffic, including the datagram, the traffic rate, the Received Signal Strength (RSS), the sequence number, the Packet Error Rate (PER), and the node availability. Nobakht *et al.* [Nobakht *et al.* (2016)] proposed an intrusion detection and mitigation framework, called IoT-IDM, to provide network-level protection for smart devices deployed in home environments. The detection module employed a logistic regression model, and the experiment was performed on a realistic Software-Defined Networking (SDN) architecture. In the feature extractor module, the authors mentioned the possibility of using packet-based features, including the source or destination Internet Protocol (IP) addresses, the transport layer protocol type, *e.g.*, Transmission Control Protocol (TCP) or User Datagram Protocol (UDP), the source or destination ports, and the network service. They also mentioned the possibility of using flow-based features, including the duration, the source and destination bytes, the mean fit, and the mean bit. However, in this framework, it is up to the users of IoT-IDM to study and select the appropriate features according to the problem.

2.1.2 Motivation

Most of the innovative works on IDSs include implementing ML methods and demonstrating the efficiency of such approaches. Yet, the proposed solutions require many essential improvements. The fundamental point that we consider in this work is identifying and evaluating the quality of the detection features. To increase the detection accuracy, most of the works in the literature only focus on improving the ML algorithms and do not consider improving the quality of the features used in the dataset. These works use different detection features; however, most papers do not discuss how the proposed detection features are chosen, why each feature is relevant, what type of attacks each feature helps to detect, or why other types of features are not considered in the dataset. Most of the available works just explore a limited subset of features [Anthi *et al.* (2019); Belenko, Chernenko, Kalinin & Krundyshev (2018); Ferdowsi & Saad (2019); Heartfield *et al.* (2021); Illy, Kaddoum, Miranda Moreira, Kaur & Garg (2019); Khan

et al. (2020); Nobakht *et al.* (2016); Shende & Thorat (2020)]. For example, in [Anthi *et al.* (2019)], the authors extracted the totality of the packet header information and performed a dimensionality reduction, concluding that the Internet Protocol (IP) and Transmission Control Protocol (TCP) flags were the most important features. Such an approach provides the best features among packet header features but misses other relevant intrusion detection features, such as content features, time-based and connection-based traffic features, and wireless features. In their experimentation, the high accuracy of the classifier can be explained by the fact that the deployed attacks were not sophisticated. In [Ferdowsi & Saad (2019)], Ferdowsi and Saad considered only the packet payload and detected only the intrusions that cause an IoT device to send data points (packet payload) that do not follow its normal data distribution. Besides that, these works assisted to detect all the malicious behaviors by investigating the packets separately, *i.e.*, without analyzing the traffic sequences where they are involved [Anthi *et al.* (2019); Ferdowsi & Saad (2019)]. Such a limited feature set may not be efficient in detecting some DoS, scanning, probing, and wireless attacks such as jamming. Instead of using only single packet features, in [Jokar & Leung (2016)] network traffic features were added, *i.e.*, features computed by considering the sequences of packets. Overall, the authors used six features, including the datagram, the traffic rate, the Received Signal Strength (RSS), the sequence number, the Packet Error Rate (PER), and the node availability. These traffic features (traffic rate, PER) and wireless communication features (RSS) can enforce detection ability. Unfortunately, this paper did not emphasize on analysis to explore more potential features. Another example is the work presented in [Nobakht *et al.* (2016)]. This paper studied only the specific attacks related to the considered IoT device (hue bulb light) vulnerabilities, without any consideration for other attacks on the smart home network. In the experimental phase, the authors proposed only three detection features, selected heuristically based on the behavior of the Hue bulb light as well as the attack model.

Although it was instructive, the previous works also lack intrusion prevention measures and IDPS deployment architectures. Many of these works focus only on detecting the attacks and do not investigate the reaction mechanisms after attack detection [Anthi *et al.* (2019); Ferdowsi & Saad

(2019); Khan *et al.* (2020); Yamauchi, Ohsita, Murata, Ueda & Kato (2020)]. Moreover, the majority of these works propose a detection model but do not shed light on the process of deployment of the model within the home network architecture [Ferdowsi & Saad (2019); Khan *et al.* (2020); Yamauchi *et al.* (2020)].

2.1.3 Contributions

To overcome the limitations of the current research works, we investigate the properties and nature of smart home security attacks and the quality of features that will be analyzed and employed in ML algorithms to detect each attack efficiently. Moreover, this research also investigates efficient intrusion prevention mechanisms to automatically react and secure the network against any illegal intrusion attempt. Last but not least, a comprehensive SDN-based deployment architecture of the IDPSs within the home network is proposed. Experimental evaluation of the proposed solution is provided using different feature sets, datasets, and ML models. The contributions of this work are listed below.

- First, based on the characteristics of the attacks that may compromise IoT network's security, this work proposes different features that can be pulled out from the smart home network and used in ML methods to improve the accuracy.

- Second, for every attack that may be detected by the smart home IDS, this work introduces different methods and measures to efficiently react and reduce eventual damages.

- Third, this paper proposes a SDN-based deployment architecture of the IDPSs in home networks for effective home security enforcement.

- Finally, the experimental evaluation demonstrates the impact of the proposed detection features on the performance of the ML models.

2.1.4 Organization

The rest of the paper is organized as follows. Section II introduced the studied smart home attacks and the detection features. Section III presents the proposed prevention measures and the deployment architecture. Section IV presents the experimental evaluation of the proposed features on different ML methods and analyzes the obtained results. Finally, Section V contains the conclusion of this work.

2.2 Smart Home Attacks and Detection Features

The first and foremost challenging part of every ML solution is selecting the correct feature set with the highest accuracy. In some applications, it is easier to take all the available features (the initial feature set) and directly apply dimensionality reduction methods to keep the best features (the final feature set). For example, in image processing, it is possible to consider all the image's pixels and avoid the difficult task of studying and selecting relevant features. However, in more complex applications, such as intrusion detection in IoT networks, the initial feature set is not that direct and easy because of the multiplicity and diversity of aspects that constitute the system (physical components, functional organization, configuration, operational principles, procedures, communication protocols, network workload, power consumption, etc.). Therefore, it is necessary to study and select an initial set of relevant features before using dimensionality reduction methods. Such an approach is difficult but convenient since it can clearly explain the necessity of each chosen feature.

In the intrusion detection application domain, the choice of the features to be considered in the ML models substantially depends on the behavior of the attacks and the computer network's characteristics. Therefore, this section first presents the proposed detection features according to the behavior of smart home attacks. Algorithm 4.2 illustrates the methodology adopted for feature selection.

Algorithm 2.1 Feature selection method

Result: Final detection feature set (Fd_Features)
Input 1 : Smart home attacks (Sh_Attacks);
Input 2 : Smart home network features (Sh_Features);
Variable : All selected features (AS_Features) ;
while $attack_x$ *in Sh_Attacks* **do**
    if $feature_y$ *in Sh_Features is affected by* $attack_x$ **then**
        AS_Features = Add (AS_Features, $feature_y$) ;
    end
end
AS_Features = Eliminate_redundant(AS_Features) ;
Fd_Features = Eliminate_correlated(AS_Features) ;

2.2.1 Smart Home Attacks

IoT devices are vulnerable to a wide range of attacks, including DoS, eavesdropping, Remote to Local (R2L), User to Root (U2R), and spoofing attacks. Most of these attacks usually start with a reconnaissance attack.

2.2.1.1 Reconnaissance Attack

The objective is to identify the vulnerabilities in the network. For that, the attacker starts by network scanning operations to discover the network equipment and their IP addresses. Network scanning can be performed using some pings on a range of IP addresses to detect the hosts currently responding on the network. This operation can be realized using a ping sweep utility which identifies viable targets on the network. Once the IP addresses of the viable hosts are known, the attacker begins probing on those hosts to discover the open ports. The network probing can be performed using hacking tools freely available online. After discovering the open ports, the attacker can then figure out the Operating System (OS), the applications, and the services running on the hosts. This is possible since most common applications use well-defined port numbers. Finally, with all the information gathered, the attacker can exploit some known vulnerabilities of the OSs and applications to orchestrate future attacks such as DoS or R2L.

2.2.1.2 Denial of Service Attack

The objective is to deny or delay the user's access to resources (services, devices, networks). For that, the attacker can send a lot of useless requests to the targeted system. He can employ multiple attacking devices to perform a Distributed DoS (DDoS) [Mirkovic & Reiher (2004)], which is more effective and harder to detect. This can overload the system communication bandwidth, the processing resources, and the memory. In such a condition, the legitimate users' requests are delayed or completely dropped. The attacker can also use a radio frequency jammer to compromise the wireless communication channel. Jamming attacks are relatively easy to carry out. Jamming equipment for different frequencies can easily be acquired, and such attack does not require any authentication. The attacker can also launch de-authentication attacks against the wireless devices. For that, he performs a scanning and gets the Basic Service Set Identifiers (BSSIDs) of all the wireless access points in the neighborhood. Then, for every BSSID, the attacker can detect all the devices connected to it. Finally, the attacker selects the devices to jam and sends de-authentication packets to constantly disconnect them from the network.

2.2.1.3 Eavesdropping Attacks

Since wireless transmissions are, for the most part, broadcast in nature, they are extremely vulnerable to eavesdropping attacks. The objective of this attack is to intercept the information exchanged in the network. For that, the attacker employs a device equipped with a wireless network adapter working on promiscuous mode and an eavesdropping software. The attack starts with a scanning operation to detect the available wireless access points. Then, the attacker targets a specific access point and intercepts the packets. Finally, the attacker extracts the information that have no encryption or weak encryptions from all the packets. For example, in WiFi networks secured using Wired Equivalent Privacy (WEP) instead of the new WiFi Protected Access (WPA), the communication can be eavesdropped by an attacker who knows the password of the wireless gateway. This attack can be performed using free software available online, such as CommView for WiFi, Hitchhiker, Aircrack-ng, and Wireshark.

2.2.1.4 Unauthorized User Access

This attack can be a Remote to Local attack (R2L) or a User to Root attack (U2R). The objective of R2L attacks is to access the network from a malicious remote station and gain the same privileges as a legitimate user. Regarding the shared nature of the wireless medium, a malicious user from any location in the distance range of the wireless access point coverage may access the communication channel. Most of the time, the only security mechanism in place is authentication using a password. Social engineering or extra knowledge detained by the illegitimate user can provide him the password. According to the latest top 10 highest security issues released by the Open Web Application Security Project (OWASP), weak and guessable passwords are the first security vulnerability [OWASP IoT Security Team, 2018 (2019)]. For strong passwords, the malicious user can employ packet sniffing tools or brute force attacks to get the password. In home IoT networks, when attackers gain remote access to victims' networks, they can imitate existing local users, *i.e.*, connect to IoT applications where private device data are displayed, and send user commands to the IoT devices (*e.g.*, open doors, shutdown lights, turn off cameras, etc.). The intruders can also escalate the attacks to a U2R via stolen credentials or Rootkits. In U2R attacks, the intruders try to obtain privileges normally reserved to the network administrator. With such privileges, the intruders can take full control of the network. In home IoT networks, when attackers gain administrator access, they can disconnect IoT devices from the servers, uninstall IoT applications, inject false data or delete information in the databases, and install malware. Even though R2L and U2R are rare compared to scanning, probing and DoS, these occasional attacks may compromise the entire smart home network.

2.2.1.5 Spoofing Attacks

The attackers often perform spoofing (also known as node forgery or impersonation) before launching any other malicious action. The objective of spoofing is to hide the intruder's identity, especially its IP address, such that the source of the attacks cannot be detected. For that, the attacker starts by identifying the trusted IP addresses of the targeted system. He then modifies his IP packets header and replaces the real IP address with a forged (spoofed) source IP address to

impersonate a trusted device. Finally, the attacker sends the malicious packets, and the targeted system sees them as originating from a trusted device. Systems implementing IP address-based authentication are easily compromised. Moreover, systems employing packets filtering to thwart DoS can be easily deceived by IP spoofing as packets with spoofed IP addresses are more difficult to filter. Identity spoofing allows the attacker to perform many other malicious operations, including MITM attack. For this, more identity information, including cryptographic credentials, are captured by the adversary and used in the impersonation.

2.2.2 Proposed Detection Features

2.2.2.1 Traffic Features

Certain types of attacks, such as reconnaissance (scanning, probing), and DoS, involve many connections in the targeted system in a short period of time. These attacks can be detected using traffic features, and especially time-based features. Traffic features are statistical data computed from the traffic behavior. Time-based features are traffic features computed with respect to a predefined interval of time. An example of a time-based feature includes the number of connections in the past 5 seconds that have the same destination host as the current connection. Five seconds is the predefined time interval. Since time-based features are efficient in detecting scanning and probing, some attackers use a much larger time interval than the predefined time set for the IDS, *e.g.*, one connection in every 10 seconds. In this context, the attacks will be less suspicious. This limitation can be addressed by employing additional traffic features, especially connection-based features, which are computed based on a window with a predefined number of connections. An example of connection-based features would be the number of connections in the past 50 connections that have the same destination host as the current connection. In this example, 50 connections are the connection window interval. Table 2.2 presents some traffic features used in the NSL-KDD [KDDCUP (1999); Stolfo, Fan, Lee, Prodromidis & Chan (2000); Tavallaee, Bagheri, Lu & Ghorbani (2009)] datasets.

Table 2.2 Examples of traffic features used in NSL-KDD dataset

Feature	Description
count	Number of connections to the same host as the current connection in the past 2 seconds
The following features refer to these same-host connections	
serror_rate	% of connections that have “SYN” errors
rerror_rate	% of connections that have “REJ” errors
same_srv_rate	% of connections to the same service
diff_srv_rate	% of connections to different services
srv_count	Number of connections to the same service as the current connection in the past 2 seconds
The following features refer to these same-service connections	
srv_serror_rate	% of connections that have “SYN” errors
srv_rerror_rate	% of connections that have “REJ” errors
srv_diff_host_rate	% of connections to different hosts

2.2.2.2 Packet Header Features

The packet header features are the fields that compose each networking layer (Physical, Data Link, Network, Transport, and Application). Examples of fields include the source and destination IP addresses, TCP source and destination port numbers, network service, IP and TCP flags, packet and frame length. Most intrusive activities, including port scanning, probing, SYN flood, MITM, IP fragmentation attacks, indicate unusual values in the packet header. For example, scanning and DoS mostly use modified TCP flag, invalid or improper settings. Moreover, specific TCP flag responses such as TCP SYN check and TCP SEQ check can indicate a MITM attack. Therefore, these attacks can be detected using packet header features [Anthi *et al.* (2019)]. For example, Internet Control Message Protocol (ICMP) code options, namely fragment protection and packet protection, can be used to identify a DoS attack, whereas the IP flags can detect UDP, ICMP, and TCP fragmentation attacks. The TCP destination port can indicate a port scanning. The packet length is also an indicator of malicious behavior, especially when it is significantly larger or smaller than usual. Table 2.3 presents some packet header features used in the Anthi 2019 IoT dataset [Anthi *et al.* (2019)].

Table 2.3 Examples of packet header features used in Anthi 2019 IoT dataset

IP header	**TCP header**	**Other**
ip.flags	tcp.dstport	caplen
ip.flags.df	tcp.flags.ack	frame.cap_len
ip.flags.mf	tcp.flags.push	frame.len
ip.frag_offset	tcp.flags.syn	len
ip.ttl	tcp.stream	icmp.code

2.2.2.3 Content Features

Unlike scanning, probing, and DoS attacks, unauthorized user access attacks, such as R2L and U2R, sometimes involve only a few connections. Therefore, intrusion detection models built on traffic features only struggle to detect such attacks. To detect these attacks, content features are a reliable option. Content features are data embedded in the packet contents that can indicate suspicious behaviors. An example of a content feature is the number of failed login attempts. To extract these features, the data portions of the packets must be inspected. This presents many challenges. The first major issue is the system's scalability since billions of candidate content features are present in the payloads. This requires big data techniques to handle this overwhelming amount of data [Abusitta, Bellaiche, Dagenais & Halabi (2019); Zhong, Yu & Ai (2020)]. This approach also suffers from encryption or payload obfuscation (achieved by polymorphism and metamorphism). To utilize these features, the application data needs to be reassembled, and domain knowledge can be essential. In end-to-end encrypted communications, the packet payload can only be decrypted on the communicating hosts and not in the network. This requires measurements at all hosts in the network [Iglesias & Zseby (2015)]. Moreover, network packet payload analysis is an active research area, and many solutions have been proposed [Wang, Zhu, Wang, Zeng & Yang (2017); Xu, Chen, Su, Yiu & Hui (2016)]. Table 2.4 presents some content features used in the NSL-KDD datasets [KDDCUP (1999); Stolfo *et al.* (2000); Tavallaee *et al.* (2009)].

Table 2.4 Examples of content features used in NSL-KDD dataset

Feature	Description
hot	number of “hot indicators”
num_failed_logins	number of failed login attempts
logged_in	1 - successfully logged in; 0 - otherwise
num_compromised	number of “compromised” conditions
root_shell	1 - root shell is obtained; 0 - otherwise
su_attempted	1 - “su root” command attempted; 0 - otherwise
num_root	number of “root” accesses
num_file_creations	number file creation operations
num_shells	number of shell prompts
num_access_file	number of operations on access control file
num_outbound_cmds	number of outbound commands in a ftp session
is_hot_login	1 - the login belongs to the “hot” list; 0 - otherwise
is_guest_login	1 - the login is a “guest”login; 0 - otherwise

2.2.2.4 Wireless Communication Features

Wireless attacks can be performed at any distance within the wireless communication coverage. In these attacks, the attacker’s device is likely to be positioned at an unusual distance within the network. For example, in home networks, an illegitimate user may need to attack from outside the house perimeter. In wireless communication, the transmitter’s distance can be estimated using the Received Signal Strength (RSS). Therefore, for active wireless attacks such as jamming, de-authentication, spoofing, and contaminating, the radio transmitter’s distance can be computed and used as a detection feature, especially in static networks like smart homes. However, only employing the distance will not detect the attacks that are performed from a usual distance. Moreover, the locations of legitimate devices can eventually be changed, causing a disturbance of false positives.

This limitation can be mitigated by employing additional wireless communication features, especially radio-frequency fingerprints. Radio-frequency fingerprinting is the process that identifies a radio transmitter by its unique signal transmission characteristics, named as fingerprint, determined by unique circuit characteristic. These circuit characteristics, which are measurable, aggregate the differences introduced during transmitter manufacturing [Tian,

Lin, Guo, Wen, Fang, Rodriguez & Mumtaz (2019)]. The radio-frequency fingerprint can be extracted and used in wireless networks as a feature to detect wireless attacks, especially identity spoofing and cloning attacks. A cloned device will have the same numeric equipment identity, may be in a usual distance inside the network, but will have a different radio fingerprint.

Nevertheless, this feature also has some limitations as the recognition rate depends on the Signal to Noise Ratio (SNR). For example, in the simulation results shown in [Tian *et al.* (2019)], for a 5dB SNR, the recognition rate can be lower than 90%. Moreover, the use of radio-frequency fingerprinting alone cannot detect a compromised device, *i.e.*, a legitimate node that is controlled by a malware. More features can be considered, including the energy profile and the cyclostationary behavior of the network. Table 4.4 recapitulates all the proposed detection features.

Table 2.5 Proposed detection features

Attacks	**Detection features**
Scanning, Probing, DoS	Traffic features (Time-based and Connection-based features), Packet header features
U2R, R2L	Content features, Packet header features
Wireless network attacks: Jamming, Deauthentication, Spoofing, Contamination	Distance of the radio transmitter, Radio-frequency fingerprint, Received Signal Strength (RSS), Signal to Noise Ratio (SNR), energy profile, Packet header features

Taken alone, none of the detection features is sufficient to efficiently detect all the threats. These features are not self-sufficient but rather dependant and complementary. They must be combined and used in the training of the ML models. Many ML algorithms have demonstrated excellent capabilities in building accurate models. This includes basic ML, Deep Learning (DL), and ensemble methods. There is no specific learning algorithm that always induces the most accurate learner. Therefore, it is necessary to try different ML methods and deploy the one that produces the best detection model. This approach results in a more advanced intrusion detection model for IoT networks in general and especially for smart home networks. An IDS built from such an approach will be able to detect zero-day attacks. For example, the connection-based and

time-based features used in the training process can help the detection model recognize zero-day DoS attacks. Likewise, the employed radio-frequency fingerprinting features can enable the detection of zero-day spoofing attacks.

2.3 Proposed Prevention Measures and Deployment Architecture

This section illustrates the prevention measures to take when an attack is detected and the proposed deployment architecture of the IDPSs in home networks.

2.3.1 Proposed Prevention Measures

The Intrusion Prevention System (IPS) function is to take defensive measures when the IDS detects an attack. These measures are the set of actions that are capable of stopping or mitigating the intrusion. Some preventive measures are implemented in the IPS to automatically launch the required defensive operations once an attack is detected. This significantly reduces the latency in aborting attacks and the eventual damages as compared to send an alert and then wait for human intervention. However, using an IPS does not exclude the implementation of an alert system. The IPS can focus on urgent and temporary defensive measures, and the alert can be sent to the security administrator in order to apply complementary and more precise prevention mechanisms. The primary prevention measure is to drop the malicious packets and block the attacker's IP and MAC addresses. This can eliminate attacks, including R2L and U2R. It can also reduce scanning, probing, and DoS attacks [Tian, Ji, Liu, Liu, Zhai, Dai & Huang (2020)]. The intruder connection can also be redirected to a Honeypot to study the behavior of the attacker. This prevention measure can easily be implemented in SDN based network.

For jamming attacks, the intended damages can be alleviated by migrating the communication to a different frequency band [Navda, Bohra, Ganguly & Rubenstein (2007)]. For that, Dynamic Channel Selection (DCS) can be employed. The DCS allows the wireless access point to monitor the noise levels on the channel on which the access point is currently operating. When the noise levels exceed the predefined DCS thresholds, the wireless access point ceases operating on the

current channel. It employs Auto Channel Selection (ACS) to select an alternate channel to operate on. Another measure to prevent jamming attacks consists of increasing the robustness of the legitimate signal to restore a secured wireless communication [Pelechrinis, Broustis, Krishnamurthy & Gkantsidis (2011)]. Furthermore, position location technologies can be employed to track down the offending station and remove it. Such technologies can be used against all active wireless attacks, including jamming, contamination, and spoofing attacks. Some WiFi equipment manufacturers have already added similar technologies to their products [Aldosari, Zohdy & Olawoyin (2019); Wang, Liang, Wei & Fan (2018)].

2.3.2 Proposed Deployment Architecture

This research work adopts SDN and Network Functions Virtualization (NFV) architectures to provide an effective implementation of the IDPS in home IoT networks. Figure 2.1 illustrates the proposed SDN-driven home IoT network architecture. In this architecture, the IoT devices are vendor-neutral and use different wireless communication protocols, including WiFi, ZigBee, and Z-waves. Some IoT devices may not support an agent deployment due to resource constraints or proprietary firmware constraints. The wireless access points provide different wireless communication protocols. The wireless gateways are connected to the home router via wired communication (ethernet). The home router supports SDN data-plane functionalities. The SDN controller, the network, and the security applications(IDS and IPS) are deployed in a local server. The IoT applications are deployed on the user devices (smartphone, tablet, laptop), the local server, or cloud servers.

2.4 Experimentation

This section presents the experimentation and analyzes the results obtained. The goal of the experiment is to evaluate the impact of each feature set (packet header, content, traffic, and wireless features) on the performance of the ML models. However, there is no existing dataset that contains all the studied features and all the studied attacks. Every dataset contains only a

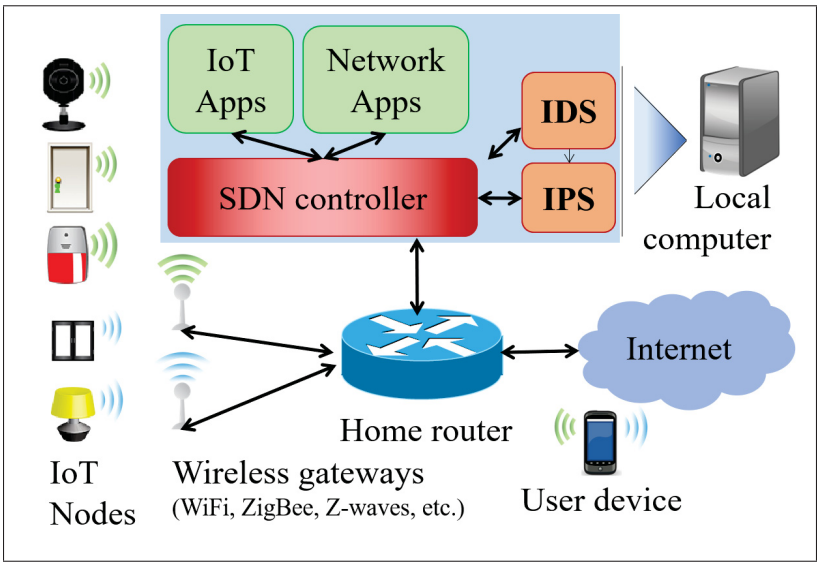

Figure 2.1 The proposed home IoT network architecture

subset of the presented features and attacks. Thus, we chose two different datasets to perform this experimentation.

The first dataset, presented in [Anthi *et al.* (2019)], is built from a realistic smart home testbed consisting of 8 popular commercially available IoT devices, connected to an access point via WiFi and Ethernet communications, and a laptop to record the network traffic. The dataset contains 3 weeks worth of benign data and 2 weeks of malicious data collected from the access point using the tcpdump tool. The malicious data was generated by performing 12 popular attacks against this smart home testbed. The attacks were classified into 4 categories, namely scanning, DoS, iot-toolkit, and MITM. After data cleaning and dimensionality reduction, the dataset is composed of 220786 packets, each represented by 10 features, as shown in Table 2.3.

The second selected dataset is NSL-KDD, which is the most complete and realistic dataset available. NSL-KDD includes a total of 39 specific attacks grouped into 4 different attack categories, namely, probing (Probe), R2L, DoS, and U2R. Table 2.6 presents the attacks that were involved in this dataset. The training dataset (KDDTrain+) and testing dataset (KDDTest+) contain 125973 and 22544 number of single connection vectors, respectively. Each connection vector is represented by 41 features and labelled as either normal or attack, with exactly one

specific attack type. The features involve 9 intrinsic features (packet header features), 19 time-based and connection-based traffic features, and 13 content features. Table 2.2, Table 2.4, and Table 2.7 present these features.

Table 2.6 Attacks involved in the NSL-KDD dataset

Attack Type	**Attacks**
Probe	Ipsweep, nmap, portsweep, satan, saint, mscan
DoS	Back, land, Neptune, pod, smurf, teardrop, mailbomb, udpstorm, apache2, processtable
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, named, snmpguess, worm, snmpgetattack, xsnoop, xlock, sendmail
U2R	buffer_overflow, loadmodule, perl, rootkit, xterm, ps, sqlattack, httptunnel

Table 2.7 Intrinsic features used in NSL-KDD dataset

Feature	**Description**
duration	Length (number of seconds) of the connection
protocol_type	Type of the protocol, *e.g.*, tcp, udp, etc.
service	Network service on the destination, *e.g.*, http, telnet, etc.
src_bytes	Number of data bytes from source to destination
dst_bytes	Number of data bytes from destination to source
flag	Normal or error status of the connection
land	1 - connection is from/to the same host/port; 0 - otherwise
wrong_fragment	Number of “wrong” fragments
urgent	Number of urgent packets

This experiment employs six different ML algorithms to build detection models. These algorithms include basic methods as well as ensemble methods. The basic methods include Decision Tree (DT) and K-Nearest Neighbour (KNN) classifiers. The ensemble methods include Random Forest (RF), Bagging, AdaBoost, and Voting Classifiers. The goal here is not to compare the performance of different ML methods but to analyze their performance on different feature sets. The implementation is performed using the Python programming language and Scikit-learn(free ML library).

The experiment is divided into two phases. The first phase employs the two datasets to analyze the impact of the employed features. Therefore, we build the detection models, first on the smart home dataset, which contains only packet header features, then on the NSL-KDD dataset using only the packet header features, and finally on the NSL-KDD dataset using all the features. Figure 2.2 and Figure 2.3 present the accuracy scores of the detection models employed on the smart home and NSL KDD datasets, respectively.

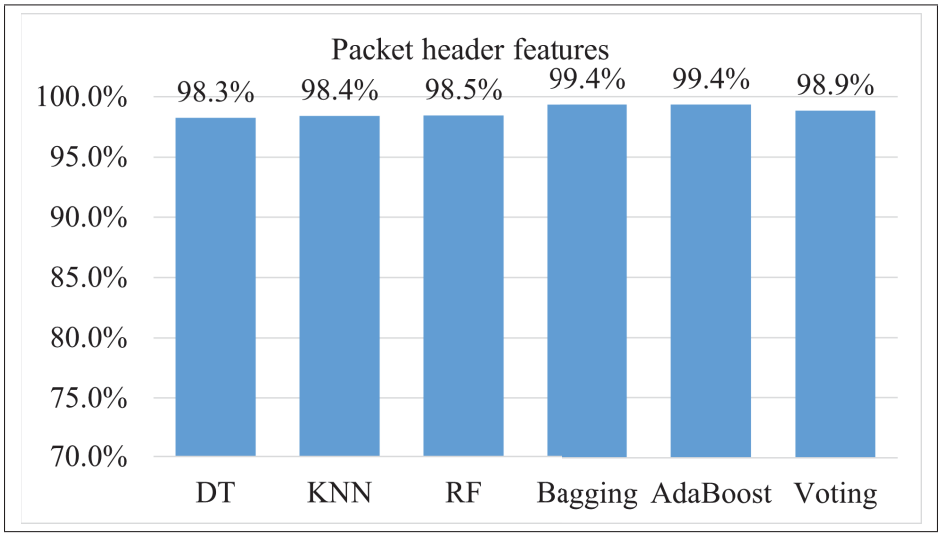

Figure 2.2 Prediction accuracy using the smart home dataset

The results of the experiment on the smart home dataset (Figure 2.2) show that the packet header features single-handedly can achieve a detection accuracy of more than 98% . In contrast, with the NSL-KDD dataset (Figure 2.3), the results show that the models built only from packet header features show lower accuracy than those built with all the features. The high accuracy with the smart home dataset can be explained by the fact that, in this dataset, the attacks involved were not sophisticated and do not represent realistic detection challenges. However, in the second dataset, the attacks involved were complex, and their detection required advanced detection features. This demonstrates that using only the packet header features is not enough to build accurate intrusion detection models. Therefore, to build more robust intrusion detection models, more features, such as content features, time-based, and connection-based traffic features are necessary.

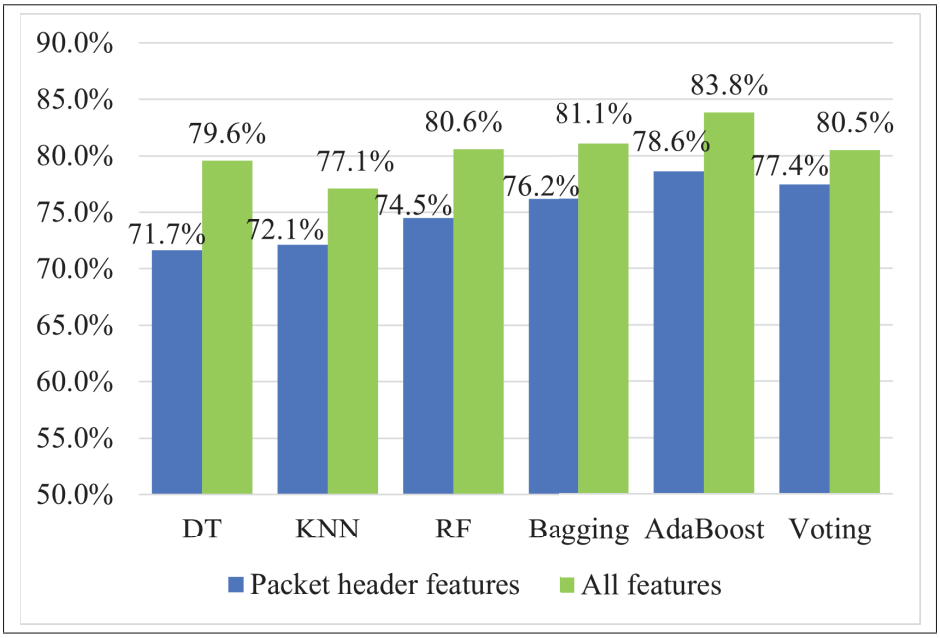

Figure 2.3 Prediction accuracy using the NSL-KDD dataset

In the second phase of the experiment, the impact of traffic and content features on the detection of various attacks is demonstrated. This phase is divided into two parts. The first part focuses on the detection of R2L and U2R attacks. Where, we build two different models for each algorithm, one exploiting traffic features and the other exploiting content features. Table 2.8 presents the accuracy of each model. In these results, according to content features, the best model achieves an accuracy of 80.26%, while the accuracy of the best model from traffic features is only 76.47%. Moreover, for each ML method, the model built from content features provides better accuracy. This demonstrates that to detect R2L, U2R, and similar attacks, content features are more effective than traffic features. Therefore, we have added content features to the packet header features and reconstructed all the models. Figure 2.4 presents the accuracy of the resulting models.

Now the best model achieves an accuracy of 85.21% instead of 80.26%. The performance has been significantly improved by providing new essential features to the same ML algorithms. These results confirm that to detect U2R and R2L attacks, the accuracy of the detection models can be improved by adding appropriate content features to the smart home dataset as

Table 2.8 Detection of R2L and U2R attacks using traffic and content features

ML-Methods	R2L & U2R	
	Traffic feature	**Content features**
Decision Tree	75.74 %	76.84 %
KNN	76.47 %	76.74 %
Random Forest	74.68 %	76.88 %
Bagging	75.51 %	77.03 %
AdaBoost	75.44 %	80.26 %
Voting	75.52 %	76.84 %
Best	**76.47 %**	**80.26 %**

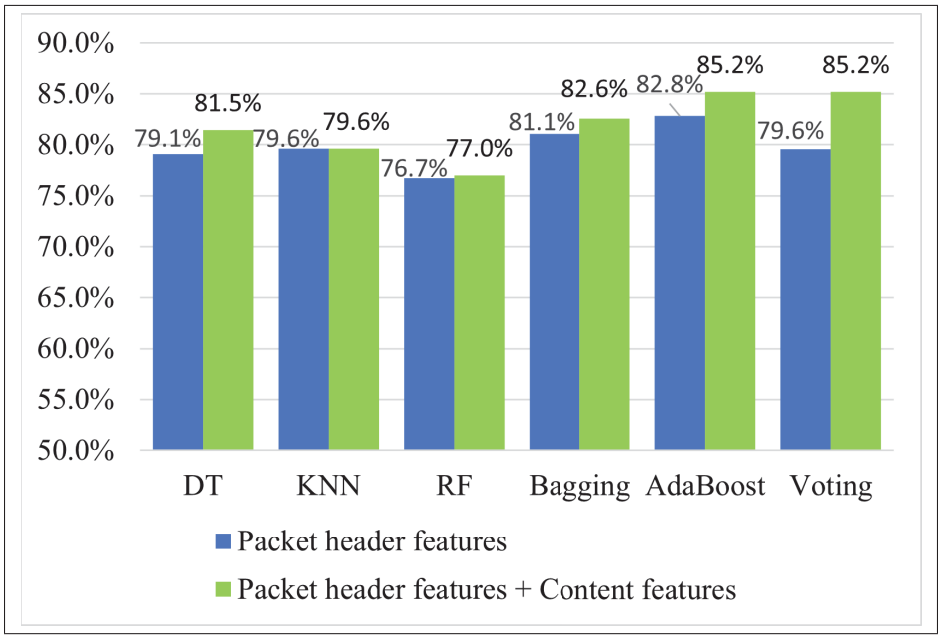

Figure 2.4 R2L and U2R prediction accuracy using packet header features and content features

complementary features. The model's performance can be further improved by employing complete features set, as proposed in this research paper.

The second part of the experimentation focuses on the detection of DoS and probe attacks. Once more, the same ML algorithms are employed to build detection models. For each algorithm, we

developed two different models, one is exploiting traffic features and the other exploiting content features. Table 2.9 presents the accuracy of each model.

Table 2.9 Detection of probe and DoS attacks using traffic and content features

ML-Methods	**Probe & DoS**	
	Traffic feature	**Content features**
Decision Tree	79.51 %	71.38 %
KNN	80.35 %	72.78 %
Random Forest	81.29 %	71.38 %
Bagging	82.59 %	71.38 %
AdaBoost	82.58 %	72.89 %
Voting	80.63 %	73.33 %
Best	**82.59 %**	**73.33 %**

For Probe and DoS attacks, for traffic features, the best model achieves an accuracy of 82.59%, while the accuracy of the best model from content features is only 73.33%. This significant difference in performance can also be noticed for each ML method. The models built from traffic features provide higher detection accuracy than the models built from content features. This demonstrates that to detect probing, DoS, and similar type of attacks, traffic features are more effective than content features. Therefore, we added the traffic features to the packet header features and reconstructed all the models. Figure 2.5 presents the accuracy of the resulting models, where the best model achieves an accuracy of 87.78%. A significant improvement in performance is observed by providing new essential features to the same ML algorithms. These results confirm that, by adding traffic features to smart home dataset as complementary features, their performance against Probe and DoS attacks can be enhanced. The model's performance can be further improved by employing a more exhaustive feature set, as proposed in this work.

This experiment demonstrates that selecting a more adequate feature set provides more efficient detection models. It showed that the completeness of the dataset determines the precision of the detection model and its ability to identify attacks. Therefore, the performance of the detection model can be improved, not only with new ML methods but, importantly, with better detection

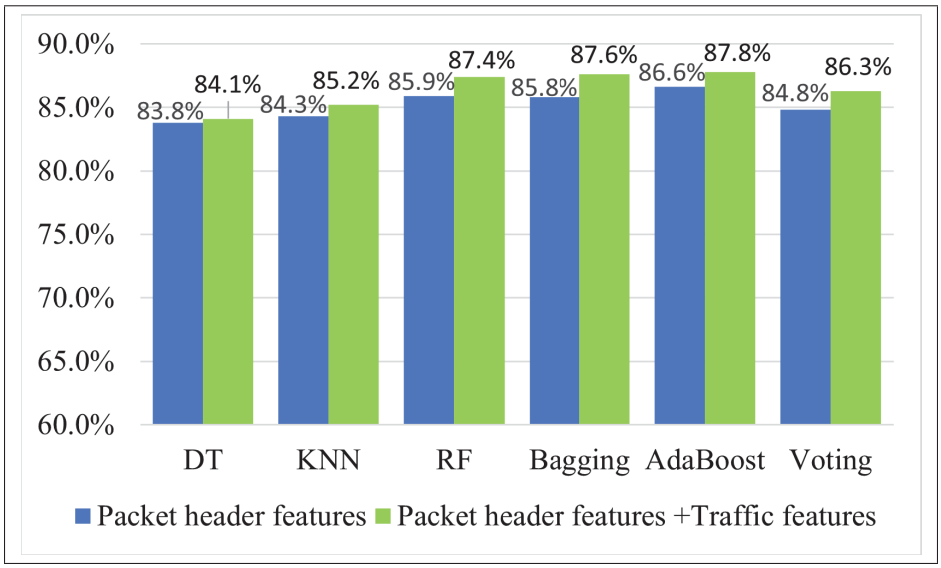

Figure 2.5 Probe and DoS Prediction accuracy using packet header features and traffic features

features. However, trying to detect attacks that are not explicitly studied when extracting the features will result in decrements in the performance of the detection model.

2.5 Conclusion

Cyber-attacks constitute major threats to smart homes' security and privacy. To address this concern, ML and SDN-based IDPSs are emerging as promising technologies. This paper identified various features that can improve the performance of ML-based IDSs. Specifically, in addition to the packet headers, time-based and connection-based traffic features have been proposed against scanning, probing, and DoS attacks. Then, different content features have been proposed to enhance the detection of unauthorized user access, such as U2R and R2L. Furthermore, to detect attacks that exploit the wireless communication channel, such as jamming, de-authentication, spoofing, and contamination attacks, more features are discussed, including the distance from the radio transmitter, radio-frequency fingerprint, received signal strength, signal to noise ratio, and the system's energy profile. This contribution can be a fundamental guideline for research projects interested in building datasets for ML-based IDS for IoT networks.

It provided an in-depth analysis of several features that can be added to the dataset to detect attacks, including well-known attacks and zero-day attacks. Moreover, various prevention measures that can be implemented in the IPS to defend the smart home network against attacks are proposed. Finally, we presented a SDN-based deployment architecture of the IDPS in the home network. This deployment architecture presents the process to optimally deploy a real-time IDPS in a smart home network. The experimentation conducted in this paper has demonstrated the predominant role of the detection feature set on the ML models' performance. The results showed that the accuracy of the detection model depends more on the feature sets than the ML methods. For future work, implementing SDN-driven collaborative IDPSs for mission-critical IoT networks like smart factories is envisioned.

CHAPTER 3

A HYBRID MULTISTAGE DNN-BASED COLLABORATIVE IDPS FOR HIGH-RISK SMART FACTORY NETWORKS

Poulmanogo Illy [a], Paulo Freitas de Araujo-Filho [a], Kuljeet Kaur [a] and Sahil Garg [b],

[a] Département de Génie Électrique, École de Technologie Supérieure,
1100 Rue Notre-Dame Ouest, Montréal, Québec, H3C 1K3, Canada
[b] Resilient Machine Learning Institute (ReMI), Montreal, Canada.

Paper published in *IEEE Transactions on Network and Service Management*, Vol. 19, Issue. 4, pp. 4273 - 4283, August 2022

3.1 Introduction

Industrial control systems (ICSs) have entered a new era of modernization enabled by the internet of things (IoT). Traditional stand-alone ICSs, which used to be isolated from information technology (IT) networks, are being transformed into smart ICSs that enable high-level process supervisory management using information and communication technology (ICT). Connected sensors, actuators, and IoT devices make it possible to easily monitor and control manufacturing equipment (pumps, valves, compressors, tanks, etc.) and manufacturing conditions (air quality, humidity, temperature, etc.) remotely using computers, tablets, or smartphones. However, a factory's industrial internet of things (IIoT) introduces significant security problems into the factory network because of the security vulnerabilities of IoT devices and software that hackers can exploit to carry out malicious activities.

In highly sensitive industrial facilities that require absolute security, safety, and privacy to manipulate dangerous chemicals and machinery, a successful cyber-attack could have major consequences, including worker injury or death, long-term environmental impacts, and substantial economic losses. For example, in May 2021, in the United States of America (USA), a ransomware attack was able to take control of Colonial Pipeline Inc.'s ICS, causing the facility to shut down its 5,500-mile natural gas pipeline for five days and resulting in more than 10,000 gas stations being out of fuel. The company had to pay a $4.4 million ransom to restore control of its system

[Alvee, Ahn, Kim, Su, Youn & Ryu (2021)]. These attacks could become constant threats and effective cyber-terrorism tactics. Therefore, in critical industrial domains, such as electric utility companies, fuel facilities (oil refineries, gas pipelines), and other dangerous-chemical plants, it is imperative to implement advanced security solutions that prioritize security over any other productivity factor.

A variety of security and safety approaches are implemented to protect these environments [Fan, Fan, Wang & Zhou (2015); Stoessel (2021)]. One such approach is intrusion detection systems (IDSs), which focus on identifying malicious attempts (successful or unsuccessful) to penetrate a network. Existing intrusion detection methods include specification, signature, and anomaly based detections, with the most commonly implemented techniques being the shallow machine learning and deep learning models. However, each of these methods has some inherent limitations that need to be addressed [Wang & Jones (2017)].

Signature-based detection, also known as misuse detection, relies on a signature database to discover attacks [Kemmerer & Vigna (2002)]. The signatures are created from rules that match the patterns of known malicious activities. Then, detection is performed by comparing the current network traffic with all the signatures in the database. When a match is found, the corresponding traffic is reported as an intrusion. The main advantage of signature-based detection systems is that they typically produce fewer false positives. On the other hand, their drawback is their inability to detect new or previously unknown attacks. A new attack goes undetected if there are no signatures matching the pattern of the attack on the ICS. Therefore, the signature database must be kept up to date. For every new attack discovered, a new signature must be created and added to the signature database, which is a labour-intensive process. Furthermore, it is very challenging to develop appropriate signatures for sophisticated attacks that are evolved from versions of previous attacks [Kim & Aminanto (2017); Zhong *et al.* (2020)].

Specification-based IDSs rely on a set of predetermined specifications, such as the maximum capacity of links, packet size, functional states and transitions, operational and protocol constraints, and statistical rules that define a system's legitimate behavior model [Jokar & Leung

(2018)]. If the characteristics of an action deviate from the specifications, an alert is issued. Since this type of approach detects deviations from normal behaviors, it can detect zero-day attacks. However, the specification rules are manually defined by human experts and strongly depend on the expertise of the network administrator. Manually defining such rules can be task-intensive in large networks, which poses a scalability issue for ICSs while increasing the likelihood of human errors, leading IDSs to produce false positives. Moreover, incomplete specifications can cause an IDS to miss benign-looking attacks and cause false negatives, which represent a considerable risk to ICS security. Furthermore, manually defined specifications may not adapt to different environments and may be incompatible with some proprietary operating systems, applications, or protocols.

Unlike the previous two methods, in which specifications and signatures must be manually predefined, anomaly-based IDSs that employ machine learning techniques can automatically learn from samples of malicious and benign network traffic. Nevertheless, the fundamental challenge of this approach is designing efficient learning models that are accurate enough to minimize the number of false alarms. Shallow learning models, which are typically built with less than three computational layers, encounter difficulties discovering useful intrusion patterns from an increasing number of training samples [Gupta, Tanwar, Tyagi & Kumar (2020); Hodo, Bellekens, Hamilton, Tachtatzis & Atkinson (2017); Zhong *et al.* (2020)]. Therefore, deep learning models, which are built with a large number of neural layers and produce better results than shallow learning models in fields like computer vision, speech recognition, and finance [Barra, Carta, Corriga, Podda & Recupero (2020); Chhikara, Singh, Tekchandani, Kumar & Guizani (2021); Miglani & Kumar (2019); Nassif, Shahin, Attili, Azzeh & Shaalan (2019)], have inspired many studies in the field of intrusion detection.

Although most existing solutions rely on a single deep learning model, they usually work well only when malicious activities do not have complex patterns from multiple types of attacks and when the number of samples that each class in the dataset is represented by is not highly imbalanced. However, intrusive attacks are increasingly varied and sophisticated due to new attack strategies continually being created. Each type of attack has its own characteristics,

strategy, and data distribution. As a result, the complexity of attack classification is increasing significantly, and the learning task is becoming more and more challenging. Moreover, only a small number of samples exist for some types of attacks, and it is difficult for the deep learning model to learn the patterns of these attacks [Dua & Du (2016); Zhong *et al.* (2020)]. In these conditions, the single deep learning model approach may experience problems understanding and predicting all classes with high accuracy. Even after complex hyper-parameter tuning of the deep neural networks (DNNs), misclassifications are common.

The goal of our work is to develop an approach that learns complex patterns amid increasingly complicated data distribution scenarios and imbalanced classes to provide more accurate detection models that satisfy the requirements of high-risk ICSs. Therefore, we took inspiration from boosting methods (*e.g.*, AdaBoost) used with shallow machine learning algorithms [Zhang & Yang (2018)], and propose a hybrid multistage DNN in which stage DNNs are built sequentially to work on the limitations of the previous stages. The learning phase of the proposed approach starts with a conventional DNN detection model that is trained and validated using the learning dataset. Then, each sample that is misclassified or classified with low decision confidence is added to a new dataset that is used to train a next stage DNN. The process is repeated until the number of misclassified samples falls below a predefined threshold. Figure 3.1 provides an overview of this learning process. In the evaluation phase, for each sample in the test dataset, the individual decisions of every DNN that constitute the stages are used in combination functions to compute the multistage DNN-based IDS's decision. The hybrid multistage DNN is built by using a different DNN algorithm or architecture during each stage. An experimental evaluation of the proposed approach is conducted using different datasets, and the results confirm that it improves the detection accuracy more effectively than previous methods. Moreover, we also propose a collaborative intrusion prevention system (IPS) with an emergency response schema to hinder the attacks automatically and as soon as they are detected. The schema performs anomaly detection first, which reduces ICSs' attack response delay, then proceeds with attack classification for security enforcement. Furthermore, this study proposes a software-defined networking (SDN)-based architecture to deploy the IDPS in ICSs' networks.

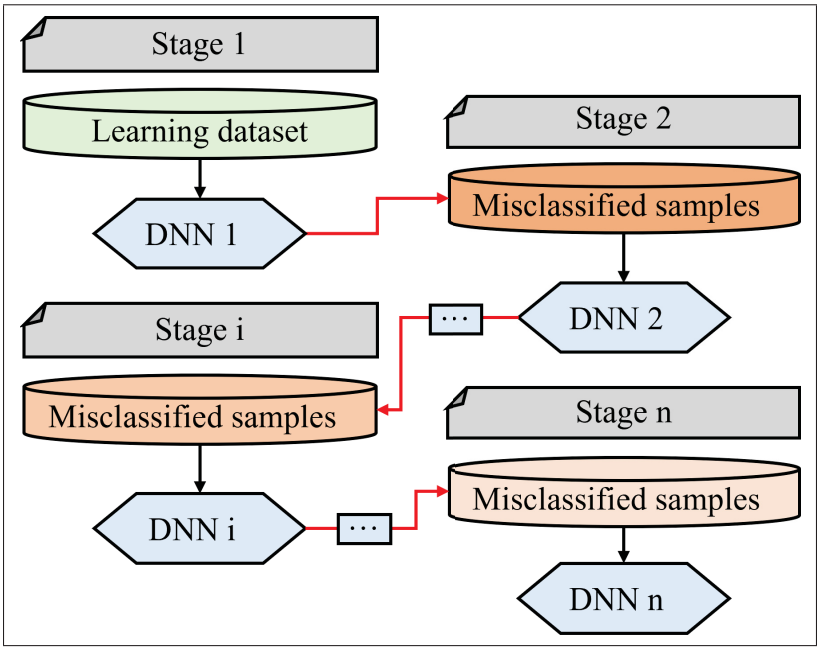

Figure 3.1 Overview of the multistage DNN learning process

3.1.1 Contributions

The main contributions of this work are listed below.

- First, with regard to the security requirements of high-risk ICSs, this study proposes a hybrid multistage DNN model that provides higher intrusion detection accuracy than existing solutions.
- Second, a collaborative intrusion response mechanism is proposed with an emergency reaction mechanism to minimize attacks' impact on ICSs.
- Third, an SDN-based architecture is proposed to deploy the IDPS in an ICS.
- Finally, an experimental evaluation is conducted using different datasets and demonstrates the effectiveness of the proposed approach.

3.1.2 Organization

The rest of this research paper is organized as follows. Section II presents related works and discusses limitations. Section III presents the methodology of the hybrid multistage DNN detection model. The collaborative intrusion response schema and the deployment architecture are presented in Section IV. Section V presents the experimental evaluation and discusses the results obtained. Finally, Section VI concludes this work.

3.2 Related Works

There exist several research works looking into using machine learning-based IDSs to secure ICSs. For example, Beaver *et al.* [Beaver, Borges-Hink & Buckner (2013)] applied machine learning algorithms to detect malicious remote terminal unit (RTU) communications. They used a labelled RTU telemetry dataset from a gas pipeline system in Mississippi State University's Critical Infrastructure Protection Center (USA). The attack traffic generated in that dataset included variants of command injection and data injection attacks. The authors implemented six different machine learning algorithms, namely, naive Bayes, random forests, one rule, J48 (a type of decision tree), non-nested generalized exemplars, and support vector machines. They demonstrated these learning methods' ability to detect the attacks in question. Lin *et al.* [Lin, Wu & Lee (2017)] developed an ICS test bed where they performed different penetration attacks, such as reconnaissance, response injection, command injection, and denial of service (DoS) attacks. The authors collected records of normal activity and the attack activity. Then, they used features from the data link and application layers to implement a machine learning IDS model that successfully detected all the test attacks performed. However, the authors did not discuss the machine learning algorithm used and proposed to focus on the machine learning phase in future research. Teixeira *et al.* [Teixeira *et al.* (2018)] studied the detection of reconnaissance attacks in ICS environments. They built a real-world test bed to conduct attacks and generated a dataset. The authors implemented different detection models using various machine learning methods, such as decision tree, random forest, naive Bayes, logistic regression and k-nearest neighbors (KNN). Ullah and Mahmoud [Ullah & Mahmoud (2017)] implemented an information

gain-based feature selection filter to reduce the time and computational complexity of machine learning-based IDSs in supervisory control and data acquisition (SCADA) networks. They used the reduced feature set to build anomaly detection and attack classification models based on J48 and Bayesian network learning algorithms. These models were evaluated using an ICS dataset developed at Mississippi State University's Distributed Analytics and Security Institute. Illy *et al.* [Illy *et al.* (2019)] combined multiple shallow learners to build different ensemble learners, including multiexpert and multistage models. They conducted experimentations and demonstrated that, when using hard-to-classify datasets, ensemble learners produce more accurate detection systems than single shallow learners.

To overcome the limitations of shallow machine learning methods, Yang *et al.* [Yang, Cheng & Chuah (2019)] proposed a deep learning-based network detection model for SCADA systems. Their proposed approach employed convolutional neural network (CNN) algorithms to characterize salient temporal patterns in SCADA traffic and identify time windows in which network attacks are present. They used realistic SCADA traffic datasets, and their experimental results showed that the proposed deep learning-based approach achieves higher detection accuracy and could be more suitable for network intrusion detection in ICSs. Xingjie *et al.* [Xingjie, Guogenp, ShiBIN & ChenHAO (2020)] constructed some attack paths from a hacker's perspective with an attack tree model and used the long short-term memory (LSTM) algorithm to identify and classify the attack behavior and then further classify the attack event by extracting atomic actions. The results showed that the model had a good effect on attack recognition and could effectively analyze the hacker's attack path and predict the next attack targets in ICSs. To secure IIoT data during the transmissions in wireless channels, Mukherjee *et al.* [Mukherjee, Goswami, Yang, Sah Tyagi, Samal & Mohapatra (2020)] used deep learning to enhance the physical layer security under different channel state conditions (environmental scenarios and application-dependent physical parameters). They selected a deep feedforward neural network as a learning model, and their simulation results showed that the proposed method had efficient security and scalability.

To overcome the limitations of single deep learning methods, Al-Abassi *et al.* [Al-Abassi, Karimipour, Dehghantanha & Parizi (2020)] proposed an ensemble deep learning-based cyber-attack detection technique to improve detection accuracy and reduce false-positive rate when using imbalanced ICS datasets. Their experimental results, obtained from two different datasets, showed that the proposed method outperforms conventional classifiers, including Random Forest (RF), single DNN, and AdaBoost. Li *et al.* [Li, Wu, Song, Lu, Li & Zhao (2021)] proposed a federated deep learning scheme that employed a CNN and a gated recurrent unit (GRU) to detect cyber threats against industrial cyber-physical systems. Javed *et al.* [Javed, Rehman, Khan, Alazab & G (2021)] proposed a combination of CNN and attention-based GRU model in vehicle security to detect single intrusion attacks as well as mixed intrusion attacks on controller area network (CAN) bus. Their experimental results showed that a combination of CNN and GRU increases the detection performance (accuracy, recall, precision, and F1 score) of attacks such as DoS, Fuzzy, and Impersonation assaults in CAN buses.

Although the works in the literature are instructive, most of them are based on shallow learning models. Thus, they are not suitable for critical ICSs as they may fail to fit complex and non-linear distributions from increasing numbers of training samples. Moreover, most of the studies that employed DNNs relied on a single detection model to keep classification quick and meet the requirements of latency-sensitive ICSs. They do not focus specifically on high-risk ICSs that cannot compromise on security for improved latency. Furthermore, most of these works proposed only anomaly detection models without mentioning attack classification, which is however crucial for preventing the attacks during the prevention phase. Another significant limitation is that most papers barely suggest response measures when attacks are detected. Last but not least, the majority of these works did not shed light on the deployment process of the models in ICSs network architecture.

3.3 The Hybrid Multistage DNN Detection Model

This section presents the components and building process of the proposed hybrid multistage DNN model. The main components used to build the model include: (i) a learning dataset split

into training and validation datasets; (ii) a test dataset used only to evaluate the final multistage DNN; and (iii) a set of DNN algorithms, including multi-layer perceptron (MLP), recurrent neural network (RNN), LSTM, and CNN. The building process has three main phases: first, the learning phase (*i.e.*, training, validation, and hyper-parameter optimization); second, the evaluation phase; and finally, the prediction phase.

3.3.1 The Training Phase

The stages of the multistage DNN are built sequentially. In the first stage, different DNN models are trained using the set of DNN algorithms and the learning dataset, called stage_1 dataset. Figure 3.2 presents how each DNN model is trained.

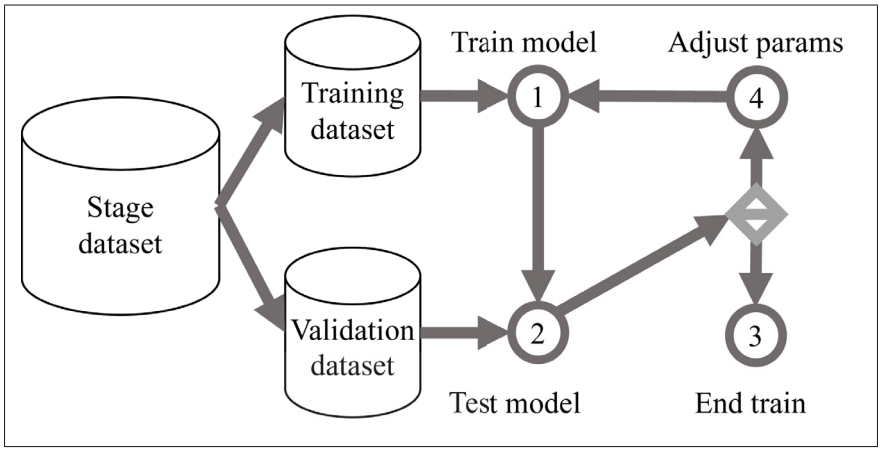

Figure 3.2 Activity diagram for building every stage model

When the first stage's learning process ends, the trained models are employed to classify all the samples in stage_1 dataset, and the model that has the best accuracy is stored as DNN_1. The samples that were misclassified or classified with low decision confidence in DNN_1's classification results are gathered in a second stage dataset, called stage_2 dataset. In the second stage, once more, different DNN models are trained using the set of DNN algorithms and the stage_2 dataset. The training process is the same as illustrated in Figure 3.2. At the end of the second stage's learning process, the trained models are employed to classify all the samples in stage_2 dataset, and the best model is stored as DNN_2. The data that were misclassified or

classified with low decision confidence in DNN_2's classification results are gathered in a dataset for the next stage. The subsequent stages are sequentially built until the number of misclassified samples falls below a predefined threshold. The algorithm used to train this multistage DNN is presented in Algorithm 3.1.

Algorithm 3.1 Proposed multistage DNN training

Result: Trained multistage DNN
{DNN_1, DNN_2, ..., DNN_n} ;
Inputs: Dataset *Initial_dataset* ; Predefined DNN algorithms *DNN_Algorithms* ; Predefined confidence threshold *Min_confidence* ; Predefined threshold to stop new stage creation *Min_dataset* ;
Variable: Misclassified samples of the current stage *Misclassified_samples* ;
Set *X* as current stage number and initialize *X* to 1 ;
Split *Initial_dataset* into *Stage_X_dataset* and *Test_dataset* ;
while *(*size of *Stage_X_dataset* larger than *Min_dataset)* **do**
    Split *Stage_X_dataset* into *Training_dataset* and *Validation_dataset* ;
    foreach *(DNN* in predefined *DNN_Algorithms)* **do**
        Recursively train *DNN*, test *DNN*, adjust *DNN* hyper-parameters **Until** (*DNN* performance improvement less than predefined threshold θ
    end
    Among all trained DNNs, set the *DNN* with the best accuracy as *DNN_X* ;
    Set *Misclassified_samples* as an empty table ;
    foreach *(*sample in *Stage_X_dataset)* **do**
        Predict the sample class using *DNN_X* ;
        if the sample is misclassified by *DNN_X* or decision confidence less than Min_confidence **then**
            Add the sample to *Misclassified_samples* ;
        end
    end
    Increment the stage number *X* ;
    Set *Misclassified_samples* as *stage_X_dataset* ;
end

3.3.2 The Evaluation and Prediction Phases

In the evaluation phase, for each sample in the test dataset, all stages DNNs' decisions are processed, assigned a weight, and used in combination functions to build the decision of the hybrid multistage DNN (Figure 3.3 illustrates this process). Various combination functions are used to classify the test dataset, including maximum, minimum, median, simple sum, weighted sum, and product (Table 3.1 defines these functions). The combination function that produces the best classification accuracy on the test dataset is assigned to the multistage DNN. Finally, this model is deployed in ICSs to classify incoming network traffic. The new traffic goes through the same process as illustrated in Figure 3.3.

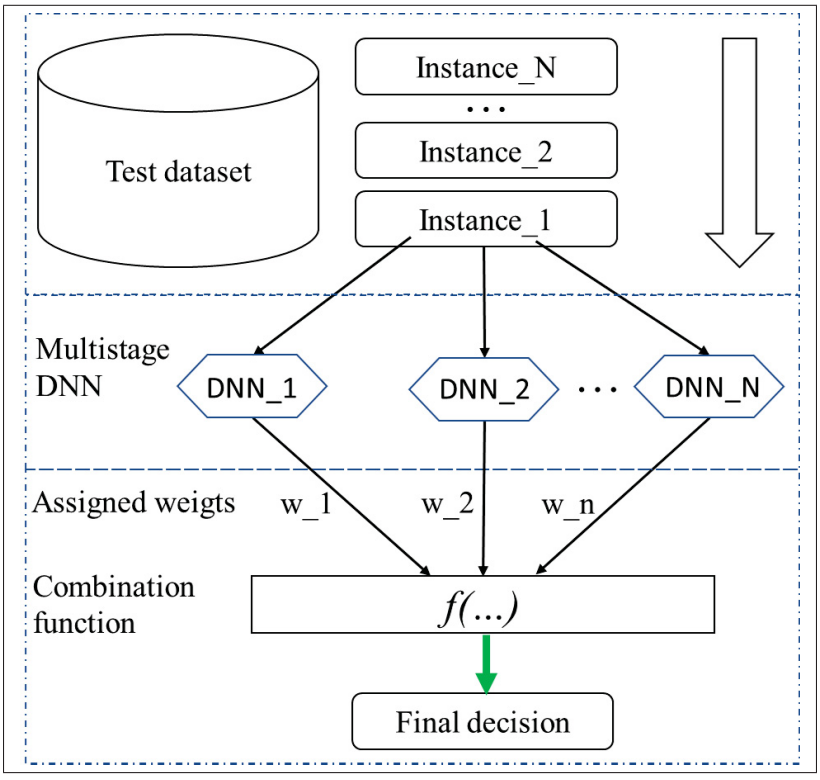

Figure 3.3 Activity diagram for the multistage DNN evaluation phase

The benefit of this approach is that the samples that are misclassified after each stage are re-used to improve the learning ability of the subsequent stage's DNN rather than being wasted.

Table 3.1 Stage DNNs combination rules

Combination	Definition of function $f(.)$
Maximum	$y_i = \max_j d_{ij}$
Minimum	$y_i = \min_j d_{ij}$
Median	$y_i = \text{median}_j d_{ij}$
Sum	$y_i = \frac{1}{N} \sum_j d_{ij}$
Weighted sum	$y_i = \sum_j w_j d_{ij}, w_j \geq 0, \sum_j w_j = 1$
Product	$y_i = \prod_j d_{ij}$

In these definitions, d_{ij} represents the output produced by DNN-j for the instance i. The number of DNNs used in the multistage DNN is represented by N. For the Weighted sum, w_i is the value to weight d_{ij}. The output of the multistage DNN for the instance i is represented by y_i.

Therefore, each new stage model focuses on a new subset of the data and can learn a new optimal decision frontier that corrects misclassifications from previous stage models. Even with imbalanced datasets, majority classes, which are easier to learn, will have their number of samples reduced with each additional stage, allowing the next stages DNNs to learn the distribution of minority classes. The improvement offered by the proposed approach is illustrated in Figure 3.4.

3.4 Proposed Collaborative IPS and Deployment Architecture

The function of intrusion prevention systems (IPSs), also known as intrusion response systems (IRSs), is to take defensive measures automatically when the IDS detects an attack. Whereas IDSs just generate reports or alarms, the automated response of IPSs reduces the time gap between detection and defensive action.

To further reduce this time gap in high-risk ICSs, this study proposes two collaborative hybrid multistage DNNs, the first of which, called IDS-ADM, is deployed on a local server and focuses on anomaly detection (binary classification) so that an emergency response can be launched as soon as an anomaly is detected. Then, the second multistage DNN, called IDS-ACM, classifies detected anomalies by attack type to identify suitable complementary response measures. Since binary classification is faster than multiclass classification, the proposed architecture makes

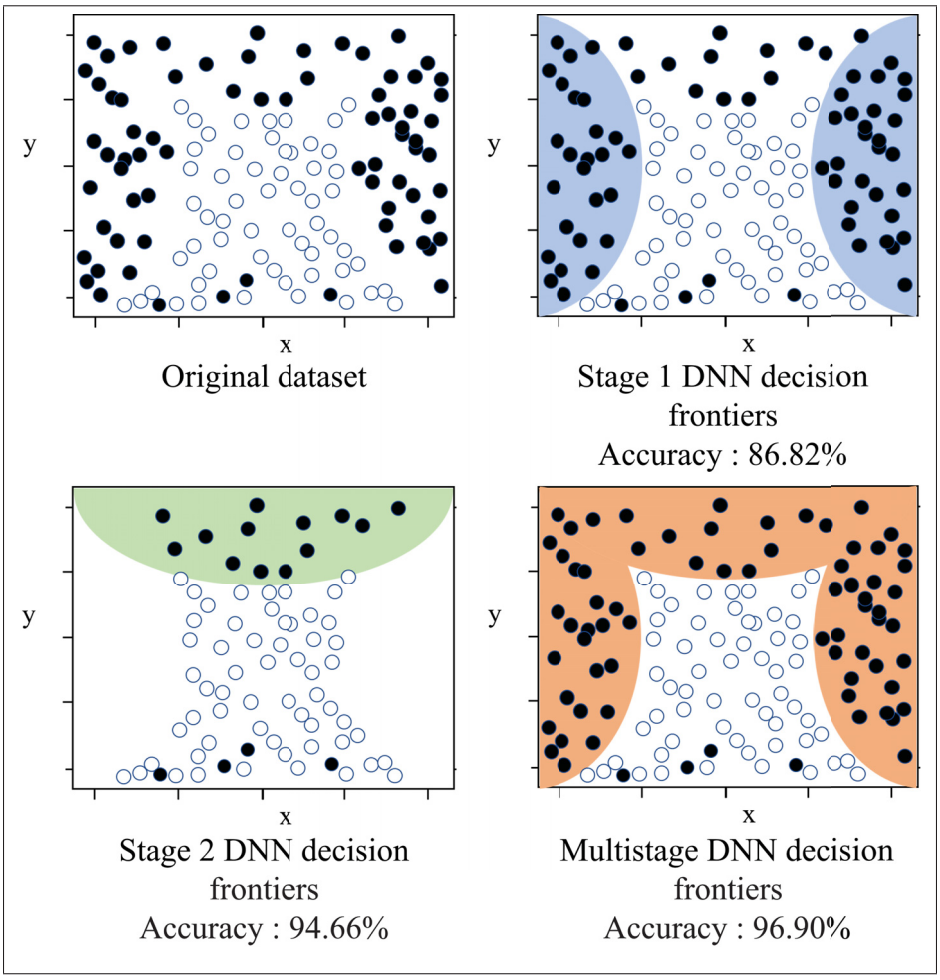

Figure 3.4 Stages decision frontiers and multistage DNN benefit

it possible for high-risk ICSs to respond sooner and limit an attack's impact on the targeted system. The attack classification module (IDS-ACM) can be deployed on cloud servers if more computation resources are needed. Only anomalous traffic, which represents a small portion of overall network data, needs to go to this second multistage DNN. This approach increases the IPS's responsiveness without reducing the IDS's classification accuracy. The proposed IDPS deployment architecture is illustrated in Figure 3.5. This architecture adopts an SDN approach [Li & Chen (2015)] to ensure efficient IDPS implementation in IIoT networks. Since our proposed system is not limited to a single SDN controller, we presented our architecture using a generic SDN architecture. For each ICS, we recommend that the SDN controller is chosen according to the system's network design and requirements in terms of programmability, scalability,

extensibility, resilience, and fault tolerance, among others. In the proposed deployment model, for every field, the IIoT devices are connected to the programmable logic controller (PLC) through an SDN switch. Data from the switch is collected by a local server on which the multistage DNN-based anomaly detection module (IDS-ADM) is deployed. Fields may be connected to remote fields having the same setup.

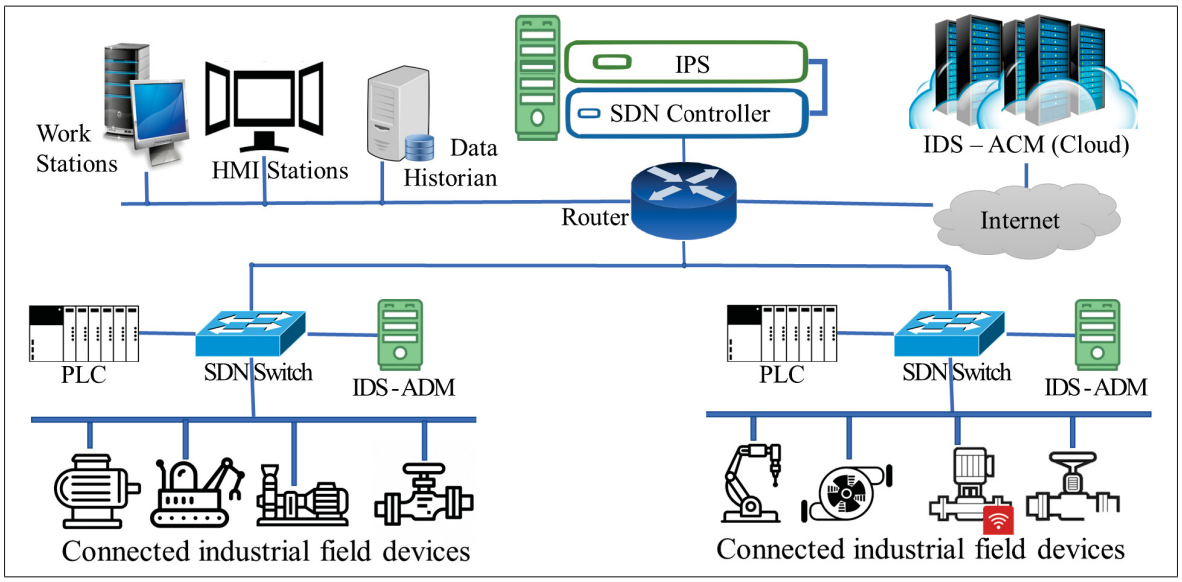

Figure 3.5 The proposed IIoT network architecture

When the IDS-ADM detects an anomaly, an alert and the traffic information related to the anomaly are sent to the IPS. Then, the IPS generates emergency response rules and transmits them to the SDN controller, which applies them to the SDN switches' flow tables. This enables the network to be configured automatically to prevent that eventual intrusion from compromising other parts of the ICS subnets. In the IPS, the first emergency response measure is to drop the malicious packets and block the attacker's IP and MAC addresses. This can stop many unauthorized user access attacks to preserve the system's integrity and confidentiality. It can also reduce scanning, probing, and DoS attacks to maintain system services. The intruder's connection can also be redirected to an attack analysis system (*e.g.*, a honeypot) to study the attacker's behavior [Tian *et al.* (2020)]. The IDS-ADM also sends the attack's features to the IDS-ACM, which classifies and sends its class and related traffic information to the IPS. Then,

the IPS generates complementary response rules and transmits them to the SDN controller, which applies these routing rules to the SDN switches' flow tables. In addition, advanced modules can be implemented to counter-attack the attacker's system and slow them down.

3.5 Experiment

This section presents our experimental evaluation and discusses the results obtained. The goal is to evaluate the performances of the proposed multistage DNN models in the anomaly detection and attack classification modules. For the sake of conciseness, the details about the response module's implementation will be the subject of a separate paper. The available ICS datasets usually present several limitations, such as the absence of realistic attacks. Thus, to validate our proposal, we conducted experiments on two different datasets having a different mix of attacks, data distribution, and classification complexity.

The first dataset, named WUSTL-IIOT-2018 [Alem & BB (2021); Teixeira *et al.* (2018)], is used for SCADA cyber-security research. The dataset was built using a SCADA system test bed. The data was generated by using scan tools to inspect the topology of the victim network and identify the devices in the network and their vulnerabilities. An assortment of port scanner, address scan, device identification and exploit attacks were carried out against the test bed. The traffic captured comprised 7,049,989 observations, of which 93.93% were normal traffic (without attacks) and 6.07% were abnormal traffic (with attacks). The raw data contained 25 detection features, from which five features were selected based on their variation during attacks, as well as during normal traffic (Table 3.2 presents the selected features). Each vector in the dataset is labelled normal or attack, depending on the case.

The second dataset, named NSL-KDD, was built from a regular (non-IIoT) network; however, it is the dataset most commonly used to compare machine learning-based IDSs [Tavallaee *et al.* (2009)]. NSL-KDD includes a total of 39 specific attacks grouped into four different attack categories, namely, probing (Probe), DoS, remote to local (R2L), and user to root (U2R). The training dataset (KDDTrain+) and testing dataset (KDDTest+) contain 125,973 and 22,544

Table 3.2 Features selected in the WUSTL-IIOT-2018 dataset

Detection features	**Descriptions**
Source port (Sport)	Port number of the source
Total packets (TotPkts)	Total transaction packet count
Total bytes (TotBytes)	Total transaction bytes
Source packets (SrcPkts)	Source to Destination packet count
Destination packets (DstPkts)	Destination to source packet count
Source bytes (SrcBytes)	Source to destination transaction bytes

single connection vectors, respectively. Each connection vector is represented by 41 features and labelled either normal or attack, with exactly one specific attack type. Table 3.3 presents the detection features used in the NSL-KDD dataset.

In our experiment, we implemented and compared three different approaches: (i) single DNNs, (ii) simple multistage DNNs using the same DNN algorithms with different hyper-parameters at each stage, and (iii) the hybrid multistage DNN. The list of DNN algorithms used for the single DNNs, simple multistage DNNs, and hybrid multistage DNN include MLP, RNN, and CNN. The experiment was done in two stages, *i.e.*, anomaly detection and then attack classification. All the tasks were performed using the Python programming language and scikit-learn (a free machine learning library).

3.5.1 The Anomaly Detection Phase

We first implemented and evaluated the anomaly detection module. Table 3.4 presents the organization of the WUSTL-IIOT-2018 and NSL-KDD datasets for this binary classification. We split the WUSTL-IIOT-2018 into stage_1 learning (80%) and test (20%) sets. For each stage, we split the learning dataset into training (60%) and validation (40%) sets. For the NSL-KDD dataset, we used KDDTrain+ as the learning dataset and KDDTest+ as the test dataset. Figure 3.6 presents the models' accuracy on the WUSTL-IIOT-2018 dataset, and Figure 3.7 shows the models' accuracy on the NSL-KDD dataset. For the hybrid multistage model, the accuracy reported was obtained from the best combination functions.

Table 3.3 Detection features used in the NSL-KDD dataset

Detection features	**Descriptions**
Duration	Length (number of seconds) of the connection
Protocol_type	Type of the protocol, *e.g.*, tcp, udp, etc.
Service	Network service on the destination, *e.g.*, http, telnet, etc.
Src_bytes	Number of data bytes from source to destination
Dst_bytes	Number of data bytes from destination to source
Flag	Normal or error status of the connection
Land	1 - connection is from/to the same host/port; 0 - otherwise
Wrong_fragment	Number of "wrong" fragments
Urgent	Number of urgent packets
Count	Number of connections to the same host as the current connection in the past 2 seconds
The following features refer to these same-host connections	
Serror_rate	% of connections that have "SYN" errors
Rerror_rate	% of connections that have "REJ" errors
Same_srv_rate	% of connections to the same service
Diff_srv_rate	% of connections to different services
Srv_count	Number of connections to the same service as the current connection in the past 2 seconds
The following features refer to these same-service connections	
Srv_serror_rate	% of connections that have "SYN" errors
Srv_rerror_rate	% of connections that have "REJ" errors
Srv_diff_host_rate	% of connections to different hosts
Hot	Number of "hot indicators"
Num_failed_logins	Number of failed login attempts
Logged_in	1 - successfully logged in; 0 - otherwise
Num_compromised	Number of "compromised" conditions
Root_shell	1 - root shell is obtained; 0 - otherwise
Su_attempted	1 - "su root" command attempted; 0 - otherwise
Num_root	Number of "root" accesses
Num_file_creations	Number file creation operations
Num_shells	Number of shell prompts
Num_access_file	Number of operations on access control file
Num_outbound_cmds	Number of outbound commands in a ftp session
Is_hot_login	1 - the login belongs to the "hot" list; 0 - otherwise
Is_guest_login	1 - the login is a "guest"login; 0 - otherwise

Table 3.4 Organization of WUSTL-IIOT-2018 and NSL-KDD datasets for the binary classification

Class	Number of records in the datasets		
	WUSTL-IIOT	***KDDTrain+***	***KDDTest+***
Normal	6,610,778	67,343	9,711
Attack	427,205	58,630	12,833
Total	7,037,983	125,973	22,544

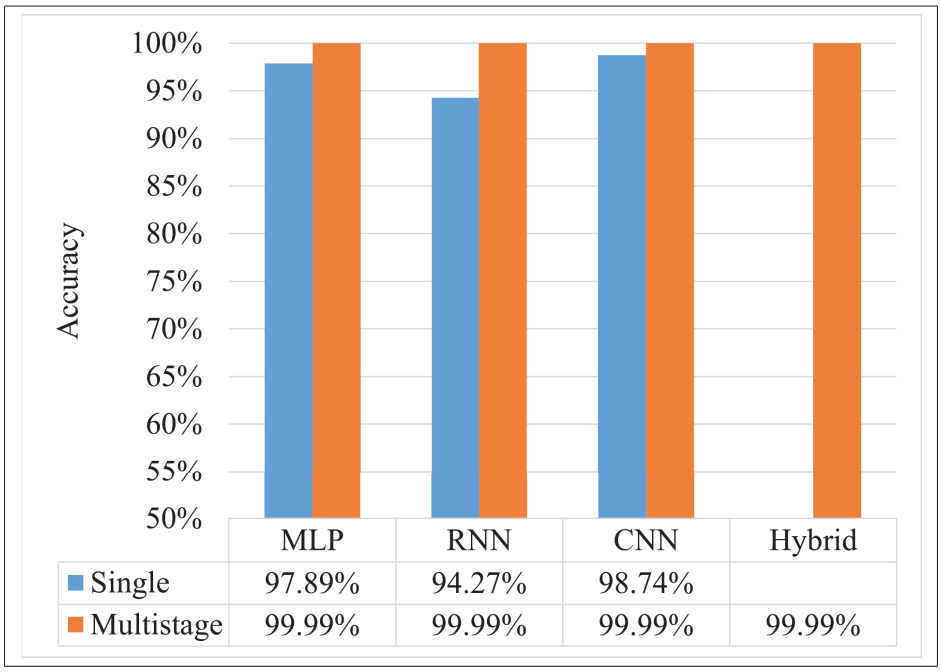

Figure 3.6 Accuracy of anomaly detection models using the WUSTL-IIOT-2018 dataset

With the WUSTL-IIOT-2018 dataset, the hybrid multistage DNN achieved the best performance, providing 99.99% detection accuracy, 0.18% undetected rate (false negative rate), and less than 0.01% false alarm rate (false positive rate). All the simple multistage DNNs achieved 99.99% detection accuracy, and the single DNNs achieved over 97% detection accuracy.

The hybrid multistage DNN also achieved the best performance with the NSL-KDD dataset, providing 97.02% detection accuracy, 2.07% undetected rate, and 4.16% false alarm rate. However, in this dataset, the simple multistage DNNs showed significantly lower performances,

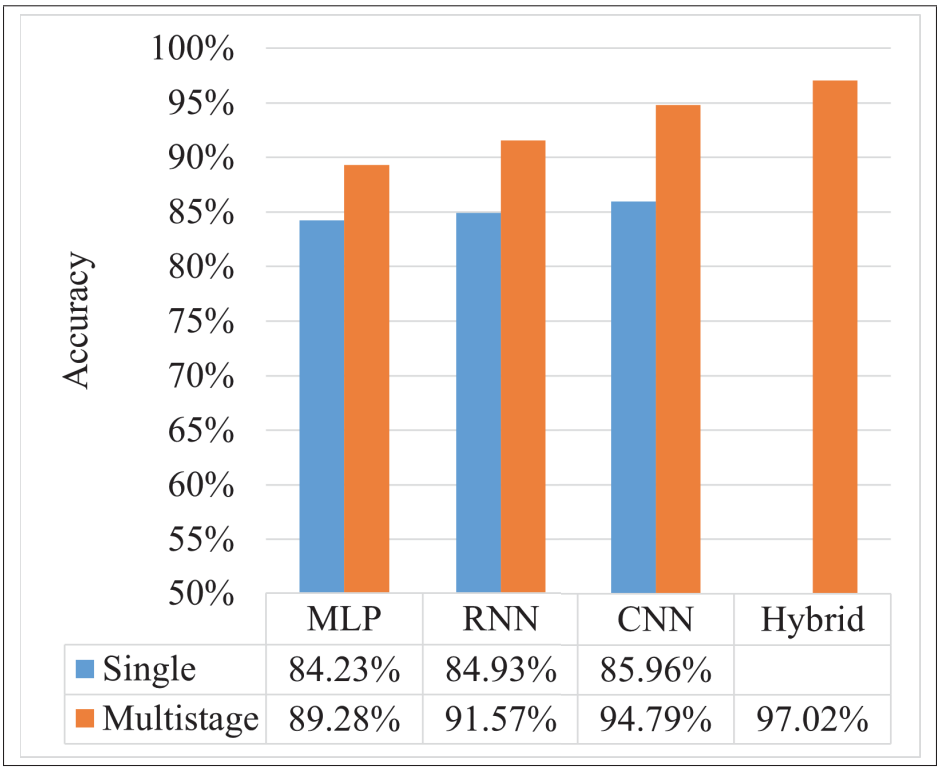

Figure 3.7 Accuracy of anomaly detection models using the NSL-KDD dataset

yielding less than 95% detection accuracy. Moreover, the performances of the single DNNs were also significantly low, attaining an accuracy of less than 86%. This can be explained by the fact that the models that employed single DNNs had difficulties understanding the complicated data distribution of intrusion patterns in the NSL-KDD dataset, which involved 39 specific attacks. The models that employed simple multistage DNNs were able to learn more intrusion patterns but less effectively than the hybrid multistage DNN. In the WUSTL-IIOT-2018 dataset, the single DNNs and simple multistage DNNs achieved high detection accuracy since the intrusion patterns in this dataset involved less complicated data distribution, which is not a realistic scenario in real ICSs.

This demonstrates that using only single DNNs is not enough to build accurate intrusion detection models for highly endangered ICSs. It also confirms that multistage DNNs is a more efficient

approach and hybrid multistage DNNs are the most effective when it comes to building more robust intrusion detection models.

3.5.2 The Attack Classification Phase

The second part of the experiment pertained to attack classification. Since the WUSTL-IIOT-2018 dataset does not include attack type, we performed attack classification on only the NSL-KDD dataset. Table 3.5 presents the organization of the NSL-KDD dataset for attack classification. We split the learning dataset, *i.e.*, KDDTrain+, into training (60%) and validation (40%) sets, and used KDDTest+ as the test dataset. The same DNNs, *i.e.*, single DNNs, simple multistage DNNs, and the hybrid multistage DNN, were used to build the attack classification models. Figure 3.8 illustrates the accuracy of the models on the NSL-KDD dataset. For the hybrid multistage DNN model, the reported accuracy was obtained from the best combination functions.

Table 3.5 Organization of the NSL-KDD dataset for attacks classification

	Number of records used in the datasets	
Class	***KDDTrain+***	***KDDTest+***
DOS	45,927	7,458
Probe	11,656	2,421
R2L	995	2,754
U2R	52	200
Total	58,630	12,833

The attack classification results also confirm that the single DNNs have lower accuracy than the multistage DNNs. Moreover, the proposed hybrid multistage DNN achieved higher detection accuracy than the simple multistage DNNs. As compared to previous works that employed the NSL-KDD dataset for anomaly detection and attack classification, the proposed hybrid multistage DNN-based IDPS is able to make predictions with more accuracy. Table 3.6 compares the accuracy of the solutions used in previous works and our attack classification

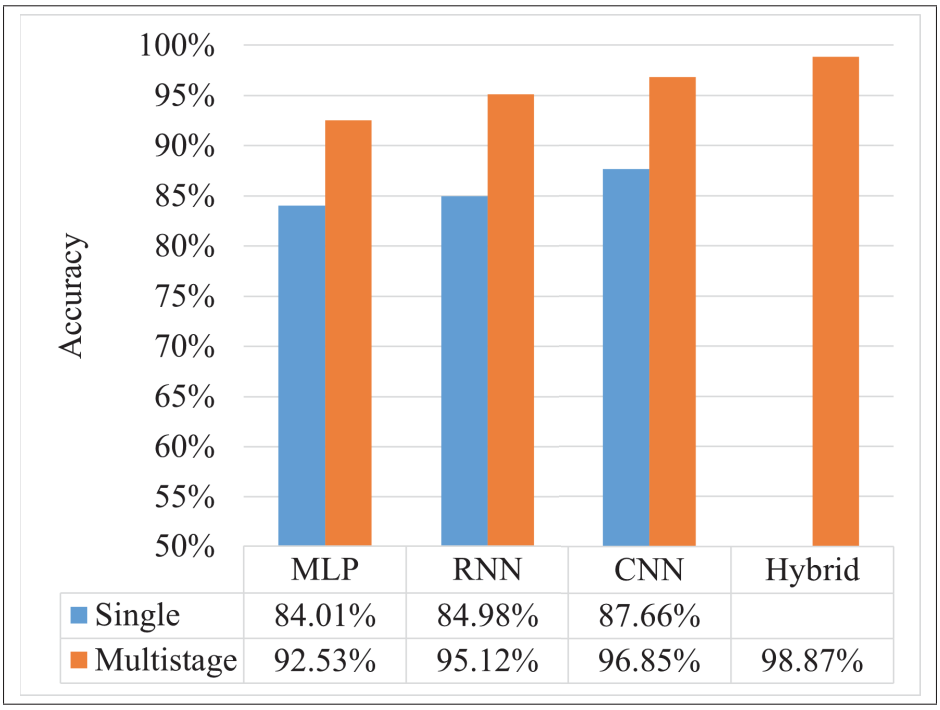

Figure 3.8 Accuracy of attack classification models using the NSL-KDD dataset

model (four-category classification). The experimental evaluation, therefore, demonstrates that our approach is more suitable for critical ICSs whose primary requirement is security.

Each module in the proposed system can be enhanced even further. First, employing more DNN algorithms in the hybrid multistage DNN, such as generative adversarial networks (GANs) and radial basis function networks (RBFNs), and exploring more combination functions can further improve the accuracy in the IDS-ADM and IDS-ACM modules, and strengthen the security in high-risk ICSs. For example, in the IDS-ADM, this may enable the detection of most of the attacks that go undetected; while in the IDS-ACM, this can enable correct classifications of most of the attacks that get confused with other attack categories. Second, the IDS-ADM can be combined with signature-based detection systems for better performance. Finally, to minimize misclassifications, it is possible to implement a rejection option in the decision model. This option enables the model to abstain from making a decision when the decision confidence

is still below the predefined threshold. In our future works, these additional features will be implemented.

Table 3.6 Comparison of solutions using NSL-KDD dataset for multiclass classification

Methods used	**Accuracy**
Proposed hybrid multistage DNN-based IDPS	98.87%
Methods in [Tavallaee *et al.* (2009)] : NBTree Random Tree Decision Trees J48 Random Forest Multi-Layer Perceptron	- 82.02% 81.59% 81.05% 80.67% 77.41%
Methods in [Bajaj & Arora (2013)]: SimpleCart Multi-layer Perception	- 82.32% 73.54%
Recurrent neural networks [Yin, Zhu, Fei & He (2017)].	81.29%
ANN with tansig transfer function, Levenberg-Marquardt (LM) and BFGS quasi-Newton Backpropagation [Ingre & Yadav (2015)].	81.20%
A Two-Layer Dimension Reduction and Two-Tier Classification Model [Pajouh, Javidan, Khayami, Dehghantanha & Choo (2019)].	84.86%
Two-stage intrusion detection technique combining Naive Bayes and k-means [Vishwakarma & Kesswani (2022)].	86.46%

3.6 Conclusion

Cyber-attacks on high-risk industrial control systems could result in dramatic consequences, including death, long-term environmental damage, and huge economic losses. These infrastructures require specific security systems that do not compromise on security for the sake of productivity. This work proposes a hybrid multistage deep learning model that offers robust intrusion detection systems for these critical networks. In the proposed approach, learning models are built sequentially to work on the limitations of the previous learners. The multistage DNN uses each stage's decision in a combination function to produce a final decision with better accuracy. This approach offers better classification accuracy at the cost of latency and is much more suitable for high-risk ICSs in which security is more important than any other aspect of productivity, such as latency. This work also proposes a collaborative intrusion prevention

system that employs one multistage DNN to detect anomalies and take emergency defensive action as soon as possible and a second multistage DNN to classify the detected anomalies and apply complementary defensive measures. An SDN-based architecture is presented to deploy the proposed collaborative IDPS in ICS networks. The experimental evaluation, which was performed on two datasets with different challenges, demonstrated the effectiveness of the proposed approach.

CHAPTER 4

A COLLABORATIVE DNN-BASED LOW-LATENCY IDPS FOR MISSION-CRITICAL SMART FACTORY NETWORKS

Poulmanogo Illy [a] and Georges Kaddoum [a,b],

[a] Département de Génie Électrique, École de Technologie Supérieure,
1100 Rue Notre-Dame Ouest, Montréal, Québec, H3C 1K3, Canada
[b] Cyber Security Systems and Applied AI Research Center,
Lebanese American University, Beirut, Lebanon

Paper published in *IEEE Access*, vol. 11, pp. 96317-96329, September 2023

4.1 Introduction

Industrial facilities are usually highly delicate and risky environments, requiring maximum safety and security to work with potentially dangerous chemicals and tools, and privacy to manufacture highly competitive products. Therefore, many security, safety, and privacy standards have been implemented over the years to protect these environments [Robla-Gómez, Becerra, Llata, González-Sarabia, Torre-Ferrero & Pérez-Oria (2017); Stoessel (2021); Zhao & Barati (2023)]. However, industrial accidents and disasters still occur frequently worldwide and cause significant damages, including deaths, injuries, economic losses, and long-term environmental impacts. According to the survey published in June 2021 by the research department of the database company Statista, there were 984 explosive incidents across the United States (USA) in 2020, which is a significant increase compared to previous years [Statista Research Department (2021)]. The sources of these explosions include manufacturing plants, electric utilities, petroleum industries (upstream, midstream, downstream, pipelines), and other chemical factories. The evolution of the industrial domain towards the new era of modernization driven by the Industrial Internet of Things (IIoT) enables real-time safety applications to prevent the traditional risk of incidents. However, it generates new security risks, notably from cyber-attacks that exploit the vulnerabilities of the connected objects.

Unlike traditional standalone Industrial Control Systems (ICSs), which used to be isolated from Information and Communication Technology (ICT) networks, new ICSs integrate these networks to enable high-level process supervisory management. Indeed, thanks to connected sensors, actuators, and IIoT devices, manufacturing equipment (pumps, valves, compressors, tanks, etc.) and manufacturing conditions (air quality, humidity, temperature, etc.) can now be easily monitored and controlled remotely using computers, tablets, or smartphones. For more efficient control systems operation (better resource allocation, easier and faster data collection), many industries are adopting the Supervisory Control And Data Acquisition (SCADA) architecture. This architecture is generally made up of three main levels. The first level consists of sensors and actuators, where the sensors collect data about the system, and the actuators control the system's state. The second level is composed of Programmable Logic Controllers (PLCs), which are connected to the first level in order to control the actuators and collect information from the sensors. The third level contains the supervisory controls, which communicate with the PLCs to send control commands from workstations, store the system data in servers (data historians), and provide a visual representation of the system on a Human-Machine Interface (HMI). Figure 4.1 illustrates a generic SCADA system.

In terms of industrial safety and security, the adoption of ICT, especially IIoT, brings a lot of opportunities and challenges. For instance, a factory safety system integrated with the factory automation system can provide additional safety services, such as personalized alerts (*e.g.*, material physical default, place humidity, abnormal temperature, etc.), access control, occupancy detection, person identification, central locking of all perimeter doors and windows, remote surveillance of security cameras and sensors over the Internet, and others. However, the factory's IIoT can bring significant security problems to the factory because of the security vulnerabilities in IoT devices and software that hackers could leverage to carry out malicious activities. For example, attackers have found vulnerabilities in Siemens Step7, a software used in PLCs, and exploited it to launch an attack named Stuxnet. Stuxnet collected surveillance data, put ICSs into a critical state, and even falsely responded to prevent alarms [Al-Rabiaah (2018)]. A successful

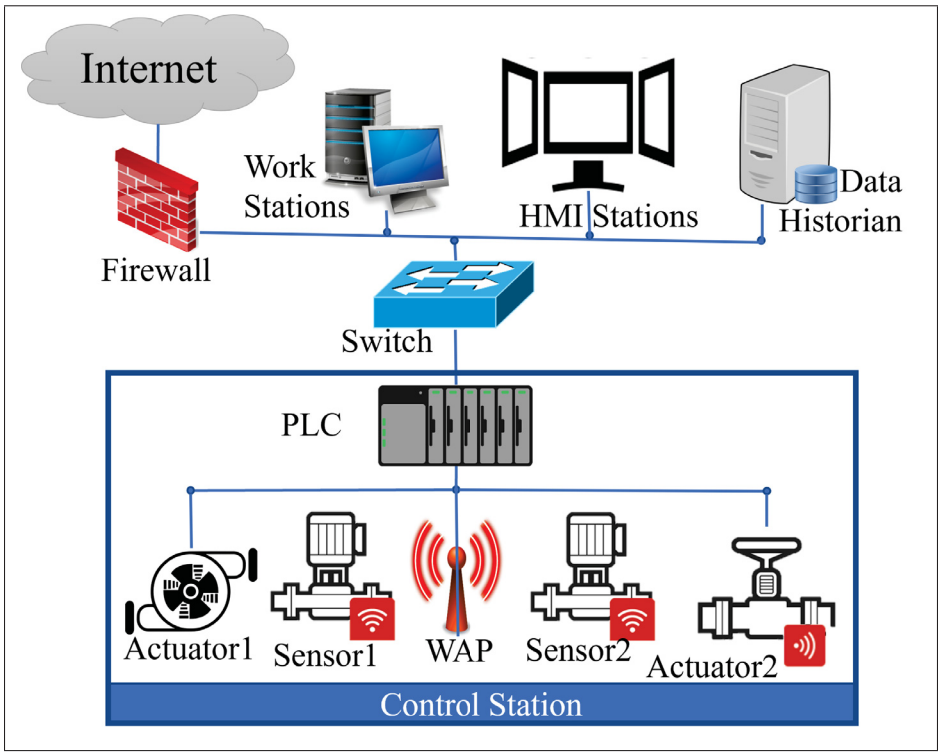

Figure 4.1 Generic SCADA architecture

cyber-attack on a smart factory may have critical consequences, including private data leakage and cyber-terrorism.

4.1.1 Related Work

To overcome smart ICSs' cyber-security challenges, the research community is active in designing solutions for different security layers, especially Intrusion Detection Systems (IDSs), to detect malicious attempts (succeeded or failed) to penetrate IIoT networks. For example, Justin M. *et al.* [Beaver *et al.* (2013)] applied machine learning algorithms to detect malicious Remote Terminal Unit (RTU) communications. The authors used a dataset composed of labelled RTU telemetry data from a gas pipeline system in the Mississippi State University's Critical Infrastructure Protection Center, USA. In this dataset, variants of command injection and data injection attacks were considered. The authors implemented six different machine learning algorithms, including Naive Bayes, Random Forests, One Rule (OneR), J48 (a type of Decision Tree

technique), Non-Nested generalized exemplars (NNge), and Support Vector Machines (SVM). Their experimental results demonstrated the ability of the learning algorithms to detect these attacks. Chih-Ta *et al.* [Lin *et al.* (2017)] established an industrial control system test-bed where they examined operational cases and developed a Modbus/Transmission Control Protocol (TCP) network attack program. They executed dozens of penetration attacks successfully, including address and function code scans, response and command injections, and Denial of Service (DoS). All network activity, including records of normal behavioral patterns, was collected. Exploiting content from the data link layer through the application layer, especially, Medium Access Control (MAC) and Internet Protocol (IP) addresses, TCP ports, and Modbus functions and data, the authors planned as future research to build on machine learning-based detection models, *e.g.*, a one-class SVM to find outliers, and thus effectively detect the occurrence of abnormal events. Marcio Andrey *et al.* [Teixeira *et al.* (2018)] developed a SCADA system test-bed where they conducted sophisticated cyber-attacks, including port and address scans, device identification attacks, and exploits. The authors captured the network traffic during the attacks, extracted features, and built a dataset for training and testing different machine learning algorithms. Five shallow machine learning algorithms were trained to detect the attacks, namely, Random Forest, Decision Tree, Logistic Regression, Naïve Bayes, and K-Nearest Neighbors (KNN). Their evaluation results showed the efficiency of the machine learning models in detecting the attacks in real-time.

To achieve low computational complexity and latency in anomaly-based intrusion detection models for SCADA networks, Ullah and Mahmoud [Ullah & Mahmoud (2017)] implemented a feature selection filter based on the information gain. Using an industrial control system dataset developed at the Distributed Analytics and Security Institute at Mississippi State University, the proposed model selected a subset of five features out of the 20 initial features.

The dataset was used to train a J48 classifier. Then, then a Bayesian network classifier was used to develop the proposed model, which correctly classified all instances of the binary-labelled and categorized-labelled datasets. An Le *et al.* [Le, Dinh, Le & Tran (2015)] proposed an Intrusion Detection and Prevention System (IDPS) that leveraged the Software-Defined Networking (SDN)

approach to reduce the cost and decrease detection and mitigation latency. In the proposed system, the authors employed a C4.5 Decision Tree algorithm to build the detection model. The data used to train the model included 25 basic features (packet headers) and derived features (features that are computed). The system was evaluated using the probing and DoS attacks present in the 1999 DARPA dataset and a small test-bed where they generated attacks consisting of 3 types of DoS and 8 types of Probe. In the deployment, OpenFlow switches were implemented to replace traditional switches, IDPS sensors, and the firewall, to reduce the total cost of the IDPS.

Most of these solutions are based on shallow machine learning methods; therefore, they may suffer from significant limitations associated with shallow learning. To overcome these limitations, recent research works investigate deep learning-based IDSs. For example, Al-Abassi *et al.* [Al-Abassi *et al.* (2020)] demonstrated an attack detection model that leveraged Deep Neural Networks (DNNs) and Decision Tree classifiers to detect cyber-attacks in an ICS environment. Li *et al.* [Li *et al.* (2021)] designed a deep learning-based intrusion detection model by making use of a Convolutional Neural Network (CNN) and a Gated Recurrent Unit (GRU). Then, the authors developed a federated learning framework, allowing multiple ICSs to collectively build a comprehensive intrusion detection model in a privacy-preserving way. Jie Ling *et al.* [Ling, Zhu, Luo & Wang (2021)] studied the limitations of intrusion detection methods based on deep learning, such as Long Short-Term Memory (LSTM) and GRU, to highlight problems these methods are still facing, such as vanishing gradients and low training efficiency. Then, the authors proposed an intrusion detection method based on a Bidirectional Simple Recurrent Unit (BiSRU). With skip connections employed, the optimized bidirectional structure in the SRU neural network alleviated the vanishing gradient problem and improved the training effectiveness. The author in [Mansour (2022)] proposed an Attention-based Bi-Directional Gated Recurrent Neural Network (ABi-GRNN) model with a Poor and Rich Optimization algorithm-based hyper-parameter optimizer to build an efficient IDS for cyber-physical systems. The proposed system applied blockchain technology to boost security in the cyber-physical environment. The

solution was evaluated using the NSL-KDD and CICIDS datasets, and the experimental results showed better precision compared to other DNNs, such as GRU and optimal GRU.

Although instructive, most of these deep learning-based IDSs for ICSs do not analyze the time complexity of DNNs and the latency of the proposed detection models. To enable a timely identification of various attacks and near real-time neutralization of threats, Shafi *et al.* [Shafi, Basit, Qaisar, Koay & Welch (2018)] proposed a fog-assisted SDN and blockchain-driven IDPS for IoT networks. The authors employed three DNN-based classifiers, namely, Recurrent Neural Network (RNN), Multi-Layer Perceptron (MLP), and Alternate Decision Tree (ADT) in parallel with a voting system to identify attacks at the edge network just beside IoT devices. This approach reduces the detection latency by bringing the detection system as near as possible to the edge devices. However, unlike our approach, all classification models are deployed in cloudlet or fog nodes, and no advantage is taken from cloud resources. Alkadi *et al.* [Alkadi, Moustafa, Turnbull & Choo (2021)] proposed a distributed IDS that employed a Bidirectional Long Short-Term Memory (BiLSTM) deep learning algorithm to deal with sequential network data. The proposed system implemented a blockchain and smart contract method to provide privacy to the distributed intrusion detection engines. This approach provides security and concurrently ensures data privacy in cloud environments but does not address latency concerns for mission-critical IoT applications.

In addition to the lack of time complexity analysis for most DNN-based IDSs for ICSs, most previous studies do not consider attacks classification and intrusion response methods when attacks are detected, or the deployment architecture of the deep learning-based IDSs within the ICS networks.

4.1.2 Contributions

The novelty of this study is in its effort to design a DNN-based IDPS that preserves mission-critical ICSs requirements. Mission-critical ICSs refer to IIoT applications that require high availability and low latency to ensure real-time operations. In such applications, security

measures must be designed to preserves the low latency requirement. Therefore, this study proposes an intrusion detection and response system that combines robustness and low latency. To achieve robustness, this work identifies various detection features and employs the most promising DNNs to build the detection models. To meet the ICSs' low latency requirement, the structure of the neural networks and their time complexity are studied. Then, a collaborative scheme that separates the classification task into two consecutive models organized based on a priority principle in the IDPS is implemented. The first model, based on a lightweight DNN, performs anomaly detection on local servers to allow timely attack detection and response. The second model performs attack classification of the anomalous traffic to help choose the suitable intrusion response measures to stop the attack. This multi-class classifier is deployed in cloud servers to benefit from the high computation resources available for running complex DNNs. The major contributions of the proposed work are four-fold:

- This paper proposes a time complexity analysis of the DNNs and highlights the variables that impact the training and the prediction latency.
- A collaborative deep learning-based IDS that employs two classification models is proposed along with its deployment architecture in the ICS.
- Various detection features, DNNs algorithms, and Intrusion Response System (IRS) measures are proposed to implement the IDPS.
- An experimental evaluation is conducted using different datasets, which demonstrates the efficiency of the proposed approach.

4.1.3 Organization

The rest of the research paper is organized as follows. Section II introduces the time complexity analysis of the DNNs and highlights the variables that impact the training and the prediction latency. Section III illustrates the collaborative deep learning-based IDPS and the deployment architecture. Section IV presents the detection features, learning algorithms, and IRS measures.

Section V presents the experimental evaluation and discusses the obtained results. Finally, Section VI concludes this work.

4.2 Time Complexity analysis of deep neural networks

The time complexity in the training and prediction of the DNNs are proportional to the number and size of elementary structures and functions involved in their networks, *i.e.*, the neurons, recursions, convolutions, pooling, etc. This section evaluates how some of these elementary structures increase training and detection latency. In order to understand the latency reduction scheme, this evaluation is essential. Figure 4.2 illustrates the DNN architectures considered in this work. However, for conciseness, the time complexity evaluation process is only detailed for the MLP architecture.

To evaluate the time complexity of the MLP, first, we evaluate the main operations involved in a single-layer neural network, *i.e.*, the perceptron, and formulate these operations' time complexities. Then, based on the training and prediction algorithms, we evaluate the whole MLP time complexity.

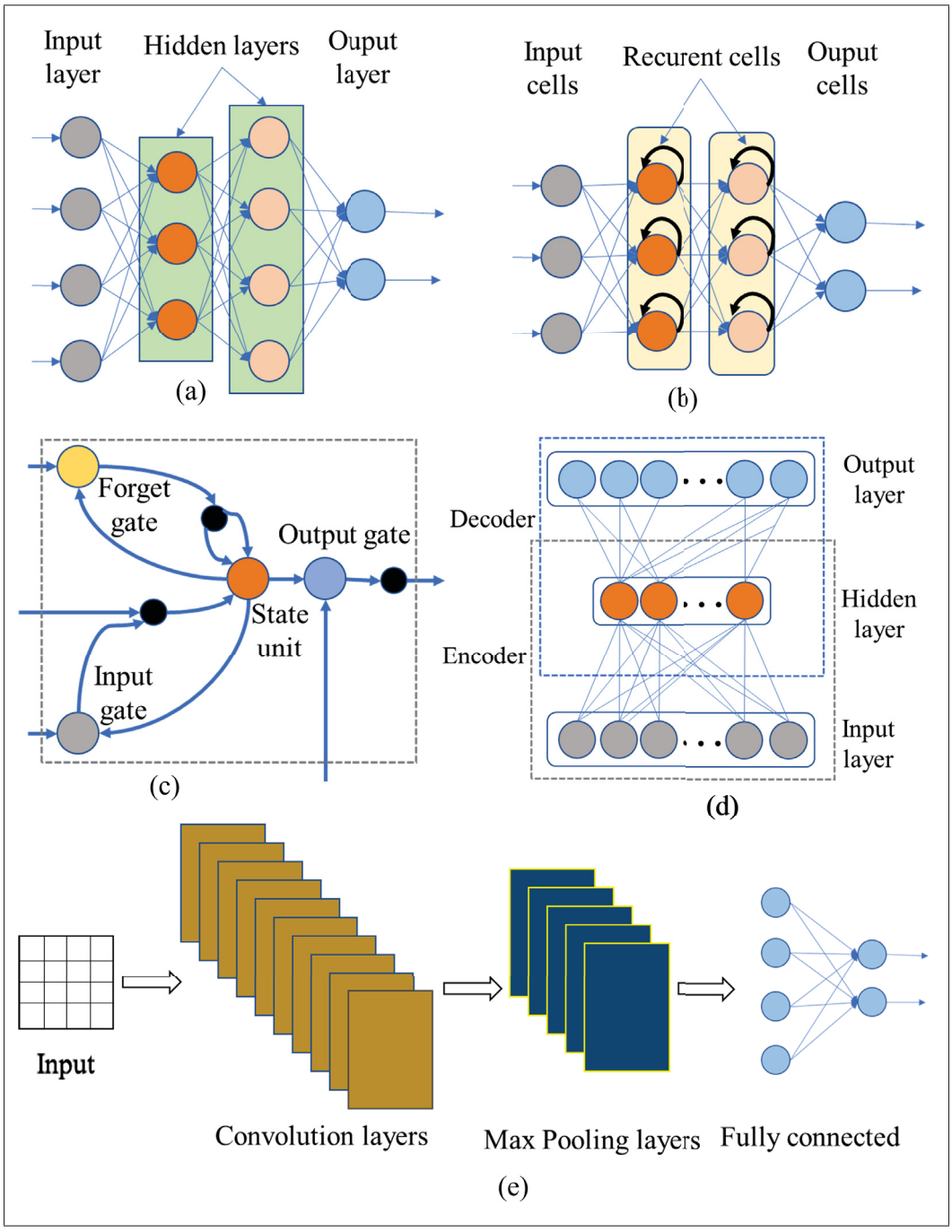

Figure 4.2 The architecture of MLP(a), RNN(b), LSTM(c), Auto Encoder(d), and CNN(e)

The main operations in the perceptron are the feed-forward, error computation, and weights updates. Considering the perceptron illustrated in Figure 4.3, these operations are defined in Table 4.1.

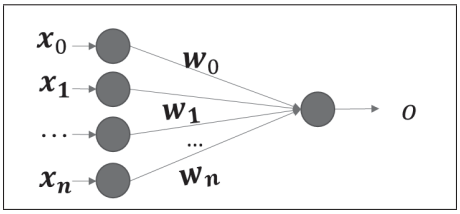

Figure 4.3 Single-layer neural network: Perceptron

Table 4.1 Perceptron training operations

Operation	Equation
1. Feed-forward	$o = \sum_{i=0}^{n} w_i x_i$
2. Error computation	$\Delta w_i = \eta(t - o)x_i$
3. Weights updates	$w_i = w_i + \Delta w_i$

In these definitions, x_0 represents the bias unit. The input values (features) are represented by $x_1, x_2, ..., x_n$. The weights are represented by w_i, and o represents the computed output unit. The learning rate is represented by η, and the targeted output is represented by t.

In time complexity analysis, operations that have constant times can be neglected because the goal is to highlight how the latency scales with respect to the variables. Therefore, to simplify the time complexity approximation, we make the following assumptions:

- In the feed-forward, the time complexity to compute each perceptron (hidden or output unit) is identical and represented by T_{FW};
- In the error computation, the time complexity to compute each output unit error is identical and identical by T_{OE};
- In the weights update, the time complexity to update each weight is identical and represented by T_{WU};

The neural network training consists of recursive feed-forward, error computation, and back-propagation to adjust the network weights. The condition of the recursion is to get the values of

weights that minimize the output error on a particular set of training data. Algorithm 4.1 defines the MLP training process.

Algorithm 4.1 MLP training

Result: Weights that minimize the prediction error.
Input 1: $training\_examples$, where each instance is represented by $\langle \vec{x}, t \rangle$, $\vec{x}$ represents the feature vector and t is the class of the instance.
Input 2: Weights $w_{i,j}$ where i and j represent the input and output units, respectively.
Input 3: Learning rate η.
Initialize all weights $w_{i,j}$ to small random numbers;
while *termination condition not satisfied* **do**
  foreach $\langle \vec{x}, t \rangle$ *in* $training\_examples$ **do**
    Execute **Feed-forward**
    foreach *hidden and output unit* j **do**
      $o_j = \sum_{i \in inputs} w_{i,j}\, o_i$
    end
    Compute **Outputs errors**
    foreach *last layer unit* k **do**
      $\delta_k = o_k(1 - o_k)(t_k - o_k)$
    end
    Execute **Back-propagation**
    foreach *hidden unit* h **do**
      $\delta_h = o_h(1 - o_h) \sum_{k \in outputs} w_{h,k}\, \delta_k$
    end
    Execute **Weights update**
    foreach *network weight* $w_{i,j}$ **do**
      $\Delta w_{i,j} = \eta \delta_j x_{i,j}$
      $w_{i,j} = w_{i,j} + \Delta w_{i,j}$
    end
  end
end

The time complexity of this training depends on the following variables:

- N_{in} = the number of input units (feature vector size),
- N_{h_i} = the number of units of the hidden layer h_i,
- N_{out} = the number of output units (number of classes),

- N_{nu} = the total number of units (perceptrons),
- N_{wg} = the total number of weights,
- N_{epoch} = the number of iterations in the training dataset,
- N_{te} = the number of training examples,

where: N_{nu} and N_{wg} are defined in (4.1) and (4.2), respectively.

$$N_{nu} = N_{in} + \sum N_{h_i} + N_{out} \tag{4.1}$$

$$N_{wg} = N_{in}N_{h_1} + \sum N_{h_i}N_{h_{i+1}} + N_{h_n}N_{out} \tag{4.2}$$

To simplify the time complexity approximation, we also assume in the back-propagation that the time complexity to compute each hidden unit is identical and represented by T_{BP}. Table 4.2 presents the time complexity of each operation for one iteration in the training. The time complexity for training the MLP is defined in (4.3). Using the trained MLP network, the prediction phase is only as complex as the feed-forward operations. The time complexity of this phase is defined in (4.4).

Table 4.2 Time complexity for MLP training operations

Operation	Time complexity
Feed-forward	$C_{O1} = N_{nu}T_{FW}$
Output Error	$C_{O2} = N_{out}T_{OE}$
Back-propagation	$C_{O3} = (\sum N_{h_i})T_{BP}$
Weights updates	$C_{O4} = N_{wg}T_{WU}$

$$
\begin{aligned}
C(training) &= N_{epoch}N_{te}[C_{O1} + C_{O2} + C_{O3} + C_{O4}] \\
&= N_{epoch}N_{te}[N_{nu}T_{FW} + N_{out}T_{OE} \\
&\quad + (\sum N_{h_i})T_{BP} + N_{wg}T_{WU}] \\
&= N_{epoch}N_{te}[(N_{in} + \sum N_{h_i} + N_{out})T_{FW} \\
&\quad + N_{out}T_{OE} + (\sum N_{h_i})T_{BP} \\
&\quad + (N_{in}N_{h_1} + \sum N_{h_i}N_{h_{i+1}} \\
&\quad + N_{h_n}N_{out})T_{WU}]
\end{aligned}
\tag{4.3}
$$

$$
\begin{aligned}
C(prediction) &= C_{O1} \\
&= N_{nu}T_{FW} \\
&= (N_{in} + \sum N_{h_i} + N_{out})T_{FW}
\end{aligned}
\tag{4.4}
$$

To better visualize how the DNN structure affects the latency, we have generated simulation data containing 100 features, 50 attacks categorized into 10 attack categories, and 200,000 (200k) training examples. This simulation dataset is used solely to illustrate the latency evolution in the training and prediction of models that use DNNs with various structures. To evaluate and compare the models' accuracy, more data are used in the experimentation section. Using the simulation dataset, we trained and evaluated five MLP models of various sizes (M1 to M5). Table 4.3 describes these models and presents the training latency expressed in minutes (mn) and the prediction latency expressed in milliseconds (ms). Figure 4.4 illustrates the latency evolution according to the MLP and data sizes. The same process is applied to evaluate the time complexity of all the other DNNs, *i.e.*, RNN, LSTM, Auto Encoder, and CNN.

In this analysis, it is crucial to point out that the size of the DNNs and consequently their latencies are also determined by the characteristics of the data to be classified, *i.e.*, feature vector

Table 4.3 Training and prediction latency on different MLP and dataset sizes

Model	M1	M2	M3	M4	M5
N_{te}	200k	200k	200k	200k	200k
N_{in}	100	90	80	70	60
h_1	150	100	100	50	50
h_2	150	100	50	50	50
h_3	150	100	100	100	50
N_{out}	50	50	10	10	1
N_{epoch}	30	25	20	15	10
$C(tr)$	**36.5 mn**	**15.8 mn**	**7.4 mn**	**3.7 mn**	**1.7 mn**
$C(pr)$	**201 ms**	**147 ms**	**114 ms**	**94 ms**	**71 ms**

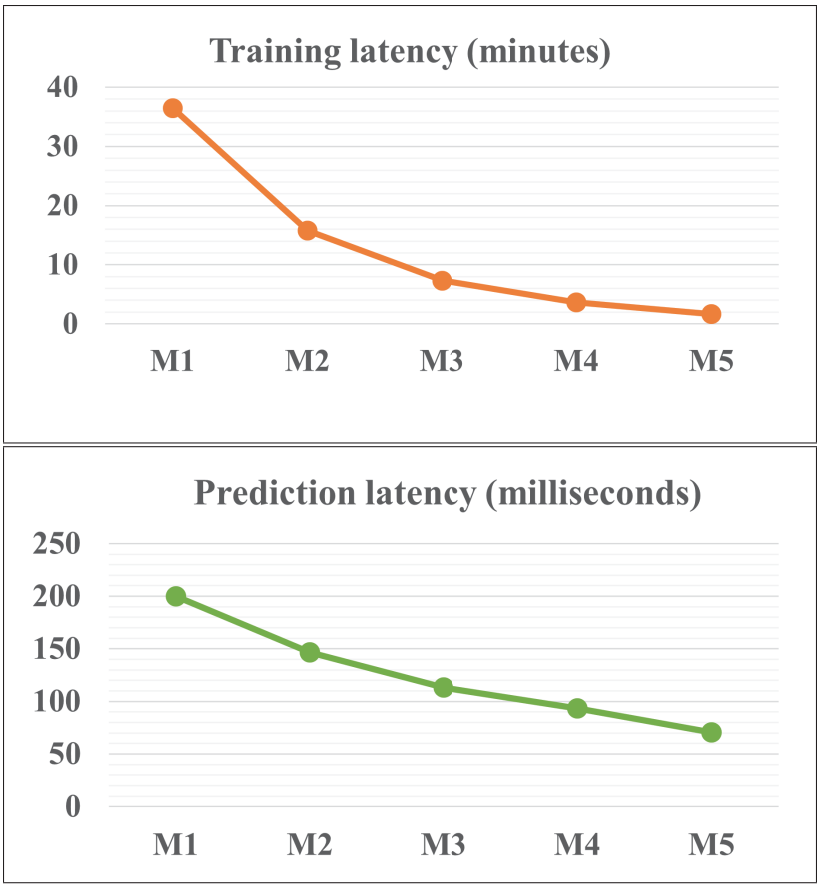

Figure 4.4 Training and prediction latency according to MLP and dataset sizes

size, number of classes, and data distribution. Firstly, the number of features and classes preset the number of neurons in the input and output layers of the DNN, respectively. Secondly, the more complex the data distribution is, the more hidden layers and neurons at each hidden layer are needed in order to shape the decision boundaries. Furthermore, the more complex this data distribution is, the more training examples and iterations on the training examples (number of epochs) are needed.

4.3 Proposed collaborative IDS and deployment architecture

As demonstrated in the previous section, IDSs that employ complex neural networks yield high training and prediction latency. Yet, complex data classification, such as IIoT datasets, generally requires complex DNNs. Specifically, the more classes (N_{out}) we have, the more features (N_{in}), training examples (N_{te}), iterations in the training examples (N_{epoch}), and layers in the DNNs (N_h) are needed to find the best decision boundaries. However, in mission-critical ICSs, intrusion detection requires very low latency. Therefore, to reduce the latency in the proposed IDPS, we employ a technique that we previously designed for heavy ensemble learners [Illy *et al.* (2019); Illy, Kaddoum, de Araujo-Filho, Kaur & Garg (2022a)] and design a classification scheme that splits the problem into two collaborative tasks: anomaly detection and attack classification. Figure 4.5 illustrates the architecture of our collaborative IDPS framework for ICSs.

By performing anomaly detection as a first task, the proposed collaborative IDPS can meet mission-critical IIoT latency requirements. This latency reduction is achieved by reducing most of the variables that impact the time complexity of the DNNs. Firstly, as a binary classification, anomaly detection has fewer neurons at the output layer ($N_{out} = 2$). Secondly, having a less complex decision function than a multi-class classification, binary classification could reach an optimal accuracy with fewer N_h, N_{h_i}, and N_{epoch}. Moreover, binary classification could require fewer features than multi-class classification. Therefore, anomaly detection can benefit from more dimensionality reduction of the features, *i.e.*, reduced N_{in}, (*e.g.*, M5 presented in section II). Finally, being less computationally intensive than a multi-class classification, the anomaly detection model can be deployed in any local server in the ICS stations and benefit from less

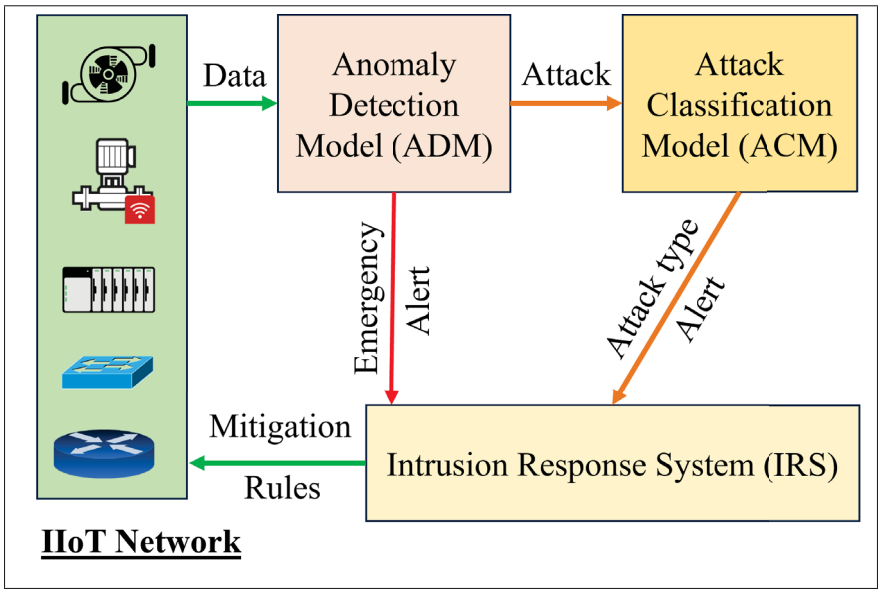

Figure 4.5 General architecture of our collaborative IDPS framework for ICSs

communication latency. When an anomaly is detected, the information is immediately sent to the IRS and the security administrators to apply urgent and emergency response measures. Then, the anomalous traffic is sent to an attack classification model.

The attack classification task is less latency-sensitive than anomaly detection. Therefore, this classification model can employ complex DNNs and be deployed on servers with more computation resources, such as cloud servers. In contrast to binary classification, attack classification could require more features. Thus, for this model, more classification features can be included during the training and prediction to improve the accuracy. When an attack class is predicted, the information is sent to the IRS and the security administrators to apply complementary and more precise prevention mechanisms according to the attack type.

The deployment architecture of our proposed IDPS is illustrated in Figure 4.6. This architecture adopts the innovative SDN and Network Functions Virtualization (NFV) to efficiently implement the IDPS in IIoT networks. In every station of the IIoT network, the sensors and actuators are connected to a PLC through an SDN switch. The data from the switch is collected by a local

server, where anomaly detection is performed. A station may be connected to remote stations through various wide-area network protocols. SDN and NFV enable on-demand provisioning of the network functions, including the IDS and IRS [Li & Chen (2015)]. For the IDS, SDN enables on-demand monitoring of the resources, where the set of resources to be monitored can be dynamically changed [Chin, Mountrouidou, Li & Xiong (2015); Hermosilla, Zarca, Bernabe, Ortiz & Skarmeta (2020)]. In addition, the IDS performance can be improved by advanced monitoring features. For the response module, SDN architectures enable automatic and real-time reactions to block or redirect malicious traffic [Maxli Campos & Martins (2017)]. Moreover, the network's programmability allows management automation and minimizes human intervention and related operational costs. Figure 4.7 details how the IDPS modules operate using the Unified Modeling Language (UML) sequence diagram. This diagram depicts the interaction between the main entities of the IDPS in the order in which these interactions take place. The main entities include the data collection gates (SDN switches), the system orchestrator, the Anomaly Detection Model (ADM), the Attack Classification Model (ACM), the IRS, and the SDN controller.

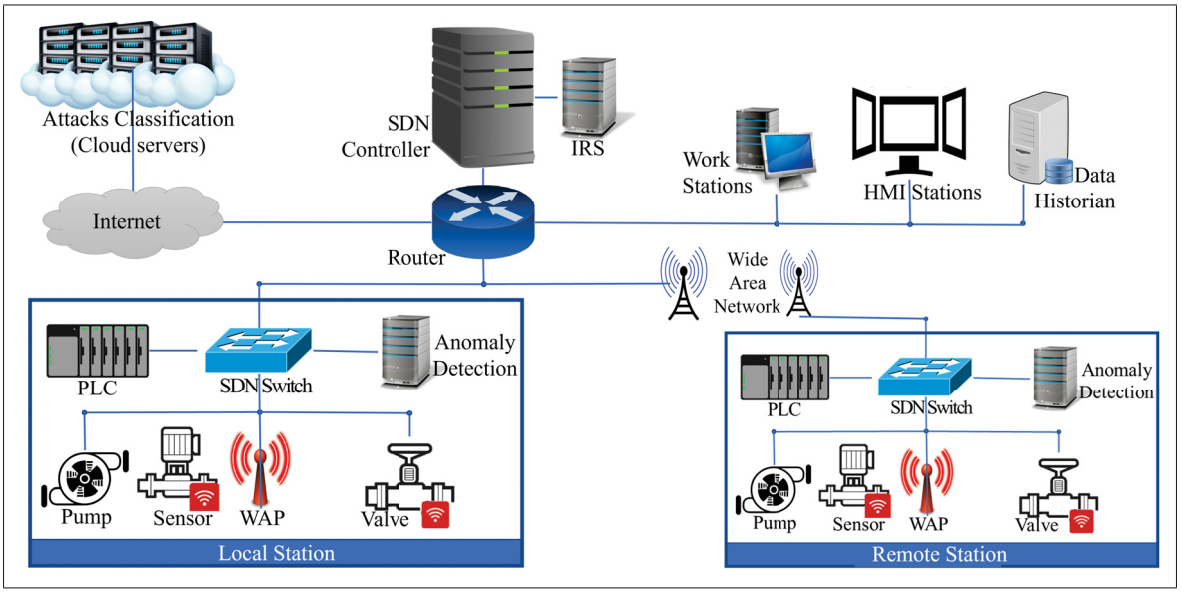

Figure 4.6 The proposed IDPS architecture in the ICS

This collaborative IDPS provides a faster intrusion detection system with attack classification for efficient response. In this architecture, only anomalous traffic needs to go through the complex

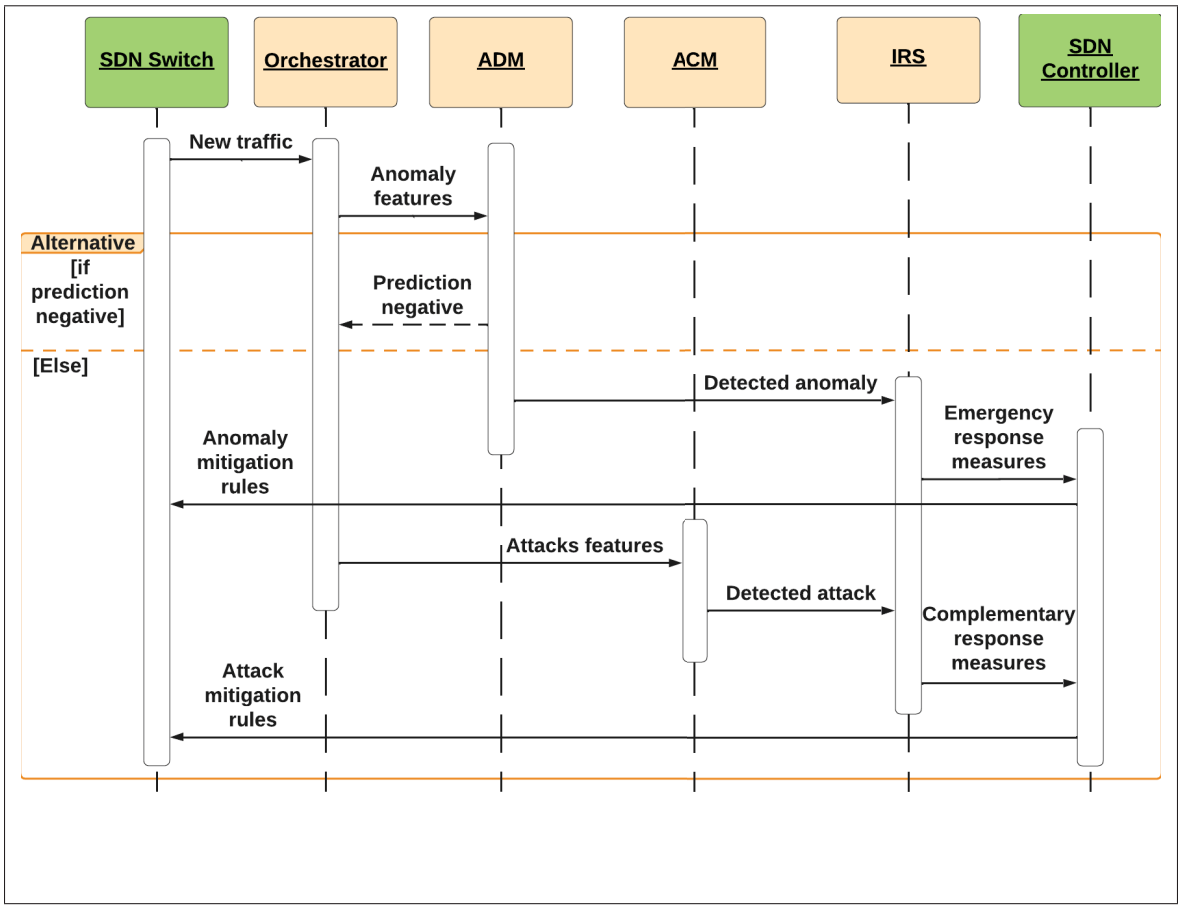

Figure 4.7 The UML sequence diagram of the proposed IDPS

attack classification DNN, which represents an extremely small portion of the IIoT data. The next section presents the detection features selection, the implemented deep learning models, and the response measures.

4.4 Proposed detection features, learning model, and IRS measures

4.4.1 Proposed detection features

The choice of the feature set is challenging in every machine learning solution. It is more convenient in some applications to gather all the features that are available (the initial feature set), then use dimensionality reduction methods to select the best features (the final feature

set). For instance, with image processing, it is feasible to take all pixels instead of studying and identifying each relevant feature. However, an initial feature set is not directly apparent in some applications. In intrusion detection for IIoT applications, feature selection is a more complex task regarding the multiplicity and variety of elements involved in cyber-physical systems (functional organization, software, operational principles, network workload, protocols, hardware, power consumption, etc.). Thus, it is crucial to study and identify relevant features and constitute an initial feature set [Illy, Kaddoum, Kaur & Garg (2022b)]. Then, the dimensionality reduction methods can be applied to keep the best features. In this study, we select the detection features according to the attacks' behavior and the IIoT network's characteristics. Algorithm 4.2 describes the methodology adopted for feature selection. This is a complex task but convenient as it can clearly explain the role of each feature used in the IDS. Table 4.4 presents all the proposed detection features and their roles.

Algorithm 4.2 IIoT Features selection logic

```
1  Output: Relevant feature set (Relevant_Features).
2  Input 1: Targeted IIoT cyber-threats (Targeted_cyber_threats).
3  Input 2: All IIoT system features (All_IIoT_Features).
4  foreach attack_x in Targeted_cyber_threats do
5      foreach feature_y in All_IIoT_Features do
6          if feature_y is changed by attack_x then
7              Add feature_y in Relevant_Features
8          end
9      end
10 end
11 Delete redundancy in Relevant_Features
12 Delete correlation in Relevant_Features
13 Reduce dimensionality of Relevant_Features
```

4.4.2 Proposed detection models

Multiple DNN algorithms have demonstrated excellent capabilities in building accurate models for different real-life problems. In spite of this, there is no specific DNN that will produce the most accurate learner for every dataset. Each DNN focuses on particular aspects of the data

Table 4.4 Relevant features for IIoT IDSs

Features	**Definition**	**Role**
Time-based traffic features	Statistics from TCP/IP traffic flows using a predefined time intervals	Detection of Scanning, Probing, and DoS attacks that operate with many connections in short time intervals
Connection-based traffic features	Statistics from TCP/IP traffic flows based on a window of a predefined number of connections [Illy *et al.* (2022b)]	Detection of Scanning, Probing, and DoS attacks that employ large time intervals
Packet header features	Metadata portions of TCP/IP packets, such as IP addresses, port numbers, services, and flags [Anthi *et al.* (2019)]	Detection of SYN flood, Man-In-The-Middle (MITM), and ICMP/UDP/TCP fragmentation attacks
Content features	Information contained in the TCP/IP packet payload [Abusitta *et al.* (2019); Iglesias & Zseby (2015); Wang *et al.* (2017); Xu *et al.* (2016); Zhong *et al.* (2020)], *e.g.*, number of failed login attempts	Detection of Remote to Local (R2L) and User to Root (U2R) attacks
Wireless communication features	Radio signal characteristics, such as Received Signal Strength (RSS), Signal-to-noise ratio (SNR), distance of the radio transmitter, Radio-frequency fingerprint (RFF) [Ramsey, Mullins, Temple & Grimaila (2015); Soltanieh, Norouzi, Yang & Karmakar (2020); Tian *et al.* (2019,1)]	Detection of attacks that involve wireless devices, including jamming, deauthentication, spoofing, and contamination

distribution to build the models. For example, CNNs learn spatial relations between separate features and perform at their best when dealing with data with a grid-like topology, such as images. Meanwhile, RNNs and LSTMs are specialized in learning temporal or sequential information and are more suitable for temporal data, such as text and speech analysis. In intrusion detection, each DNN focuses on some part of the data patterns. Therefore, it is necessary to implement and evaluate different DNNs and deploy only those that produce the best detection models for the specified data. Thus, this work studies and implements the most promising DNNs, including CNN, RNN, LSTM, and MLP. The anomaly detection model may employ a simple DNN, such as MLP; however, the attack classification model uses a more complex DNN, such as RNN, LSTM, and CNN.

4.4.3 Proposed response measures

The success of some attacks depends mainly on the time gap between the detection and the defensive response against the attack. Intrusion Response Systems (IRSs), also referred to as Intrusion Prevention Systems (IPSs), launch counteractions automatically when attacks are detected by the IDS, in order to defend the targeted system. Compared to IDSs that just generate reports or alarms, IRSs reduce the vulnerability window between when an intrusion is detected and when defensive actions are taken. Therefore, this work proposes an emergency and quasi-real-time response when an anomaly is detected. Then, it classifies this anomaly to identify suitable complementary measures according to the attack type.

The basic mitigation procedure is to block the packets involved in the reported intrusion and reject all incoming and outgoing traffic of the malicious device (*e.g.*, using blacklist or whitelist MAC filtering). This can stop many attacks, including R2L and U2R, and safeguard the ICS's confidentiality and integrity. This emergency measure can also maintain the system's services by reducing scanning, probing, and DoS attacks. It is also possible to deflect and redirect the malicious users into a Honeypot or other attack analysis systems to gain more information on the way they are operating [Matin & Rahardjo (2020); Tian *et al.* (2020)]. These preventative measures are implemented in the SDN-based deployment architecture. For instance, when the IRS receives an anomaly message from the ADM, the IRS uses the information contained in the received message to build the set of SDN rules and send it to the SDN controller (Figure 4.7). Then, the controller transmits those rules to the forwarding devices' flow tables to drop the concerned packets or forward them to an attack analysis system. Likewise, when the IRS receives an attack message from the ACM, it uses this information to build the set of SDN rules and send it to the controller. The SDN controller transmits this routing information to the forwarding devices' flow tables to mitigate the attack. For further response, it is possible to develop an additional module that can counterattack the attacking entity to neutralize or attenuate its impact.

The complementary response measures are defined according to each specific attack type. For example, to overcome a detected jamming attack, communication bandwidth can be switched to other frequencies. To realize that, Dynamic Channel Selection (DCS) can be implemented [Navda *et al.* (2007)]. The DCS enables the wireless transmitters to monitor the interference level, and when it exceeds the predefined DCS threshold, the wireless transmitters stop operating on that channel. Then, the wireless access point (WAP) uses automatic channel selection to determine an alternative channel to switch the communication. Furthermore, the robustness of the legitimate signal can be increased in order to maintain secure wireless communication during jamming attacks. For advanced defence, tracking systems can be employed to locate the jamming station and terminate it [Aldosari *et al.* (2019)]. These technologies are also effective against other active wireless attacks, including contamination and spoofing.

4.5 Experimentation

This section presents the experiment and discusses the results. The experimentation aims to evaluate the performance of the DNN-based collaborative IDS. Therefore, we exploit three datasets that present different challenges.

The first dataset, WUSTL-IIOT-2018, presented in [Alem & BB (2021); Teixeira *et al.* (2018)], is used for SCADA cyber-security research. The dataset was built using a SCADA system test-bed. To generate this data, scan tools were used to inspect the topology of the victim network and identify the devices in the network, as well as their vulnerabilities. The attacks carried out against the test-bed include Port Scanner, Address Scan Attacks, Device Identification Attacks, and Exploit. All network traffic (normal and abnormal traffic) was monitored by the Audit Record Generation and Utilization System (ARGUS) tool. The traffic captured comprises 7,049,989 observations, with 93.93% being normal traffic (without attacks) and 6.07% being abnormal traffic (attacked traffic). The raw data has 25 networking features, where some features are used to classify the data, while others are used to train and test machine learning algorithms. After the data cleaning process and dimensionality reduction, a Comma-Separated Values (CSV) file

containing 7,037,983 observations (vectors in the dataset) and five features were provided. Each vector in the dataset is labelled as normal or attack, depending on the case.

The second dataset, NSL-KDD, was built from a regular network (not an IIoT); however, it is one of the most complete, realistic, and challenging datasets available, which is used to compare machine learning-based IDSs [Tavallaee *et al.* (2009)]. NSL-KDD includes a total of 39 specific attacks regrouped into four different attack categories, namely, probing (Probe), DoS, R2L, and U2R. The training dataset (KDDTrain+) and testing dataset (KDDTest+) contain 125,973 and 22,544 single connection vectors, respectively. Each connection vector is represented by 41 features and labelled as either normal or attack, with exactly one specific attack type. This dataset is also available in a CSV file.

The third dataset, UNSW-NB15, presented in [Moustafa & Slay (2015a,1)] was also built from a regular network; however, unlike the NSL-KDD dataset, it contains a hybrid of the modern normal and abnormal network traffic. To generate this data, the IXIA PerfectStorm tool was utilized in the cyber range lab of the Australian Centre for Cyber Security. UNSW-NB15 includes a total of 205 specific attacks regrouped into nine attack categories, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. The dataset, provided in a CSV file, contains a total of 2,540,044 records, each represented by 47 features and labelled as either normal or attack, including the specific attack category.

The experimentation employs four different DNNs to build the detection models. These DNNs include MLP, Simple RNN, LSTM, and CNN. The implementation is conducted in the anomaly detection and the attacks classification stages. The machine learning platform used to train and evaluate the models is scikit-learn (sklearn), a free software library for the Python programming language, installed with Anaconda, an open-source distribution for Python and R.

4.5.1 The anomaly detection model

We begin the implementation with the anomaly detection model. Table 4.5 presents the distribution of the instances in the binary WUSTL-IIOT-2018, NSL-KDD, and UNSW-NB15 datasets.

Table 4.5 Instances distribution in the binary WUSTL-IIOT-2018, NSL-KDD, and UNSW-NB15 datasets

Class	Number of instances			
	WUSTL-IIOT	***KDDTrain+***	***KDDTest+***	***UNSW-NB15***
Normal	6,610,778	67,343	9,711	2,218,761
Attack	427,205	58,630	12,833	321,283
Total	**7,037,983**	**125,973**	**22,544**	**2,540,044**

For each DNN model, we tuned the hyper-parameters, such as the number of hidden layers, number of units for each hidden layer, learning rates, activation and solver functions, number and size of the CNN filters, and number of epochs. For WUSTL-IIOT-2018 and UNSW-NB15, we split the dataset into training (80%) and testing (20%) sets. For the NSL-KDD, we trained the models using KDDTrain+ and evaluated them with KDDTest+. Figure 4.8, Figure 4.9, and Figure 4.10 present the accuracy of the models on the WUSTL-IIOT-2018, NSL-KDD, and UNSW-NB15 datasets, respectively. For each model, the accuracy presented is obtained from tuned hyper-parameters.

The experimentation results with the WUSTL-IIOT-2018 and UNSW-NB15 datasets confirm that anomaly detection does not require complex DNNs to produce a high-accuracy prediction model. In both datasets, lightweight DNNs (*e.g.*, MLP with one hidden layer of 100 neurons) were able to reach over 99% detection accuracy. Such DNNs can be easily trained and deployed on a local server for ultra-low latency intrusion detection and response. In our local server, the training time takes less than 15 minutes and 6 minutes for WUSTL-IIOT-2018 and UNSW-NB15, respectively. The prediction time for the test datasets takes approximately 0.7 seconds (WUSTL-IIOT-2018) and 0.4 seconds (UNSW-NB15). However, for the NSL-KDD dataset, the models show a lower accuracy (less than 87%). This can be explained by the fact that this dataset was built

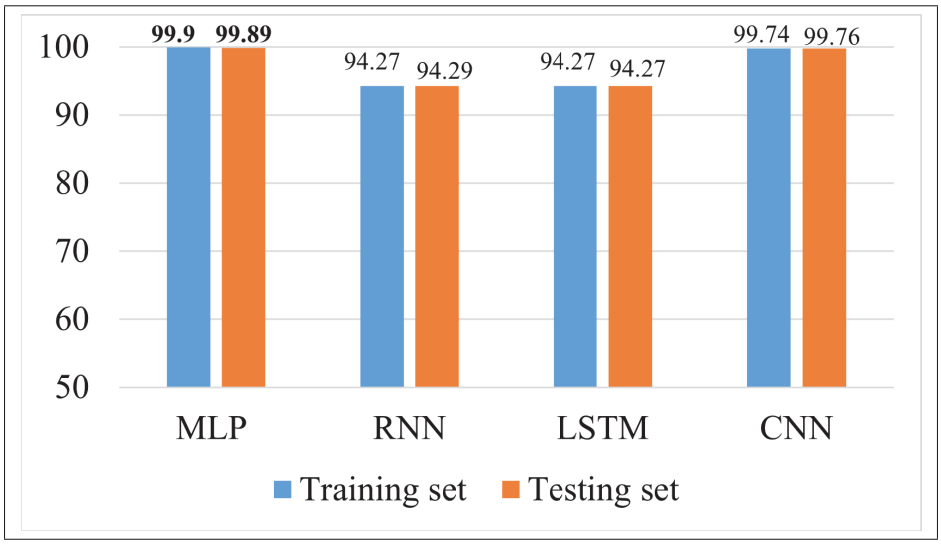

Figure 4.8 Accuracy of anomaly detection models using binary-labelled WUSTL-IIOT-2018 dataset

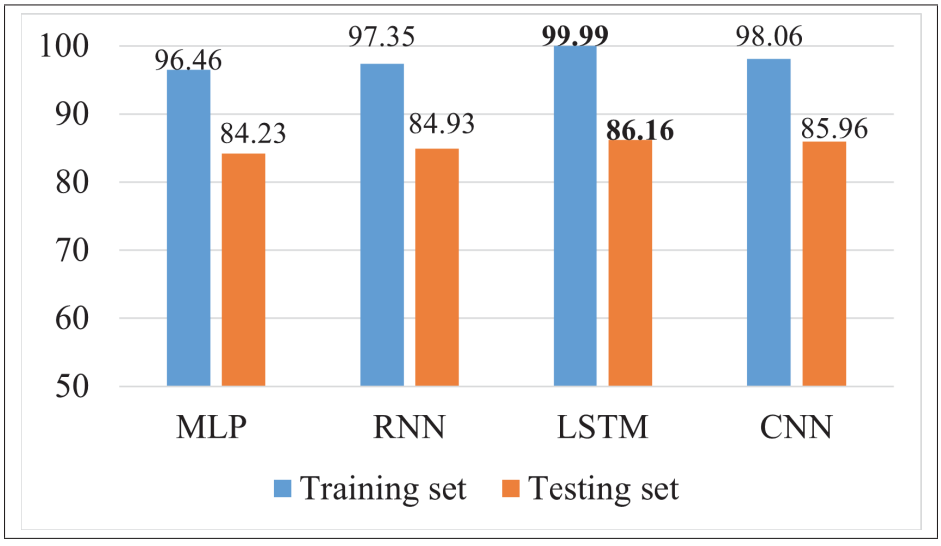

Figure 4.9 Accuracy of anomaly detection models using binary-labelled NSL-KDD dataset

deliberately from hard-to-classify samples. Therefore, the classification in this dataset presents more challenges compared to other datasets.

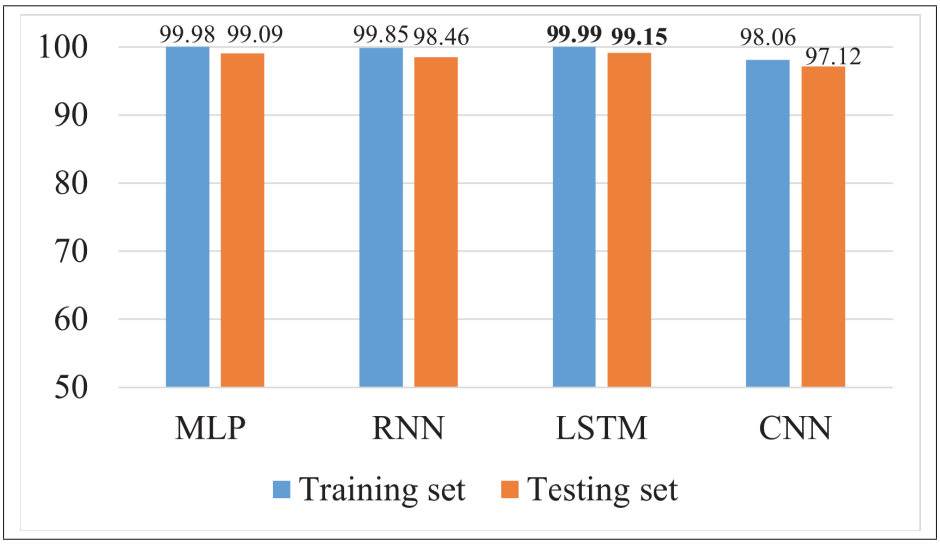

Figure 4.10 Accuracy of anomaly detection models using binary-labelled UNSW-NB15 dataset

4.5.2 The attack classification model

The second part of the experimentation concerns the attacks classification model. Since the published WUSTL-IIOT-2018 dataset does not indicate the attack types, we only use the NSL-KDD and UNSW-NB15 datasets. Table 4.6 and Table 4.7 present these datasets' organization for the multi-class classification.

Table 4.6 Attacks instances distribution in the NSL-KDD dataset

	Number of instances	
Class	***KDDTrain+***	***KDDTest+***
DOS	45,927	7,458
Probe	11,656	2,421
R2L	995	2,754
U2R	52	200
Total	**58,630**	**12,833**

The same DNNs, *i.e.*, MLP, Simple RNN, LSTM, and CNN, are used to build the attack classification models. Figure 4.11 and Figure 4.12 illustrate the accuracy of the models on

Table 4.7 Attacks instances distribution in the UNSW-NB15 dataset

Class	**Number of instances**
Fuzzers	24,246
Analysis	2,677
Backdoors	2,329
DoS	16,353
Exploits	44,525
Generic	215,481
Reconnaissance	13,987
Shellcode	1,511
Worms	174
Total	**321,283**

NSL-KDD and UNSW-NB15 datasets, respectively. For every model, the presented accuracy is obtained from the tuned hyper-parameters.

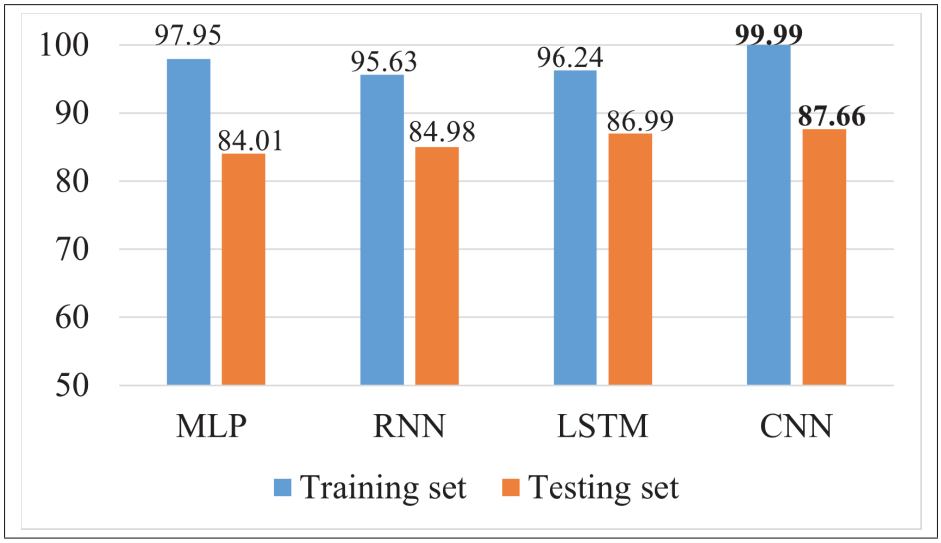

Figure 4.11 Accuracy of attack classification models using categorized-labelled NSL-KDD dataset

This experiment confirms that the attack classification may require complex DNNs and a long training time. With NSL-KDD, the best accuracy of 87.66% was provided by the CNN model, followed by the LSTM model with 86.99% accuracy. Compared to previous works that employed the NSL-KDD dataset, the proposed collaborative DNN-based IDS shows better prediction

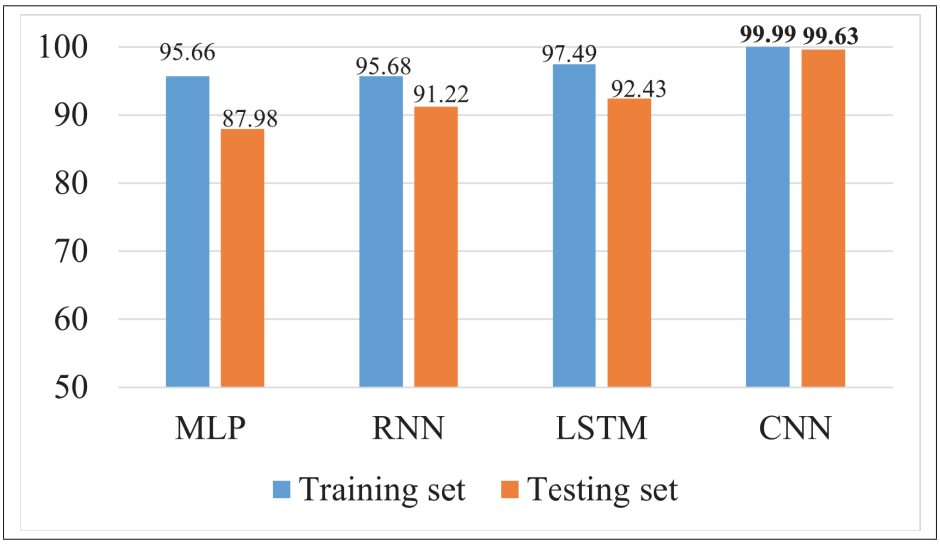

Figure 4.12 Accuracy of attack classification models using categorized-labelled UNSW-NB15 dataset

accuracy. Table 4.8 compares these previous solutions with our attack classification model (Four-category classification). With UNSW-NB15, the best accuracy of 99.63% was also provided by the CNN model, followed by the LSTM model with 92.43% accuracy. Compared to previous works that employed the UNSW-NB15 dataset, the proposed collaborative DNN-based IDS shows again better prediction accuracy. Table 4.9 compares these previous solutions with our attack classification model (nine-category classification). In our experimentation environment, the training of the attack classification models can take several hours, and the prediction takes approximately 10 seconds and 3 minutes 36 seconds for NSL-KDD and UNSW-NB15 test datasets, respectively. However, as this task can be done in the cloud, the complexity can be handled thanks to the availability of more computation resources.

This experiment demonstrates that this approach is efficient for intrusion detection in mission-critical smart factories. The anomaly detection is performed locally with a lightweight DNN, which offers an ultra-low latency IDS for fast response in the IRS. Then, a larger DNN deployed in the cloud provides a robust attack classification for more precise responses in the IRS, *i.e.*, responses that launch adequate mitigation measures according to the category of the detected attacks.

Table 4.8 Comparison with solutions that exploits the NSL-KDD dataset for multi-class classification

Methods used	**Accuracy**
Proposed collaborative deep learning-based IDS	87.66 %
Methods in [Tavallaee *et al.* (2009)]: - NBTree - Random Tree - Decision Trees J48 - Random Forest - MLP	- 82.02 % 81.59 % 81.05 % 80.67 % 77.41 %
Methods in [Bajaj & Arora (2013)]: - SimpleCart - MLP	- 82.32 % 73.54 %
RNNs [Yin *et al.* (2017)].	81.29 %
ANN with tansig transfer function,Levenberg-Marquardt (LM) and BFGS quasi-Newton Back-propagation [Ingre & Yadav (2015)]	81.20 %
A Two-Layer Dimension Reduction and Two-Tier Classification Model [Pajouh *et al.* (2019)]	84.86 %
Two-stage intrusion detection technique combining Naive Bayes and k-means [Vishwakarma & Kesswani (2022)]	86.46 %

4.6 Conclusion

A deep learning-based IDS produces better predictions for future networks, but the more complex the neural network structure, the greater its impact on latency. To leverage these innovative learning models and deal with the latency requirement of ICSs, this work introduced a time complexity analysis of the DNN algorithms to highlight the variables with the most impact on the training and prediction latency. Based on this analysis, a collaborative DNN-based IDPS that employs two classification models is proposed. The first model employs a lightweight DNN that performs low latency anomaly detection, *i.e.*, a simple binary classification. This lightweight DNN is deployed on local servers to allow faster threat detection and emergency response. The second model performs attack classifications of the anomalous traffic to guide the intrusion response tasks. This second classifier can be deployed in the cloud to benefit from more computation resources to run more complex DNNs.

Table 4.9 Comparison with solutions that exploits the UNSW-NB15 dataset for multi-class classification

Methods used	**Accuracy**
Proposed collaborative deep learning-based IDS	99.63 %
Methods in [Moustafa & Slay (2016)]: - Decision Tree - Logistic Regression - Naïve Bayes - Artificial Neural Network - Expectation-Maximisation Clustering	- 85.56 % 83.15 % 82.07 % 81.34 % 78.47 %
Random Forest using different subsets of features [Janarthanan & Zargari (2017)]: - Subset 2 - Subset 1	- 81.61% 75.66 %
Deep neural network models in [Čavojský, Bugár & Levický (2023)]: - Dense-layer - LSTM	- 79.34% 79.21 %
Extreme gradient boosting (XGBoost) [Husain, Salem, Jim & Dimitoglou (2019)]	75.88 %
Support Vector Machine [Jing & Chen (2019)]	75.77 %

Moreover, an SDN-based deployment architecture of the proposed collaborative IDPS in ICS networks was presented. This architecture provided an efficient implementation of the IDPS with key innovative features, such as on-demand resources monitoring on the IDS and real-time response on the IRS. Furthermore, this work proposed various detection features, response measures, and implemented different learning methods, including CNN, LSTM, RNN, and MLP. The experimentation, which was performed on three datasets with different challenges, demonstrated the efficiency of the proposed approach. This can be further improved by investigating more DNNs in our future research.

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion and Lessons Learned

The global IoT market size, valued at 380 billion U.S. dollars (USD) in 2021, is projected to grow from 478 billion USD in 2022 to 2.4 trillion USD by 2029, exhibiting a compound annual growth rate (CAGR) of 26.4% during the forecast period [Al-Sarawi, Anbar, Abdullah & Al Hawari (2020); Bhuiyan, Rahman, Billah & Saha (2021); Taşıran (2019)]. This growth will be fueled by emerging technologies, like low-power wireless communication protocols, SDN, cloud/edge computing, and 5G high-speed and massive connectivity. Meanwhile, security risks for IoT applications are becoming increasingly critical, with a large attack surface, numerous vulnerabilities, and a diverse and increasing number of sophisticated attacks. This is particularly concerning since most traditional ICT security mechanisms are not suitable to IoT specifications and resource constraints. In this context, numerous research works are aiming to provide suitable security solutions to protect IoT applications, and IDPSs, as an essential security layer, have become one of the most active IoT research domains.

From a full cyber-security framework perspective, intrusion detection is the second main security layer which provides real-time events monitoring, attack detection and alerts to the intrusion prevention layer, which can automatically apply response measures to block attacks. Indeed, IDPSs are empowering the security of many applications by enabling the detection and mitigation of intrusion attempts (successful or failed) to violate systems security principles. In particular, this thesis targeted the design of an intelligent IDPS that detects a large variety of IoT attacks with high accuracy in a deployment model that reduces the detection latency and facilitates the systems' scalability. In this vein, this thesis proposed promising collaborative IDPS approaches for future IoT applications. Specifically, the contributions of the thesis are summarized as follows:

- In Chapter 2, we proposed various detection features that can enhance ML-based IDSs. We evaluated the efficiency of the proposed features using different ML algorithms. The experimentation demonstrated the predominant role of feature identification on the ML models' performance. Particularly, the results showed that traffic features are more effective in detecting probing and DoS attacks, while content features are more effective in detecting U2R and R2L attacks. Combining these features with other relevant features, such as TCP/IP packet headers and features based on wireless communication significantly improves ML-based IDSs' ability to detect a wide range of attacks with high accuracy.

- In Chapter 3, we proposed a hybrid multistage DNN-based IDPS for high-risk IoT applications. In the training phase, each new stage used a different DNN and was trained with the data that were misclassified or classified with low confidence by the previous stage's DNN. This allowed new stage DNNs to focus on the limitations of the previous stages. At the prediction phase, the resulting multistage DNN used each stage's decision in a function to produce final decisions. The experimental evaluation showed that this approach offers better classification accuracy at the cost of increased latency and is much more suitable for high-risk IoT applications that cannot afford to trade off security for latency.

- Chapter 4 provided a collaborative DNN-based low-latency IDPS for mission-critical IoT applications. We conducted a time complexity analysis of neural networks and designed a deep learning-based collaborative IDPS that employed two levels of classifications. The first level performed a lightweight DNN-based anomaly detection in local servers to allow fast attack detection and response. The second level performed the classifications of the anomalous traffic in cloud servers to guide the intrusion response tasks. In addition, an SDN-based deployment architecture of the proposed collaborative IDPS in ICSs was provided. Experimentations demonstrated the efficiency of the proposed approach in terms of detection accuracy and latency.

As a summary, according to the conducted research, valuable lessons we learned are listed as follows:

- The detection features identification and extraction are fundamental for intrusion detection in IoT. Fully exploring and exploiting relevant detection features can improve the ability to detect a large variety of attacks with high accuracy.
- Ensemble and collaborative machine learning models are especially promising as each model can focus on part of the data patterns to provide accurate classification while the rest of the patterns are learned by other models.
- A distributed deployment architecture of the detection models where edge-based and cloud-based modules are combined can significantly improve the detection latency.
- The performance of the IPS can be enhanced by leveraging SDN-based deployment architecture as it enables the network to be intelligently and centrally controlled and automatically reprogrammed and reconfigured to mitigate detected attacks.

Author's Publications

The outcomes of the author's Ph.D. research are the articles listed below, published in IEEE journals.

P. Illy, G. Kaddoum, K. Kaur and S. Garg, "ML-Based IDPS Enhancement With Complementary Features for Home IoT Networks," in IEEE Transactions on Network and Service Management, vol. 19, no. 2, pp. 772-783, June 2022, doi: 10.1109/TNSM.2022.3141942.

P. Illy, G. Kaddoum, P. F. de Araujo-Filho, K. Kaur and S. Garg, "A Hybrid Multistage DNN-Based Collaborative IDPS for High-Risk Smart Factory Networks," in IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 4273-4283, Dec. 2022, doi: 10.1109/TNSM.2022.3202801.

P. Illy and G. Kaddoum, "A Collaborative DNN-Based Low-Latency IDPS for Mission-Critical Smart Factory Networks," in IEEE Access, vol. 11, pp. 96317-96329, Sept. 2023, doi: 10.1109/ACCESS.2023.3311822.

Beside the above articles that contribute to the main contents of this dissertation, other publications that the author was involved in and which are not included in this dissertation are:

M. Hachimi, G. Kaddoum, G. Gagnon and P. Illy, "Multi-stage Jamming Attacks Detection using Deep Learning Combined with Kernelized Support Vector Machine in 5G Cloud Radio Access Networks," 2020 International Symposium on Networks, Computers and Communications (ISNCC), 2020, pp. 1-5, doi: 10.1109/ISNCC49221.2020.9297290.

P. Illy, G. Kaddoum, C. Miranda Moreira, K. Kaur and S. Garg, "Securing Fog-to-Things Environment Using Intrusion Detection System Based On Ensemble Learning," 2019 IEEE Wireless Communications and Networking Conference (WCNC), 2019, pp. 1-7, doi: 10.1109/WCNC.2019.8885534.

5.2 Recommendations: Intrusion Detection and Prevention for 6G IoT Networks

The sixth generation of wireless technology (6G) envisions the realization of the internet of everything (IoE), a collection of billions of heterogeneous devices operating in a connected intelligence [Alwis, Kalla, Pham, Kumar, Dev, Hwang & Liyanage (2021); Saad, Bennis & Chen (2020); Siriwardhana, Porambage, Liyanage & Ylianttila (2021)]. Some fundamental security models will not be practical for the IoE in 6G, especially with the small form factor devices, such as in-body sensors. For instance, key distribution and management functions may be highly inefficient in massive networks [Wang, Wang, Alipour-Fanid, Jiao & Zeng (2019)]. The resource-constrained IoT devices, which cannot afford complicated cryptography to maintain strong security, will remain a primary target for attackers. Moreover, new attack models are emerging continuously with increased levels of sophistication and evasion, especially AI-powered cyber attacks [Guembe, Azeta, Misra, Osamor, Fernandez-Sanz & Pospelova (2022); Kaloudi & Li

(2020); Veprytska & Kharchenko (2022)]. These challenges will require security approaches that can provide proactive IDPSs for threat discovery and the application of intelligent mitigation techniques. Regarding these requirements and challenges, some potential directions for future research are discussed below.

5.2.1 ML Techniques to Overcome the Lack of Learning Datasets

The lack of labelled datasets remains a major challenge in learning-based IDSs, since obtaining sufficient labelled samples is cumbersome and requires the efforts of domain experts [Aouedi, Piamrat, Muller & Singh (2022,2); Huang, Yang & Gong (2022)]. Another related challenge is the lack of attack samples, which generally causes highly imbalanced datasets with significant drawbacks in the learning performance [Nie, Wu, Wang, Guo, Wang, Gao & Li (2022); Park, Lee, Kim, Park, Kim & Hong (2022)]. To overcome these limitations, semi-supervised learning and generative adversarial networks (GANs) can play a key role in providing more learning data. Semi-supervised learning addresses the challenge by considering large amounts of unlabeled samples together with the labelled samples to build the detection models. Meanwhile, GANs can solve the insufficient samples problem, especially the attack samples. The generator in GANs can theoretically generate data, which can greatly expand the diversity of samples.

5.2.2 Network Softwarization for Effective Intrusion Prevention

Network softwarization technologies in 5G, such as SDN, NFV, and network slicing are still applicable for 6G systems; thus, their security benefit will remain in 6G [Abdulqadder & Zhou (2022); Hashima, Fadlullah, Fouda, Mohamed, Hatano, ElHalawany & Guizani (2022); Siriwardhana *et al.* (2021)]. These paradigms can introduce new security enablers, such as real-time adaptive reconfiguration of the network to tackle intrusion attempts. Specifically, thanks to the SDN centralized controller with real-time feedback control capability and the greater programmability of the network, it will be possible to implement proactive IPSs with intelligent intrusion response

capabilities to increase the level of IoT systems' protection. Moreover, NFV has the potential to enable flexible and on-demand deployment of virtual network security functions to optimize IoT systems' protection.

5.2.3 Blockchain-based Collaborative IDPSs

In collaborative IDPSs, the modules need to exchange data with each other. However, data and trust management remain two challenges for current IDPS architectures, which may degrade the effectiveness of such security systems. Blockchain technology is a decentralized and distributed ledger that enables recording transactions across a set of nodes [Monrat, Schelén & Andersson (2019); Samaniego, Jamsrandorj & Deters (2016)]. It can be implemented in a peer-to-peer network without the need for a trusted third party. Blockchain technology has shown its adaptability in many fields, such as edge computing, spectrum sharing, federated learning, and network virtualization [Hewa, Gür, Kalla, Ylianttila, Bracken & Liyanage (2020); Nguyen, Tran, Loven, Partala, Kechadi & Pirttikangas (2020)]. As blockchain can protect the integrity of data storage and ensure process transparency, it has the potential to be applied to collaborative intrusion detection and prevention models [Li & Meng (2022); Liu, Zhang, Zhang, Zhou, Shao, Pu & Zhang (2021); Meng, Tischhauser, Wang, Wang & Han (2018)].

5.2.4 Reinforcement Learning and Game Theory for Intelligent IDPS

Due to the dynamic nature of IoT topologies, IDPSs face numerous challenges, especially in balancing accuracy and efficiency. In this context, reinforcement learning and game theory can enhance IDPSs' perception of the environmental change and their adaptability to different scenarios [Hammar & Stadler (2022); Liang, Ma & Tan (2022)]. Moreover, reinforcement learning can automatically find security strategies for intrusion prevention. This approach can enable evolutive attack defence strategies without human intervention [Hammar & Stadler (2020)]. Furthermore, game theory can help optimize the system by switching some modules'

status to active or idle to reduce the IDPS's energy consumption, overhead, and detection time [Arisdakessian, Wahab, Mourad, Otrok & Guizani (2022); Han, Zhou, Jia, Dalil & Xu (2019)].

LIST OF REFERENCES

Aartech. (2019, January, 3). Smart Home Automation [html]. Consulted at https://www.aartech.ca/home-automation.html.

Abate, A. F., Cimmino, L., Cuomo, I., Nardo, M. D. & Murino, T. (2022). On the Impact of Multimodal and Multisensor Biometrics in Smart Factories. *IEEE Transactions on Industrial Informatics*, 18(12), 9092-9100. doi: 10.1109/TII.2022.3178376.

Abdel-Basset, M., Hawash, H., Chakrabortty, R. K. & Ryan, M. J. (2021). Semi-Supervised Spatiotemporal Deep Learning for Intrusions Detection in IoT Networks. *IEEE Internet of Things Journal*, 8(15), 12251-12265. doi: 10.1109/JIOT.2021.3060878.

Abdulqadder, I. H. & Zhou, S. (2022). SliceBlock: Context-Aware Authentication Handover and Secure Network Slicing Using DAG-Blockchain in Edge-Assisted SDN/NFV-6G Environment. *IEEE Internet of Things Journal*, 9(18), 18079-18097. doi: 10.1109/JIOT.2022.3161838.

Abusitta, A., Bellaiche, M., Dagenais, M. & Halabi, T. (2019). A deep learning approach for proactive multi-cloud cooperative intrusion detection system. *Future Generation Computer Systems*, 98, 308–318.

Ahmad, I., Kumar, T., Liyanage, M., Okwuibe, J., Ylianttila, M. & Gurtov, A. (2018). Overview of 5G Security Challenges and Solutions. *IEEE Communications Standards Magazine*, 2(1), 36-43. doi: 10.1109/MCOMSTD.2018.1700063.

Aitken, R., Chandra, V., Myers, J., Sandhu, B., Shifren, L. & Yeric, G. (2014). Device and technology implications of the Internet of Things. *2014 Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers*, pp. 1-4. doi: 10.1109/VLSIT.2014.6894339.

Al-Abassi, A., Karimipour, H., Dehghantanha, A. & Parizi, R. M. (2020). An ensemble deep learning-based cyber-attack detection in industrial control system. *IEEE Access*, 8, 83965–83973.

Al-Rabiaah, S. (2018). The "Stuxnet" virus of 2010 as an example of a "APT" and its "Recent" variances. *2018 21st Saudi Computer Society National Computer Conference (NCC)*, pp. 1–5.

Al-Sarawi, S., Anbar, M., Alieyan, K. & Alzubaidi, M. (2017). Internet of Things (IoT) communication protocols: Review. *2017 8th International Conference on Information Technology (ICIT)*, pp. 685-690. doi: 10.1109/ICITECH.2017.8079928.

Al-Sarawi, S., Anbar, M., Abdullah, R. & Al Hawari, A. B. (2020). Internet of Things Market Analysis Forecasts, 2020–2030. *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pp. 449-453. doi: 10.1109/WorldS450073.2020.9210375.

Aldosari, W., Zohdy, M. & Olawoyin, R. (2019). Tracking the Mobile Jammer in Wireless Sensor Networks Using Extended Kalman Filter. *2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pp. 0207-0212. doi: 10.1109/UEMCON47517.2019.8993050.

Alem, S. & BB, B. E. M. (2021). New Dataset for Industry 4.0 to Address the Change in Threat Landscape. *Risks and Security of Internet and Systems: 15th International Conference, CRiSIS 2020, Paris, France, November 4–6, 2020, Revised Selected Papers 15*, pp. 273–288.

Alhakbani, N., Hassan, M. M., Hossain, M. A. & Alnuem, M. (2014). A framework of adaptive interaction support in cloud-based internet of things (IoT) environment. *International conference on internet and distributed computing systems*, pp. 136–146.

Alkadi, O., Moustafa, N., Turnbull, B. & Choo, K.-K. R. (2021). A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks. *IEEE Internet of Things Journal*, 8(12), 9463-9472. doi: 10.1109/JIOT.2020.2996590.

Alladi, T., Chamola, V., Sikdar, B. & Choo, K.-K. R. (2020). Consumer IoT: Security Vulnerability Case Studies and Solutions. *IEEE Consumer Electronics Magazine*, 9(2), 17-25. doi: 10.1109/MCE.2019.2953740.

Alsamani, B. & Lahza, H. (2018). A taxonomy of IoT: Security and privacy threats. *2018 International Conference on Information and Computer Technologies (ICICT)*, pp. 72-77. doi: 10.1109/INFOCT.2018.8356843.

Alvee, S. R. B., Ahn, B., Kim, T., Su, Y., Youn, Y. & Ryu, M. (2021). Ransomware Attack Modeling and Artificial Intelligence-Based Ransomware Detection for Digital Substations. *2021 6th IEEE Workshop on the Electronic Grid (eGRID)*, pp. 01-05. doi: 10.1109/eGRID52793.2021.9662158.

Alwis, C. D., Kalla, A., Pham, Q.-V., Kumar, P., Dev, K., Hwang, W.-J. & Liyanage, M. (2021). Survey on 6G Frontiers: Trends, Applications, Requirements, Technologies and Future Research. *IEEE Open Journal of the Communications Society*, 2, 836-886. doi: 10.1109/OJCOMS.2021.3071496.

Amouri, A., Alaparthy, V. T. & Morgera, S. D. (2018). Cross layer-based intrusion detection based on network behavior for IoT. *2018 IEEE 19th Wireless and Microwave Technology Conference (WAMICON)*, pp. 1-4. doi: 10.1109/WAMICON.2018.8363921.

Andrea, I., Chrysostomou, C. & Hadjichristofi, G. (2015a). Internet of Things: Security vulnerabilities and challenges. *2015 IEEE Symposium on Computers and Communication (ISCC)*, pp. 180-187. doi: 10.1109/ISCC.2015.7405513.

Andrea, I., Chrysostomou, C. & Hadjichristofi, G. (2015b). Internet of Things: Security vulnerabilities and challenges. *2015 IEEE Symposium on Computers and Communication (ISCC)*, pp. 180–187.

Anthi, E., Williams, L., Słowińska, M., Theodorakopoulos, G. & Burnap, P. (2019). A Supervised Intrusion Detection System for Smart Home IoT Devices. *IEEE Internet of Things Journal*, 6(5), 9042–9053.

Aouedi, O., Piamrat, K., Muller, G. & Singh, K. (2022). FLUIDS: Federated Learning with semi-supervised approach for Intrusion Detection System. *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*, pp. 523-524. doi: 10.1109/CCNC49033.2022.9700632.

Aouedi, O., Piamrat, K., Muller, G. & Singh, K. (2023). Federated Semi-supervised Learning for Attack Detection in Industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, 19(1), 286-295. doi: 10.1109/TII.2022.3156642.

Apruzzese, G., Andreolini, M., Marchetti, M., Venturi, A. & Colajanni, M. (2020). Deep Reinforcement Adversarial Learning Against Botnet Evasion Attacks. *IEEE Transactions on Network and Service Management*, 17(4), 1975-1987. doi: 10.1109/TNSM.2020.3031843.

Arafin, M. T., Gao, M. & Qu, G. (2017). VOLtA: Voltage over-scaling based lightweight authentication for IoT applications. *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 336-341. doi: 10.1109/ASPDAC.2017.7858345.

Arisdakessian, S., Wahab, O. A., Mourad, A., Otrok, H. & Guizani, M. (2022). A Survey on IoT Intrusion Detection: Federated Learning, Game Theory, Social Psychology and Explainable AI as Future Directions. *IEEE Internet of Things Journal*, 1-1. doi: 10.1109/JIOT.2022.3203249.

Ashton, K. et al. (2009). That 'internet of things' thing. *RFID journal*, 22(7), 97–114.

Atzori, L., Iera, A., Morabito, G. & Nitti, M. (2012). The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer networks*, 56(16), 3594–3608.

Aujla, G. S., Chaudhary, R., Kumar, N., Rodrigues, J. J. P. C. & Vinel, A. (2017). Data Offloading in 5G-Enabled Software-Defined Vehicular Networks: A Stackelberg-Game-Based Approach. *IEEE Communications Magazine*, 55(8), 100-108. doi: 10.1109/MCOM.2017.1601224.

Baek, J. (2022). *Artificial intelligence-empowered resource management system for fog computing networks*. (Ph.D. Thesis, École de technologie supérieure).

Bajaj, K. & Arora, A. (2013). Improving the intrusion detection using discriminative machine learning approach and improve the time complexity by data mining feature selection methods. *International Journal of Computer Applications*, 76(1).

Ballerini, M., Polonelli, T., Brunelli, D., Magno, M. & Benini, L. (2020). NB-IoT Versus LoRaWAN: An Experimental Evaluation for Industrial Applications. *IEEE Transactions on Industrial Informatics*, 16(12), 7802-7811. doi: 10.1109/TII.2020.2987423.

Barker, S. & Parsons, D. (2022). Smart Homes or Real Homes: Building a Smarter Grid With "Dumb" Houses. *IEEE Pervasive Computing*, 21(2), 100-104. doi: 10.1109/MPRV.2022.3160752.

Barra, S., Carta, S. M., Corriga, A., Podda, A. S. & Recupero, D. R. (2020). Deep learning and time series-to-image encoding for financial forecasting. *IEEE/CAA Journal of Automatica Sinica*, 7(3), 683-692. doi: 10.1109/JAS.2020.1003132.

Beaver, J. M., Borges-Hink, R. C. & Buckner, M. A. (2013). An evaluation of machine learning methods to detect malicious SCADA communications. *2013 12th international conference on machine learning and applications*, 2, 54–59.

Bedhief, I., Kassar, M. & Aguili, T. (2016). SDN-based architecture challenging the IoT heterogeneity. *2016 3rd Smart Cloud Networks Systems (SCNS)*, pp. 1-3. doi: 10.1109/SCNS.2016.7870558.

Belenko, V., Chernenko, V., Kalinin, M. & Krundyshev, V. (2018). Evaluation of GAN applicability for intrusion detection in self-organizing networks of cyber physical systems. *2018 International Russian Automation Conference (RusAutoCon)*, pp. 1–7.

Bera, S., Misra, S. & Vasilakos, A. V. (2017). Software-Defined Networking for Internet of Things: A Survey. *IEEE Internet of Things Journal*, 4(6), 1994-2008. doi: 10.1109/JIOT.2017.2746186.

Betser, J., Walther, A., Erlinger, M., Buchheim, T., Feinstein, B., Matthews, G., Pollock, R. & Levitt, K. (2001). GlobalGuard: creating the IETF-IDWG Intrusion Alert Protocol (IAP). *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01*, 1, 22-34 vol.1. doi: 10.1109/DISCEX.2001.932189.

Beyene, Y. D., Jantti, R., Tirkkonen, O., Ruttik, K., Iraji, S., Larmo, A., Tirronen, T., Torsner & Johan. (2017). NB-IoT Technology Overview and Experience from Cloud-RAN Implementation. *IEEE Wireless Communications*, 24(3), 26-32. doi: 10.1109/MWC.2017.1600418.

Bhuiyan, M. N., Rahman, M. M., Billah, M. M. & Saha, D. (2021). Internet of Things (IoT): A Review of Its Enabling Technologies in Healthcare Applications, Standards Protocols, Security, and Market Opportunities. *IEEE Internet of Things Journal*, 8(13), 10474-10498. doi: 10.1109/JIOT.2021.3062630.

Bhuyan, M. H., Bhattacharyya, D. K. & Kalita, J. K. (2014). Network Anomaly Detection: Methods, Systems and Tools. *IEEE Communications Surveys Tutorials*, 16(1), 303-336. doi: 10.1109/SURV.2013.052213.00046.

Bisson, P. & Waryet, J. (2017). 5G PPP phase1 security landscape. *5G PPP Security Group White Paper*.

Bonomi, F., Milito, R., Zhu, J. & Addepalli, S. (2012). Fog computing and its role in the internet of things. *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16.

Botta, A., De Donato, W., Persico, V. & Pescapé, A. (2016). Integration of cloud computing and internet of things: a survey. *Future generation computer systems*, 56, 684–700.

Brandao, J. E., Silva Fraga, J. d., Mafra, P. M. & Obelheiro, R. R. (2006). A WS-based infrastructure for integrating intrusion detection systems in large-scale environments. *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pp. 462–479.

Cao, K., Ding, H., Wang, B., Lv, L., Tian, J., Wei, Q. & Gong, F. (2022). Enhancing Physical Layer Security for IoT with Non-Orthogonal Multiple Access Assisted Semi-Grant-Free Transmission. *IEEE Internet of Things Journal*, 1-1. doi: 10.1109/JIOT.2022.3193189.

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C. & Faruki, P. (2019). Network Intrusion Detection for IoT Security based on Learning Techniques. *IEEE Communications Surveys & Tutorials*.

Chandola, V., Banerjee, A. & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3), 1–58.

Chaudhary, C. (2018, August, 30). Massive and critical IoT: Helping developers choose the right connectivity [html]. Consulted at https://internetofthingsagenda.techtarget.com/blog/IoT-Agenda/Massive-and-critical-IoT-Helping-developers-choose-the-right-connectivity.

Chaudhary, R., Aujla, G. S., Kumar, N. & Zeadally, S. (2019). Lattice-Based Public Key Cryptosystem for Internet of Things Environment: Challenges and Solutions. *IEEE Internet of Things Journal*, 6(3), 4897-4909. doi: 10.1109/JIOT.2018.2878707.

Chen, K., Zhang, S., Li, Z., Zhang, Y., Deng, Q., Ray, S. & Jin, Y. (2018). Internet-of-Things security and vulnerabilities: Taxonomy, challenges, and practice. *Journal of Hardware and Systems Security*, 2(2), 97–110.

Cheng, Y., Xu, Y., Zhong, H. & Liu, Y. (2019). HS-TCN: A Semi-supervised Hierarchical Stacking Temporal Convolutional Network for Anomaly Detection in IoT. *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*, pp. 1-7. doi: 10.1109/IPCCC47392.2019.8958755.

Chettri, L. & Bera, R. (2020). A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems. *IEEE Internet of Things Journal*, 7(1), 16-32. doi: 10.1109/JIOT.2019.2948888.

Chhikara, P., Singh, P., Tekchandani, R., Kumar, N. & Guizani, M. (2021). Federated Learning Meets Human Emotions: A Decentralized Framework for Human–Computer Interaction for IoT Applications. *IEEE Internet of Things Journal*, 8(8), 6949-6962. doi: 10.1109/JIOT.2020.3037207.

Chin, T., Mountrouidou, X., Li, X. & Xiong, K. (2015). An SDN-supported collaborative approach for DDoS flooding detection and containment. *MILCOM 2015 - 2015 IEEE Military Communications Conference*, pp. 659-664. doi: 10.1109/MILCOM.2015.7357519.

Choudhary, G., Astillo, P. V., You, I., Yim, K., Chen, I.-R. & Cho, J.-H. (2020). Lightweight Misbehavior Detection Management of Embedded IoT Devices in Medical Cyber Physical Systems. *IEEE Transactions on Network and Service Management*, 17(4), 2496-2510. doi: 10.1109/TNSM.2020.3007535.

Cuff, D., Hansen, M. & Kang, J. (2008). Urban sensing: out of the woods. *Communications of the ACM*, 51(3), 24.

Deebak, B. D., Memon, F. H., Khowaja, S. A., Dev, K., Wang, W., Qureshi, N. M. F. & Su, C. (2022). Lightweight Blockchain Based Remote Mutual Authentication for AI-Empowered IoT Sustainable Computing Systems. *IEEE Internet of Things Journal*, 1-1. doi: 10.1109/JIOT.2022.3152546.

Deogirikar, J. & Vidhate, A. (2017). Security attacks in IoT: A survey. *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 32-37. doi: 10.1109/I-SMAC.2017.8058363.

Ding, Z. (2021). Harvesting Devices' Heterogeneous Energy Profiles and QoS Requirements in IoT: WPT-NOMA vs BAC-NOMA. *IEEE Transactions on Communications*, 69(5), 2837-2850. doi: 10.1109/TCOMM.2021.3052948.

Ding, Z., Liu, Y., Choi, J., Sun, Q., Elkashlan, M., Chih-Lin, I. & Poor, H. V. (2017). Application of Non-Orthogonal Multiple Access in LTE and 5G Networks. *IEEE Communications Magazine*, 55(2), 185-191. doi: 10.1109/MCOM.2017.1500657CM.

Dua, S. & Du, X. (2016). *Data mining and machine learning in cybersecurity*. CRC press.

Dutta, A. & Hammad, E. (2020). 5G Security Challenges and Opportunities: A System Approach. *2020 IEEE 3rd 5G World Forum (5GWF)*, pp. 109-114. doi: 10.1109/5GWF49715.2020.9221122.

Dutta, I. K., Ghosh, B. & Bayoumi, M. (2019). Lightweight Cryptography for Internet of Insecure Things: A Survey. *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0475–0481.

Eclipse IoT Working Group. (2016). The three software stacks required for IoT architectures. *IoT software requirements and how to implement then using open source technology*.

Ejaz, W., Anpalagan, A., Imran, M. A., Jo, M., Naeem, M., Qaisar, S. B. & Wang, W. (2016). Internet of Things (IoT) in 5G Wireless Communications. *IEEE Access*, 4, 10310-10314. doi: 10.1109/ACCESS.2016.2646120.

Elmiligi, H., Gebali, F. & El-Kharashi, M. W. (2016). Multi-dimensional analysis of embedded systems security. *Microprocessors and Microsystems*, 41, 29–36.

EOP. (2012). A National Strategic Plan for Advanced Manufacturing. National Science and Technology Council.

Eskandari, M., Janjua, Z. H., Vecchio, M. & Antonelli, F. (2020). Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices. *IEEE Internet of Things Journal*, 7(8), 6882-6897. doi: 10.1109/JIOT.2020.2970501.

European Commission. (2010). EUROPE 2020: A strategy for smart, sustainable and inclusive growth. *Working paper {COM (2010) 2020}*.

Facchini, S., Giorgi, G., Saracino, A. & Dini, G. (2020). Multi-level Distributed Intrusion Detection System for an IoT based Smart Home Environment. *ICISSP*, pp. 705–712.

Fan, X., Fan, K., Wang, Y. & Zhou, R. (2015). Overview of cyber-security of industrial control system. *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*, pp. 1-7. doi: 10.1109/SSIC.2015.7245324.

Farris, I., Taleb, T., Khettab, Y. & Song, J. (2019). A Survey on Emerging SDN and NFV Security Mechanisms for IoT Systems. *IEEE Communications Surveys Tutorials*, 21(1), 812-837. doi: 10.1109/COMST.2018.2862350.

Ferdowsi, A. & Saad, W. (2019). Generative Adversarial Networks for Distributed Intrusion Detection in the Internet of Things. *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-6.

Flauzac, O., González, C., Hachani, A. & Nolot, F. (2015). SDN Based Architecture for IoT and Improvement of the Security. *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, pp. 688-693. doi: 10.1109/WAINA.2015.110.

Forbes. (2019). Roundup Of Internet Of Things Forecasts And Market Estimates. Consulted at http://tinyurl.com/yar5llet.

Foundation Biotope. (2017, August, 4). The bIoTope Project [html]. Consulted at https://biotope-project.eu/.

Furfaro, A., Argento, L., Parise, A. & Piccolo, A. (2017). Using virtual environments for the assessment of cybersecurity issues in IoT scenarios. *Simulation Modelling Practice and Theory*, 73, 43–54.

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G. & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & security*, 28(1-2), 18–28.

Gershenfeld, N. A. (1999). *When things start to think*. Macmillan.

Gomes, M. M., Righi, R. d. R. & da Costa, C. A. (2014). Future directions for providing better IoT infrastructure. *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct Publication*, pp. 51–54.

Gomes, T., Salgado, F., Pinto, S., Cabral, J. & Tavares, A. (2018). A 6LoWPAN Accelerator for Internet of Things Endpoint Devices. *IEEE Internet of Things Journal*, 5(1), 371-377. doi: 10.1109/JIOT.2017.2785659.

Gope, P. & Sikdar, B. (2019). Lightweight and Privacy-Preserving Two-Factor Authentication Scheme for IoT Devices. *IEEE Internet of Things Journal*, 6(1), 580-589. doi: 10.1109/JIOT.2018.2846299.

Grand, J. (2004). Practical secure hardware design for embedded systems. *Proceedings of the 2004 Embedded Systems Conference, San Francisco, California.*

Guembe, B., Azeta, A., Misra, S., Osamor, V. C., Fernandez-Sanz, L. & Pospelova, V. (2022). The emerging threat of ai-driven cyber attacks: A Review. *Applied Artificial Intelligence*, 36(1), 2037254.

Gunathilake, N. A., Al-Dubai, A. & Buchana, W. J. (2020). Recent Advances and Trends in Lightweight Cryptography for IoT Security. *2020 16th International Conference on Network and Service Management (CNSM)*, pp. 1-5. doi: 10.23919/CNSM50824.2020.9269083.

Gupta, R., Tanwar, S., Tyagi, S. & Kumar, N. (2020). Machine learning models for secure data analytics: A taxonomy and threat model. *Computer Communications*, 153, 406–440.

Hammar, K. & Stadler, R. (2020). Finding Effective Security Strategies through Reinforcement Learning and Self-Play. *2020 16th International Conference on Network and Service Management (CNSM)*, pp. 1-9. doi: 10.23919/CNSM50824.2020.9269092.

Hammar, K. & Stadler, R. (2022). Intrusion Prevention Through Optimal Stopping. *IEEE Transactions on Network and Service Management*, 19(3), 2333-2348. doi: 10.1109/TNSM.2022.3176781.

Hammi, M. T., Livolant, E., Bellot, P., Serhrouchni, A. & Minet, P. (2017). A lightweight IoT security protocol. *2017 1st Cyber Security in Networking Conference (CSNet)*, pp. 1-8. doi: 10.1109/CSNET.2017.8242001.

Han, G., Xiao, L. & Poor, H. V. (2017). Two-dimensional anti-jamming communication based on deep reinforcement learning. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2087–2091.

Han, L., Zhou, M., Jia, W., Dalil, Z. & Xu, X. (2019). Intrusion detection model of wireless sensor networks based on game theory and an autoregressive model. *Information Sciences*, 476, 491–504.

Haroon, A., Shah, M. A., Asim, Y., Naeem, W., Kamran, M. & Javaid, Q. (2016). Constraints in the IoT: the world in 2020 and beyond. *International Journal of Advanced Computer Science and Applications*, 7(11).

Hashima, S., Fadlullah, Z. M., Fouda, M. M., Mohamed, E. M., Hatano, K., ElHalawany, B. M. & Guizani, M. (2022). On Softwarization of Intelligence in 6G Networks for Ultra-Fast Optimal Policy Selection: Challenges and Opportunities. *IEEE Network*, 1-9. doi: 10.1109/MNET.103.2100587.

Heartfield, R., Loukas, G., Bezemskij, A. & Panaousis, E. (2021). Self-Configurable Cyber-Physical Intrusion Detection for Smart Homes Using Reinforcement Learning. *IEEE Transactions on Information Forensics and Security*, 16, 1720-1735. doi: 10.1109/TIFS.2020.3042049.

Heer, T., Garcia-Morchon, O., Hummen, R., Keoh, S. L., Kumar, S. S. & Wehrle, K. (2011). Security Challenges in the IP-based Internet of Things. *Wireless Personal Communications*, 61(3), 527–542.

Hermosilla, A., Zarca, A. M., Bernabe, J. B., Ortiz, J. & Skarmeta, A. (2020). Security Orchestration and Enforcement in NFV/SDN-Aware UAV Deployments. *IEEE Access*, 8, 131779-131795. doi: 10.1109/ACCESS.2020.3010209.

Hernandez, G., Arias, O., Buentello, D. & Jin, Y. (2014). Smart nest thermostat: A smart spy in your home. *Black Hat USA*, (2015).

Hewa, T., Gür, G., Kalla, A., Ylianttila, M., Bracken, A. & Liyanage, M. (2020). The Role of Blockchain in 6G: Challenges, Opportunities and Research Directions. *2020 2nd 6G Wireless Summit (6G SUMMIT)*, pp. 1-5. doi: 10.1109/6GSUMMIT49458.2020.9083784.

Hewlett Packard. (2015). HP Internet of things research study. *Internet of Things Research Study*.

Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C. & Atkinson, R. (2017). Shallow and deep networks intrusion detection system: A taxonomy and survey. *arXiv preprint arXiv:1701.02145*.

Hong, W., Jiang, Z. H., Yu, C., Zhou, J., Chen, P., Yu, Z., Zhang, H., Yang, B., Pang, X., Jiang, M. et al. (2017). Multibeam antenna technologies for 5G wireless communications. *IEEE Transactions on Antennas and Propagation*, 65(12), 6231–6249.

Houda, Z. A. E., Brik, B. & Khoukhi, L. (2022). "Why Should I Trust Your IDS?": An Explainable Deep Learning Framework for Intrusion Detection Systems in Internet of Things Networks. *IEEE Open Journal of the Communications Society*, 3, 1164-1176. doi: 10.1109/OJCOMS.2022.3188750.

Howz. (2019, January, 3). Howz: a Smart Home for the elderly [html]. Consulted at https://www.edf.fr/en/howz-smart-home-senior-citizens.

Huang, Z., Yang, J. & Gong, C. (2022). They are Not Completely Useless: Towards Recycling Transferable Unlabeled Data for Class-Mismatched Semi-Supervised Learning. *IEEE Transactions on Multimedia*, 1-1. doi: 10.1109/TMM.2022.3179895.

Husain, A., Salem, A., Jim, C. & Dimitoglou, G. (2019). Development of an Efficient Network Intrusion Detection Model Using Extreme Gradient Boosting (XGBoost) on the UNSW-NB15 Dataset. *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 1-7. doi: 10.1109/ISSPIT47144.2019.9001867.

Ibrahim, L. M., Basheer, D. T. & Mahmod, M. S. (2013). A comparison study for intrusion database (KDD99, NSL-KDD) based on self organization map (SOM) artificial neural network. *Journal of Engineering Science and Technology*, 8(1), 107–119.

Iglesias, F. & Zseby, T. (2015). Analysis of network traffic features for anomaly detection. *Machine Learning*, 101(1), 59–84.

Illy, P., Kaddoum, G., Miranda Moreira, C., Kaur, K. & Garg, S. (2019). Securing Fog-to-Things Environment Using Intrusion Detection System Based On Ensemble Learning. *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1-7.

Illy, P., Kaddoum, G., de Araujo-Filho, P. F., Kaur, K. & Garg, S. (2022a). A Hybrid Multistage DNN-Based Collaborative IDPS for High-Risk Smart Factory Networks. *IEEE Transactions on Network and Service Management*, 19(4), 4273-4283. doi: 10.1109/TNSM.2022.3202801.

Illy, P., Kaddoum, G., Kaur, K. & Garg, S. (2022b). ML-based IDPS Enhancement With Complementary Features For Home IoT networks. *IEEE Transactions on Network and Service Management*, 1-1. doi: 10.1109/TNSM.2022.3141942.

Ingre, B. & Yadav, A. (2015). Performance analysis of NSL-KDD dataset using ANN. *Signal Processing And Communication Engineering Systems (SPACES) - IEEE*, pp. 92–96.

Iqbal, A. & Lee, T.-J. (2019). GWINs: Group-Based Medium Access for Large-Scale Wireless Powered IoT Networks. *IEEE Access*, 7, 172913-172927. doi: 10.1109/ACCESS.2019.2956029.

Iqbal, M., Abdullah, A. Y. M. & Shabnam, F. (2020). An Application Based Comparative Study of LPWAN Technologies for IoT Environment. *2020 IEEE Region 10 Symposium (TENSYMP)*, pp. 1857-1860. doi: 10.1109/TENSYMP50017.2020.9230597.

Jain, R. & Paul, S. (2013). Network virtualization and software defined networking for cloud computing: a survey. *IEEE Communications Magazine*, 51(11), 24-31. doi: 10.1109/MCOM.2013.6658648.

Janarthanan, T. & Zargari, S. (2017). Feature selection in UNSW-NB15 and KDDCUP'99 datasets. *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pp. 1881-1886. doi: 10.1109/ISIE.2017.8001537.

Javed, A. R., Rehman, S. u., Khan, M. U., Alazab, M. & G, T. R. (2021). CANintelliIDS: Detecting In-Vehicle Intrusion Attacks on a Controller Area Network Using CNN and Attention-Based GRU. *IEEE Transactions on Network Science and Engineering*, 8(2), 1456-1466. doi: 10.1109/TNSE.2021.3059881.

Jiang, X., Zhang, H., Barsallo Yi, E. A., Raghunathan, N., Mousoulis, C., Chaterji, S., Peroulis, D., Shakouri, A. & Bagchi, S. (2021). Hybrid Low-Power Wide-Area Mesh Network for IoT Applications. *IEEE Internet of Things Journal*, 8(2), 901-915. doi: 10.1109/JIOT.2020.3009228.

Jing, D. & Chen, H.-B. (2019). SVM Based Network Intrusion Detection for the UNSW-NB15 Dataset. *2019 IEEE 13th International Conference on ASIC (ASICON)*, pp. 1-4. doi: 10.1109/ASICON47005.2019.8983598.

Jokar, P. & Leung, V. C. M. (2018). Intrusion Detection and Prevention for ZigBee-Based Home Area Networks in Smart Grids. *IEEE Transactions on Smart Grid*, 9(3), 1800-1811. doi: 10.1109/TSG.2016.2600585.

Jokar, P. & Leung, V. C. (2016). Intrusion detection and prevention for zigbee-based home area networks in smart grids. *IEEE Transactions on Smart Grid*, 9(3), 1800–1811.

Kaloudi, N. & Li, J. (2020). The ai-based cyber threat landscape: A survey. *ACM Computing Surveys (CSUR)*, 53(1), 1–34.

Kaur, K., Dhand, T., Kumar, N. & Zeadally, S. (2017). Container-as-a-Service at the Edge: Trade-off between Energy Efficiency and Service Availability at Fog Nano Data Centers. *IEEE Wireless Communications*, 24(3), 48-56. doi: 10.1109/MWC.2017.1600427.

KDDCUP. (1999). KDDCUP99 [Online] Available:. *http://kdd.ics.uci.edu/databases/ kddcup99/kddcup99.html*.

Kemmerer, R. A. & Vigna, G. (2002). Intrusion detection: a brief history and overview. *Computer*, 35(4), 27–30.

Khan, M. A., Abbas, S., Rehman, A., Saeed, Y., Zeb, A., Uddin, M. I., Nasser, N. & Ali, A. (2020). A Machine Learning Approach for Blockchain-Based Smart Home Networks Security. *IEEE Network*, 1-7. doi: 10.1109/MNET.011.2000514.

Kim, C.-M., Lim, S.-K., Jeong, J.-D., Choi, Y. & Koh, S.-J. (2022). 6LoWPAN Over Optical Wireless Communications for IPv6 Transport in Internet of Things Networks. *IEEE Wireless Communications Letters*, 11(6), 1142-1145. doi: 10.1109/LWC.2022.3159257.

Kim, K. & Aminanto, M. E. (2017). Deep learning in intrusion detection perspective: Overview and further challenges. *2017 International Workshop on Big Data and Information Security (IWBIS)*, pp. 5-10. doi: 10.1109/IWBIS.2017.8275095.

Krejčí, R., Hujňák, O. & Švepeš, M. (2017). Security survey of the IoT wireless protocols. *2017 25th Telecommunication Forum (TELFOR)*, pp. 1-4. doi: 10.1109/TELFOR.2017.8249286.

Krupka, L., Vojtech, L. & Neruda, M. (2016). The issue of LPWAN technology coexistence in IoT environment. *2016 17th International Conference on Mechatronics - Mechatronika (ME)*, pp. 1-8.

Lam, K.-Y., Mitra, S., Gondesen, F. & Yi, X. (2022). ANT-Centric IoT Security Reference Architecture—Security-by-Design for Satellite-Enabled Smart Cities. *IEEE Internet of Things Journal*, 9(8), 5895-5908. doi: 10.1109/JIOT.2021.3073734.

Lasi, H., Fettke, P., Kemper, H., Feld, T. & Hoffmann, M. (2014). Industry 4.0. Bus Inf Syst Eng 6 (4): 239–242.

Le, A., Dinh, P., Le, H. & Tran, N. C. (2015). Flexible Network-Based Intrusion Detection and Prevention System on Software-Defined Networks. *2015 International Conference on Advanced Computing and Applications (ACOMP)*, pp. 106-111. doi: 10.1109/A-COMP.2015.19.

Leo, M., Battisti, F., Carli, M. & Neri, A. (2014). A federated architecture approach for Internet of Things security. *2014 Euro Med Telco Conference (EMTC)*, pp. 1–5.

Li, B., Wu, Y., Song, J., Lu, R., Li, T. & Zhao, L. (2021). DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber–Physical Systems. *IEEE Transactions on Industrial Informatics*, 17(8), 5615-5624. doi: 10.1109/TII.2020.3023430.

Li, H., Dong, M. & Ota, K. (2016). Control Plane Optimization in Software-Defined Vehicular Ad Hoc Networks. *IEEE Transactions on Vehicular Technology*, 65(10), 7895-7904. doi: 10.1109/TVT.2016.2563164.

Li, W. & Meng, W. (2022). BCTrustFrame: Enhancing Trust Management via Blockchain and IPFS in 6G Era. *IEEE Network*, 36(4), 120-125. doi: 10.1109/MNET.013.2100768.

Li, W., Tug, S., Meng, W. & Wang, Y. (2019). Designing collaborative blockchained signature-based intrusion detection in IoT environments. *Future Generation Computer Systems*, 96, 481–489.

Li, Y. & Chen, M. (2015). Software-Defined Network Function Virtualization: A Survey. *IEEE Access*, 3, 2542-2553.

Liang, J., Ma, M. & Tan, X. (2022). GaDQN-IDS: A Novel Self-Adaptive IDS for VANETs Based on Bayesian Game Theory and Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(8), 12724-12737. doi: 10.1109/TITS.2021.3117028.

Lin, C.-T., Wu, S.-L. & Lee, M.-L. (2017). Cyber attack and defense on industry control systems. *2017 IEEE Conference on Dependable and Secure Computing*, pp. 524–526.

Ling, J., Zhu, Z., Luo, Y. & Wang, H. (2021). An intrusion detection method for industrial control systems based on bidirectional simple recurrent unit. *Computers & Electrical Engineering*, 91, 107049.

Liu, H., Zhang, S., Zhang, P., Zhou, X., Shao, X., Pu, G. & Zhang, Y. (2021). Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing. *IEEE Transactions on Vehicular Technology*, 70(6), 6073-6084. doi: 10.1109/TVT.2021.3076780.

Liyanage, M., Abro, A. B., Ylianttila, M. & Gurtov, A. (2016). Opportunities and challenges of software-defined mobile networks in network security. *IEEE Security & Privacy*, 14(4), 34–44.

Mahmood, K., Chilwan, A., Østerbø, O. & Jarschel, M. (2015). Modelling of OpenFlow-based software-defined networks: the multiple node case. *IET Networks*, 4(5), 278–284.

Mahmoud, R., Yousuf, T., Aloul, F. & Zualkernan, I. (2015a). Internet of things (IoT) security: Current status, challenges and prospective measures. *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 336-341. doi: 10.1109/ICITST.2015.7412116.

Mahmoud, R., Yousuf, T., Aloul, F. & Zualkernan, I. (2015b). Internet of things (IoT) security: Current status, challenges and prospective measures. *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 336–341.

Maiti, M. & Ghosh, U. (2021). Next Generation Internet of Things in Fintech Ecosystem. *IEEE Internet of Things Journal*, 1-1. doi: 10.1109/JIOT.2021.3063494.

Mansour, R. F. (2022). Artificial intelligence based optimization with deep learning model for blockchain enabled intrusion detection in CPS environment. *Scientific Reports*, 12(1), 12937.

Markowsky, L. & Markowsky, G. (2015). Scanning for vulnerable devices in the Internet of Things. *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 1, 463-467. doi: 10.1109/IDAACS.2015.7340779.

Matin, I. M. M. & Rahardjo, B. (2020). The Use of Honeypot in Machine Learning Based on Malware Detection: A Review. *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1-6. doi: 10.1109/CITSM50537.2020.9268794.

Maxli Campos & Martins, J. S. B. (2017). A Sdn-Based Flexible System For On-The-Fly Monitoring And Treatment Of Security Events. doi: 10.5281/ZENODO.1291094.

Mekki, K., Bajic, E., Chaxel, F. & Meyer, F. (2018). Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT. *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 197-202. doi: 10.1109/PERCOMW.2018.8480255.

Mekki, K., Bajic, E., Chaxel, F. & Meyer, F. (2019). A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT express*, 5(1), 1–7.

Mell, P., Grance, T. et al. (2011). The NIST definition of cloud computing.

Meng, W., Tischhauser, E. W., Wang, Q., Wang, Y. & Han, J. (2018). When Intrusion Detection Meets Blockchain Technology: A Review. *IEEE Access*, 6, 10179-10188. doi: 10.1109/ACCESS.2018.2799854.

Miglani, A. & Kumar, N. (2019). Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges. *Vehicular Communications*, 20, 100184.

Mirkovic, J. & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2), 39–53.

Mishra, P., Pilli, E. S., Varadharajan, V. & Tupakula, U. (2017). Out-VM monitoring for Malicious Network Packet Detection in cloud. *2017 ISEA Asia Security and Privacy (ISEASP)*, pp. 1-10. doi: 10.1109/ISEASP.2017.7976995.

Misra, N. N., Dixit, Y., Al-Mallahi, A., Bhullar, M. S., Upadhyay, R. & Martynenko, A. (2022). IoT, Big Data, and Artificial Intelligence in Agriculture and Food Industry. *IEEE Internet of Things Journal*, 9(9), 6305-6324. doi: 10.1109/JIOT.2020.2998584.

Mohammadi, M., Al-Fuqaha, A., Sorour, S. & Guizani, M. (2018). Deep Learning for IoT Big Data and Streaming Analytics: A Survey. *IEEE Communications Surveys Tutorials*, 20(4), 2923-2960. doi: 10.1109/COMST.2018.2844341.

Monrat, A. A., Schelén, O. & Andersson, K. (2019). A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities. *IEEE Access*, 7, 117134-117151. doi: 10.1109/ACCESS.2019.2936094.

Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R. H., Morrow, M. J. & Polakos, P. A. (2018). A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges. *IEEE Communications Surveys Tutorials*, 20(1), 416-464. doi: 10.1109/COMST.2017.2771153.

Moustafa, N. & Slay, J. (2015a). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1-6. doi: 10.1109/MilCIS.2015.7348942.

Moustafa, N. & Slay, J. (2015b). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 military communications and information systems conference (MilCIS)*, pp. 1–6.

Moustafa, N. & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective*, 25(1-3), 18–31.

Mpitziopoulos, A., Gavalas, D., Konstantopoulos, C. & Pantziou, G. (2009). A survey on jamming attacks and countermeasures in WSNs. *IEEE Communications Surveys & Tutorials*, 11(4), 42–56.

Mughal, M. A., Luo, X., Ullah, A., Ullah, S. & Mahmood, Z. (2018). A Lightweight Digital Signature Based Security Scheme for Human-Centered Internet of Things. *IEEE Access*, 6, 31630-31643. doi: 10.1109/ACCESS.2018.2844406.

Mukherjee, A., Goswami, P., Yang, L., Sah Tyagi, S. K., Samal, U. C. & Mohapatra, S. (2020). Deep neural network-based clustering technique for secure IIoT. *Neural Computing and Applications*, 32(20), 16109–16117.

Mylrea, M., Gourisetti, S. N. G. & Nicholls, A. (2017). An introduction to buildings cybersecurity framework. *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1-7. doi: 10.1109/SSCI.2017.8285228.

Najafabadi, M. M., Khoshgoftaar, T. M., Kemp, C., Seliya, N. & Zuech, R. (2014). Machine Learning for Detecting Brute Force Attacks at the Network Level. *2014 IEEE International Conference on Bioinformatics and Bioengineering*, pp. 379-385. doi: 10.1109/BIBE.2014.73.

Nassif, A. B., Shahin, I., Attili, I., Azzeh, M. & Shaalan, K. (2019). Speech Recognition Using Deep Neural Networks: A Systematic Review. *IEEE Access*, 7, 19143-19165. doi: 10.1109/ACCESS.2019.2896880.

Navda, V., Bohra, A., Ganguly, S. & Rubenstein, D. (2007). Using Channel Hopping to Increase 802.11 Resilience to Jamming Attacks. *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pp. 2526-2530.

Nawir, M., Amir, A., Yaakob, N. & Lynn, O. B. (2016). Internet of Things (IoT): Taxonomy of security attacks. *2016 3rd International Conference on Electronic Design (ICED)*, pp. 321–326.

Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G. & Ghani, N. (2019). Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. *IEEE Communications Surveys Tutorials*, 21(3), 2702-2733. doi: 10.1109/COMST.2019.2910750.

Nguyen, T., Tran, N., Loven, L., Partala, J., Kechadi, M.-T. & Pirttikangas, S. (2020). Privacy-Aware Blockchain Innovation for 6G: Challenges and Opportunities. *2020 2nd 6G Wireless Summit (6G SUMMIT)*, pp. 1-5. doi: 10.1109/6GSUMMIT49458.2020.9083832.

Nie, L., Wu, Y., Wang, X., Guo, L., Wang, G., Gao, X. & Li, S. (2022). Intrusion Detection for Secure Social Internet of Things Based on Collaborative Edge Computing: A Generative Adversarial Network-Based Approach. *IEEE Transactions on Computational Social Systems*, 9(1), 134-145. doi: 10.1109/TCSS.2021.3063538.

Nobakht, M., Sivaraman, V. & Boreli, R. (2016). A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow. *2016 11th International conference on availability, reliability and security (ARES)*, pp. 147–156.

Nokia. (2014). Networks,"5G use cases and requirements,". *White Paper*.

Norris, D. F., Mateczun, L. K. & Forno, R. F. (2022). The NIST Cybersecurity Framework Demystified. In *Cybersecurity and Local Government* (pp. 151-165). doi: 10.1002/9781119788317.ch9.

Oh, S.-R. & Kim, Y.-G. (2017). Security Requirements Analysis for the IoT. *2017 International Conference on Platform Technology and Service (PlatCon)*, pp. 1-6. doi: 10.1109/PlatCon.2017.7883727.

Osman, N. I. & Abbas, E. B. (2018). Simulation and Modelling of LoRa and Sigfox Low Power Wide Area Network Technologies. *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pp. 1-5. doi: 10.1109/ICCCEEE.2018.8515816.

Otoum, S., Kantarci, B. & Mouftah, H. (2019). Empowering Reinforcement Learning on Big Sensed Data for Intrusion Detection. *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1-7. doi: 10.1109/ICC.2019.8761575.

OWASP IoT Security Team, 2018. (2019). OWASP IoT Top 10 2018. Consulted at https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf.

Pajouh, H. H., Javidan, R., Khayami, R., Dehghantanha, A. & Choo, K.-K. R. (2019). A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks. *IEEE Transactions on Emerging Topics in Computing*, 7(2), 314-323. doi: 10.1109/TETC.2016.2633228.

Palattella, M. R., Dohler, M., Grieco, A., Rizzo, G., Torsner, J., Engel, T. & Ladid, L. (2016). Internet of things in the 5G era: Enablers, architecture, and business models. *IEEE Journal on Selected Areas in Communications*, 34(3), 510–527.

Palma, D. (2018). Enabling the Maritime Internet of Things: CoAP and 6LoWPAN Performance Over VHF Links. *IEEE Internet of Things Journal*, 5(6), 5205-5212. doi: 10.1109/JIOT.2018.2868439.

Park, C., Lee, J., Kim, Y., Park, J.-G., Kim, H. & Hong, D. (2022). An enhanced AI-based Network Intrusion Detection System using Generative Adversarial Networks. *IEEE Internet of Things Journal*, 1-1. doi: 10.1109/JIOT.2022.3211346.

Pelechrinis, K., Broustis, I., Krishnamurthy, S. V. & Gkantsidis, C. (2011). A Measurement-Driven Anti-Jamming System for 802.11 Networks. *IEEE/ACM Transactions on Networking*, 19(4), 1208-1222.

Perera, C., Liu, C. H. & Jayawardena, S. (2015). The emerging internet of things marketplace from an industrial perspective: A survey. *IEEE Transactions on Emerging Topics in Computing*, 3(4), 585–598.

Ploennigs, J., Cohn, J. & Stanford-Clark, A. (2018). The Future of IoT. *IEEE Internet of Things Magazine*, 1(1), 28-33. doi: 10.1109/IOTM.2018.1700021.

Rahouti, M., Xiong, K., Xin, Y., Jagatheesaperumal, S. K., Ayyash, M. & Shaheed, M. (2022). SDN Security Review: Threat Taxonomy, Implications, and Open Challenges. *IEEE Access*, 10, 45820-45854. doi: 10.1109/ACCESS.2022.3168972.

Ramsey, B. W., Mullins, B. E., Temple, M. A. & Grimaila, M. R. (2015). Wireless Intrusion Detection and Device Fingerprinting through Preamble Manipulation. *IEEE Transactions on Dependable and Secure Computing*, 12(5), 585-596. doi: 10.1109/TDSC.2014.2366455.

Rawat, D. B. & Reddy, S. R. (2017). Software Defined Networking Architecture, Security and Energy Efficiency: A Survey. *IEEE Communications Surveys Tutorials*, 19(1), 325-346. doi: 10.1109/COMST.2016.2618874.

REMAP. (2022, October, 02). Smart Manufacturing for a Connected World [html]. Consulted at https://www.remapnetwork.org/smart-manufacturing-events.

Restuccia, F., D'Oro, S. & Melodia, T. (2018). Securing the internet of things in the age of machine learning and software-defined networking. *IEEE Internet of Things Journal*, 5(6), 4829–4842.

Reynders, B., Meert, W. & Pollin, S. (2016). Range and coexistence analysis of long range unlicensed communication. *2016 23rd International Conference on Telecommunications (ICT)*, pp. 1-6. doi: 10.1109/ICT.2016.7500415.

Robla-Gómez, S., Becerra, V. M., Llata, J. R., González-Sarabia, E., Torre-Ferrero, C. & Pérez-Oria, J. (2017). Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments. *IEEE Access*, 5, 26754-26773. doi: 10.1109/ACCESS.2017.2773127.

Rubio-Aparicio, J., Cerdan-Cartagena, F., Suardiaz-Muro, J. & Ybarra-Moreno, J. (2019). Design and implementation of a mixed IoT LPWAN network architecture. *Sensors*, 19(3), 675.

Saad, W., Bennis, M. & Chen, M. (2020). A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems. *IEEE Network*, 34(3), 134-142. doi: 10.1109/MNET.001.1900287.

Sabahi, F. & Movaghar, A. (2008). Intrusion Detection: A Survey. *2008 Third International Conference on Systems and Networks Communications*, pp. 23-26. doi: 10.1109/ICSNC.2008.44.

Sadhu, P. K., Yanambaka, V. P. & Abdelgawad, A. (2022). Physical Unclonable Function and Machine Learning Based Group Authentication and Data Masking for In-Hospital Segments. *Electronics*, 11(24), 4155.

Salamatian, S., Huleihel, W., Beirami, A., Cohen, A. & Médard, M. (2019). Why Botnets Work: Distributed Brute-Force Attacks Need No Synchronization. *IEEE Transactions on Information Forensics and Security*, 14(9), 2288-2299. doi: 10.1109/TIFS.2019.2895955.

Salih, K. O. M., Rashid, T. A., Radovanovic, D. & Bacanin, N. (2022). A comprehensive survey on the Internet of Things with the industrial marketplace. *Sensors*, 22(3), 730.

Salman, T. (2015). Internet of things protocols and standards. *M. Of END, Affairs*.

Salnitri, M., Alizadeh, M., Giovanella, D., Zannone, N. & Giorgini, P. (2018). From security-by-design to the identification of security-critical deviations in process executions. *International Conference on Advanced Information Systems Engineering*, pp. 218–234.

Samaniego, M., Jamsrandorj, U. & Deters, R. (2016). Blockchain as a Service for IoT. *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 433-436. doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2016.102.

Samie, F., Bauer, L. & Henkel, J. (2016). IoT technologies for embedded computing: A survey. *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pp. 1-10.

Sarkar, S., Chatterjee, S. & Misra, S. (2018). Assessment of the Suitability of Fog Computing in the Context of Internet of Things. *IEEE Transactions on Cloud Computing*, 6(1), 46–59.

Sedjelmaci, H., Senouci, S. M. & Taleb, T. (2017). An Accurate Security Game for Low-Resource IoT Devices. *IEEE Transactions on Vehicular Technology*, 66(10), 9381-9393. doi: 10.1109/TVT.2017.2701551.

Shafi, Q., Basit, A., Qaisar, S., Koay, A. & Welch, I. (2018). Fog-Assisted SDN Controlled Framework for Enduring Anomaly Detection in an IoT Network. *IEEE Access*, 6, 73713-73723. doi: 10.1109/ACCESS.2018.2884293.

Shafique, K., Khawaja, B. A., Sabir, F., Qazi, S. & Mustaqim, M. (2020). Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios. *IEEE Access*, 8, 23022-23040. doi: 10.1109/ACCESS.2020.2970118.

Sharma, V., You, I., Yim, K., Chen, I.-R. & Cho, J.-H. (2019). BRIoT: Behavior Rule Specification-Based Misbehavior Detection for IoT-Embedded Cyber-Physical Systems. *IEEE Access*, 7, 118556-118580. doi: 10.1109/ACCESS.2019.2917135.

Sheikh, T. U., Rahman, H., Al-Qahtani, H. S., Kumar Hazra, T. & Sheikh, N. U. (2019). Countermeasure of Attack Vectors using Signature-Based IDS in IoT Environments. *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 1130-1136. doi: 10.1109/IEMCON.2019.8936231.

Shende, S. & Thorat, S. (2020). A Review on Deep Learning Method for Intrusion Detection in Network Security. *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pp. 173–177.

Siboni, S., Shabtai, A., Tippenhauer, N. O., Lee, J. & Elovici, Y. (2016). Advanced security testbed framework for wearable IoT devices. *ACM Transactions on Internet Technology (TOIT)*, 16(4), 1–25.

Sinche, S., Raposo, D., Armando, N., Rodrigues, A., Boavida, F., Pereira, V. & Silva, J. S. (2020). A Survey of IoT Management Protocols and Frameworks. *IEEE Communications Surveys Tutorials*, 22(2), 1168-1190. doi: 10.1109/COMST.2019.2943087.

Singh, J., Millard, C., Reed, C., Cobbe, J. & Crowcroft, J. (2018). Accountability in the IoT: Systems, Law, and Ways Forward. *Computer*, 51(7), 54-65. doi: 10.1109/MC.2018.3011052.

Siniosoglou, I., Radoglou-Grammatikis, P., Efstathopoulos, G., Fouliras, P. & Sarigiannidis, P. (2021). A Unified Deep Learning Anomaly Detection and Classification Approach for Smart Grid Environments. *IEEE Transactions on Network and Service Management*, 18(2), 1137-1151. doi: 10.1109/TNSM.2021.3078381.

Siriwardhana, Y., Porambage, P., Liyanage, M. & Ylianttila, M. (2021). AI and 6G Security: Opportunities and Challenges. *2021 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*, pp. 616-621. doi: 10.1109/EuCNC/6GSummit51104.2021.9482503.

Soltanieh, N., Norouzi, Y., Yang, Y. & Karmakar, N. C. (2020). A Review of Radio Frequency Fingerprinting Techniques. *IEEE Journal of Radio Frequency Identification*, 4(3), 222-233. doi: 10.1109/JRFID.2020.2968369.

Statista Research Department. (2021). Number of explosion incidents in the United States from 2012 to 2020. Consulted at https://www.statista.com/statistics/785950/number-of-explosion-incidents-in-the-united-states/.

Stine, J. E., Castellanos, I., Wood, M., Henson, J., Love, F., Davis, W. R., Franzon, P. D., Bucher, M., Basavarajaiah, S., Oh, J. & Jenkal, R. (2007). FreePDK: An Open-Source Variation-Aware Design Kit. *2007 IEEE International Conference on Microelectronic Systems Education (MSE'07)*, pp. 173-174. doi: 10.1109/MSE.2007.44.

Stoessel, F. (2021). *Thermal safety of chemical processes: risk assessment and process design*. John Wiley & Sons.

Stolfo, J., Fan, W., Lee, W., Prodromidis, A. & Chan, P. K. (2000). Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection. *Results from the JAM Project by Salvatore*, 1–15.

Strielkina, A., Kharchenko, V. & Uzun, D. (2018). Availability models for healthcare IoT systems: Classification and research considering attacks on vulnerabilities. *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, pp. 58-62. doi: 10.1109/DESSERT.2018.8409099.

Sun, M., Lai, Y., Wang, Y., Liu, J., Mao, B. & Gu, H. (2022). Intrusion Detection System Based on In-depth Understandings of Industrial Control Logic. *IEEE Transactions on Industrial Informatics*, 1-12. doi: 10.1109/TII.2022.3200363.

Surendran, S., Nassef, A. & Beheshti, B. D. (2018). A survey of cryptographic algorithms for IoT devices. *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1–8.

Taşıran, A. C. (2019). Internet of things and statistical analysis. In *Performability in Internet of Things* (pp. 127–136). Springer.

Tavallaee, M., Bagheri, E., Lu, W. & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. *Computational Intelligence for Security and Defense Applications - IEEE-CISDA*, pp. 1–6.

Teixeira, M. A., Salman, T., Zolanvari, M., Jain, R., Meskin, N. & Samaka, M. (2018). SCADA system testbed for cybersecurity research using machine learning approach. *Future Internet*, 10(8), 76.

Tekeoglu, A. & Tosun, A. (2016). A Testbed for Security and Privacy Analysis of IoT Devices. *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 343-348. doi: 10.1109/MASS.2016.051.

Thales Group. (2019). 2020 Thales Data Threat Report Survey. 26. Consulted at https://www.statista.com/statistics/1202640/internet-of-things-security-concerns/.

Tian, Q., Lin, Y., Guo, X., Wen, J., Fang, Y., Rodriguez, J. & Mumtaz, S. (2019). New Security Mechanisms of High-Reliability IoT Communication Based on Radio Frequency Fingerprint. *IEEE Internet of Things Journal*, 6(5), 7980-7987.

Tian, W., Ji, X., Liu, W., Liu, G., Zhai, J., Dai, Y. & Huang, S. (2020). Prospect Theoretic Study of Honeypot Defense Against Advanced Persistent Threats in Power Grid. *IEEE Access*, 8, 64075-64085.

Tiburski, R. T., Moratelli, C. R., Johann, S. F., Neves, M. V., Matos, E. d., Amaral, L. A. & Hessel, F. (2019). Lightweight Security Architecture Based on Embedded Virtualization and Trust Mechanisms for IoT Edge Devices. *IEEE Communications Magazine*, 57(2), 67-73. doi: 10.1109/MCOM.2018.1701047.

Transforma Insights. (2022). Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030. Consulted at https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.

Trappe, W., Howard, R. & Moore, R. S. (2015). Low-Energy Security: Limits and Opportunities in the Internet of Things. *IEEE Security Privacy*, 13(1), 14-21. doi: 10.1109/MSP.2015.7.

Ullah, I. & Mahmoud, Q. H. (2017). A hybrid model for anomaly-based intrusion detection in SCADA networks. *2017 IEEE International Conference on Big Data (Big Data)*, pp. 2160-2167. doi: 10.1109/BigData.2017.8258164.

Ullah, I. & Mahmoud, Q. H. (2021). Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks. *IEEE Access*, 9, 103906-103926. doi: 10.1109/ACCESS.2021.3094024.

U.S. Department of Justice—FBI. (2019). Crime in the United States, 2018. Consulted at https://ucr.fbi.gov/crime-in-the-u.s/2018/crime-in-the-u.s.-2018.

Vaigandla, K., Azmi, N. & Karne, R. (2022). Investigation on Intrusion Detection Systems (IDSs) in IoT. *International Journal*, 10(3).

Valdivieso Caraguay, Á. L., Benito Peral, A., Barona Lopez, L. I. & Garcia Villalba, L. J. (2014). SDN: Evolution and opportunities in the development IoT applications. *International Journal of Distributed Sensor Networks*, 10(5), 735142.

Vangelista, L. (2017). Frequency Shift Chirp Modulation: The LoRa Modulation. *IEEE Signal Processing Letters*, 24(12), 1818-1821. doi: 10.1109/LSP.2017.2762960.

Varshney, G. & Gupta, H. (2017). A security framework for IOT devices against wireless threats. *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, pp. 1-6. doi: 10.1109/TEL-NET.2017.8343548.

Vasserman, E. Y. & Hopper, N. (2013). Vampire Attacks: Draining Life from Wireless Ad Hoc Sensor Networks. *IEEE Transactions on Mobile Computing*, 12(2), 318-332. doi: 10.1109/TMC.2011.274.

Veprytska, O. & Kharchenko, V. (2022). AI powered attacks against AI powered protection: classification, scenarios and risk analysis. *2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, pp. 1-7. doi: 10.1109/DESSERT58054.2022.10018770.

Verma, S. (2018, August, 6). Top 5 Use Cases of IoT in Transportation [html]. Consulted at https://dzone.com/articles/top-5-applications-of-iot-in-transportation.

Vidgren, N., Haataja, K., Patiño-Andres, J. L., Ramírez-Sanchis, J. J. & Toivanen, P. (2013). Security Threats in ZigBee-Enabled Systems: Vulnerability Evaluation, Practical Experiments, Countermeasures, and Lessons Learned. *2013 46th Hawaii International Conference on System Sciences*, pp. 5132-5138. doi: 10.1109/HICSS.2013.475.

Vishwakarma, M. & Kesswani, N. (2022). A Two-Stage Intrusion Detection System (TIDS) for Internet of Things. In *Advances in Deep Learning, Artificial Intelligence and Robotics* (pp. 89–97). Springer.

Wan, J., Yang, J., Wang, Z. & Hua, Q. (2018). Artificial Intelligence for Cloud-Assisted Smart Factory. *IEEE Access*, 6, 55419-55430. doi: 10.1109/ACCESS.2018.2871724.

Wan, J., sXia, M., Hong, J., Pang, Z., Jayaraman, B. & Shen, F. (2019). IEEE Access Special Section Editorial: Key Technologies for Smart Factory of Industry 4.0. *IEEE Access*, 7, 17969-17974. doi: 10.1109/ACCESS.2019.2895516.

Wang, J. (2021). An introduction to wireless technologies in IoT- LPWAN. Consulted at https://www.allion.com/iot-lpwan/.

Wang, L. & Jones, R. (2017). Big data analytics for network intrusion detection: A survey. *International Journal of Networks and Communications*, 7(1), 24–31.

Wang, N., Wang, P., Alipour-Fanid, A., Jiao, L. & Zeng, K. (2019). Physical-Layer Security of 5G Wireless Networks for IoT: Challenges and Opportunities. *IEEE Internet of Things Journal*, 6(5), 8169-8181. doi: 10.1109/JIOT.2019.2927379.

Wang, T., Liang, T., Wei, X. & Fan, J. (2018). Localization of Directional Jammer in Wireless Sensor Networks. *2018 International Conference on Robots Intelligent System (ICRIS)*, pp. 198-202. doi: 10.1109/ICRIS.2018.00059.

Wang, W., Zhu, M., Wang, J., Zeng, X. & Yang, Z. (2017). End-to-end encrypted traffic classification with one-dimensional convolution neural networks. *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 43–48.

Wei, J. (2014). How Wearables Intersect with the Cloud and the Internet of Things: Considerations for the developers of wearables. *IEEE Consumer Electronics Magazine*, 3(3), 53–56.

Weingart, S. H. (2000). Physical security devices for computer subsystems: A survey of attacks and defenses. *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 302–317.

Wurm, J., Hoang, K., Arias, O., Sadeghi, A.-R. & Jin, Y. (2016). Security analysis on consumer and industrial IoT devices. *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 519-524. doi: 10.1109/ASPDAC.2016.7428064.

Xiao, L., Li, Y., Han, G., Liu, G. & Zhuang, W. (2016). PHY-layer spoofing detection with reinforcement learning in wireless networks. *IEEE Transactions on Vehicular Technology*, 65(12), 10037–10047.

Xiao, L., Li, Y., Huang, X. & Du, X. (2017). Cloud-based malware detection game for mobile devices with offloading. *IEEE Transactions on Mobile Computing*, 16(10), 2742–2750.

Xiao, L., Wan, X., Lu, X., Zhang, Y. & Wu, D. (2018). IoT security techniques based on machine learning. *arXiv preprint arXiv:1801.06275*.

Xiaocong, M., Jiao, Q. X. & Shaohong, S. (2015). An IoT-Based System for Water Resources Monitoring and Management. *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2, 365-368. doi: 10.1109/IHMSC.2015.150.

Xingjie, F., Guogenp, W., ShiBIN, Z. & ChenHAO. (2020). Industrial Control System Intrusion Detection Model based on LSTM amp; Attack Tree. *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 255-260. doi: 10.1109/ICCWAMTIP51612.2020.9317477.

Xu, C., Chen, S., Su, J., Yiu, S. M. & Hui, L. C. K. (2016). A Survey on Regular Expression Matching for Deep Packet Inspection: Applications, Algorithms, and Hardware Platforms. *IEEE Communications Surveys Tutorials*, 18(4), 2991-3029.

Xu, L. D., He, W. & Li, S. (2014). Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics*, 10(4), 2233-2243. doi: 10.1109/TII.2014.2300753.

Yamauchi, M., Ohsita, Y., Murata, M., Ueda, K. & Kato, Y. (2020). Anomaly Detection in Smart Home Operation From User Behaviors and Home Conditions. *IEEE Transactions on Consumer Electronics*, 66(2), 183-192. doi: 10.1109/TCE.2020.2981636.

Yang, H., Cheng, L. & Chuah, M. C. (2019). Deep-Learning-Based Network Intrusion Detection for SCADA Systems. *2019 IEEE Conference on Communications and Network Security (CNS)*, pp. 1-7. doi: 10.1109/CNS.2019.8802785.

Yang, Z. & Chang, C. H. (2019). 6LoWPAN Overview and Implementations. *EWSN*, pp. 357–361.

Yao, H., Gao, P., Zhang, P., Wang, J., Jiang, C. & Lu, L. (2019). Hybrid Intrusion Detection System for Edge-Based IIoT Relying on Machine-Learning-Aided Detection. *IEEE Network*, 33(5), 75-81. doi: 10.1109/MNET.001.1800479.

Yin, C., Zhu, Y., Fei, J. & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5, 21954–21961.

Yun, K., Astillo, P. V., Lee, S., Kim, J., Kim, B. & You, I. (2021). Behavior-Rule Specification-based IDS for Safety-Related Embedded Devices in Smart Home. *2021 World Automation Congress (WAC)*, pp. 65-70. doi: 10.23919/WAC50355.2021.9559588.

Zanella, A., Bui, N., Castellani, A., Vangelista, L. & Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1), 22–32.

Zarpelão, B. B., Miani, R. S., Kawakani, C. T. & de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications*, 84, 25–37.

Zhang, J., Shen, C., Su, H., Arafin, M. T. & Qu, G. (2022). Voltage Over-Scaling-Based Lightweight Authentication for IoT Security. *IEEE Transactions on Computers*, 71(2), 323-336. doi: 10.1109/TC.2021.3049543.

Zhang, P. & Jetter, A. (2016). Understanding Risk Perception Using Fuzzy Cognitive Maps. *2016 Portland International Conference on Management of Engineering and Technology (PICMET)*, pp. 606–622.

Zhang, P.-B. & Yang, Z.-X. (2018). A Novel AdaBoost Framework With Robust Threshold and Structural Optimization. *IEEE Transactions on Cybernetics*, 48(1), 64-76. doi: 10.1109/TCYB.2016.2623900.

Zhang, W., Yang, Q. & Geng, Y. (2009). A Survey of Anomaly Detection Methods in Networks. *2009 International Symposium on Computer Network and Multimedia Technology*, pp. 1-3.

Zhang, Z.-K., Cho, M. C. Y., Wang, C.-W., Hsu, C.-W., Chen, C.-K. & Shieh, S. (2014). IoT security: ongoing challenges and research opportunities. *2014 IEEE 7th international conference on service-oriented computing and applications*, pp. 230–234.

Zhao, J. (2017). On Resilience and Connectivity of Secure Wireless Sensor Networks Under Node Capture Attacks. *IEEE Transactions on Information Forensics and Security*, 12(3), 557-571. doi: 10.1109/TIFS.2016.2613841.

Zhao, M. & Barati, M. (2023). Substation Safety Awareness Intelligent Model: Fast Personal Protective Equipment Detection using GNN Approach. *IEEE Transactions on Industry Applications*, 1-9. doi: 10.1109/TIA.2023.3234515.

Zhong, W., Yu, N. & Ai, C. (2020). Applying big data based deep learning system to intrusion detection. *Big Data Mining and Analytics*, 3(3), 181-195. doi: 10.26599/BDMA.2020.9020003.

Zhou, J. (2015). Intelligent Manufacturing——Main Direction of" Made in China 2025". *China Mechanical Engineering*, 26(17), 2273.

Zhou, L. & Guo, H. (2018). Anomaly Detection Methods for IIoT Networks. *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pp. 214-219. doi: 10.1109/SOLI.2018.8476769.

Zhu, S., Ota, K. & Dong, M. (2022). Energy-Efficient Artificial Intelligence of Things With Intelligent Edge. *IEEE Internet of Things Journal*, 9(10), 7525-7532. doi: 10.1109/JIOT.2022.3143722.

Zou, Y., Zhu, J., Wang, X. & Hanzo, L. (2016). A survey on wireless security: Technical challenges, recent advances, and future trends. *Proceedings of the IEEE*, 104(9), 1727–1765.

ÉTS. (2019, January, 3). Smart Residences [html]. Consulted at http://quartierinnovationmontreal.com/en/open-air-smart-living-laboratory/smart-residences.

Čavojský, M., Bugár, G. & Levický, D. (2023). Comparative Analysis of Feed-Forward and RNN Models for Intrusion Detection in Data Network Security with UNSW-NB15 Dataset. *2023 33rd International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 1-6. doi: 10.1109/RADIOELEKTRONIKA57919.2023.10109068.