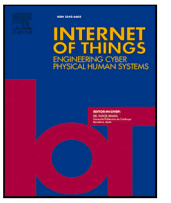
Internet of Things 25 (2024) 101021

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot



Review article

Survey and classification of Dos and DDos attack detection and validation approaches for IoT environments

Mohamed Riadh Kadri [a], Abdelkrim Abdelli [a,*], Jalel Ben Othman [b,c,d], Lynda Mokdad [e]

[a] LSI Laboratory, Faculty of Computer Science, Bp 32 El Alia, Bab Ezzouar, 16111, Algiers, Algeria
[b] Université Paris-Saclay, CNRS, Gif-sur-Yvette 91190, Paris, France
[c] Université Sorbonne Paris Nord, Gif-sur-Yvette 91190, Paris, France
[d] College of Technological Innovation, Zayed University, Abu Dhabi, 144534, United Arab Emirates
[e] UPEC, LACL, F-94010, Creteil, Paris, France

ARTICLE INFO

Keywords:
IoT
DoS attack
DDoS attack
Taxonomy
Metrics
Detection
Tools
Validation
Simulation
Datasets
Machine learning
Mitigation
Analytical
Empiric

ABSTRACT

The Internet of Things (IoT) has emerged over the past ten years as the newest technology trend that is luring researchers and developers from every sector of industry and academia. However, IoT is experiencing a number of security issues that are impeding its rapid development, especially those related to service availability, which has grown into a significant obstacle to be overcome. Denial of service (DoS) and Distributed Denial of Service (DDoS) attacks are among the threats that can disturb even inactivate the functionalities of the IoT networks, like their ability to collect, process, and transfer data. To date, many methods have been proposed to identify, detect, and mitigate such attacks in the IoT domain, while many surveys have been conducted to review and classify these solutions. However, to the best of our knowledge, none of them has conducted a holistic study to review, classify, and correlate both theoretical and practical aspects used in the design and validation of those approaches.

To address this need, we have examined recent and noteworthy research on DoS and DDoS attacks, resulting in the selection of 80 papers to be considered in our study. As a starting point, after identifying in-depth the theoretical aspects commonly used in the detection of such attacks, we provide a comprehensive taxonomy that classifies them. In the second step, we inventoried and produced a complete classification of all the practical aspects used in the validation and evaluation of detection and mitigation approaches, including methods, testbeds, hardware, software, datasets, libraries, and metrics. In the third step, we conducted a technical analysis of the surveyed literature by considering different aspects. As a final step, we perform a statistical analysis in order to discuss various research questions that strive to provide a better insight of the prevalent tendencies in this domain by exploring, for each attack, the most appropriate approach and validation process to consider in dealing with it. The main findings of this analysis show that the research is leaning towards using machine learning, mainly by considering supervised algorithms to detect different variants of flooding attacks. Moreover, simulation appears to be the most operated method to validate the surveyed literature while analytical and empiric validations have been seldom adopted mainly to evaluate classical approaches to detect perception layer attacks.

* Corresponding author.
E-mail addresses: mkadri@usthb.dz (M.R. Kadri), Aabdelli@usthb.dz (A. Abdelli), jalel.benothman@centralesupelec.fr (J. Ben Othman), lynda.mokdad@u-pec.fr (L. Mokdad).

https://doi.org/10.1016/j.iot.2023.101021
Received 9 April 2023; Received in revised form 9 October 2023; Accepted 28 November 2023
Available online 29 November 2023

1. Introduction

The domain of connected objects or what is usually known as the Internet of Things (IoT) has witnessed a large increase in its adoption rate in the last few years [1]. The immense potential of this technology in automating a wide range of tasks across various applications is a key driver behind the growing interest in it, including day-to-day chores, industrial processes, and healthcare, to name a few. Moreover, significant technological advancements in semiconductor manufacturing processes have resulted in microcontrollers (MCs) becoming smaller, more cost-effective, and simultaneously enhancing their processing power and storage capacity. This has paved the way for the integration of MCs in a wide range of devices, allowing them connected, automated, and remotely manageable. As a result, smart homes are turning out more prevalent as smart household appliances are becoming affordable and more accessible. At the industrial level, companies are more and more increasing their reliance on robots as they proved to be an efficient and cost-effective alternative to human labor. Moreover, energy conservation is one of the many fields where IoT can make a significant added value. Smart objects can play a crucial role in reducing energy consumption by remaining active nodes only when required, thus resulting in a significant reduction in overall energy usage. Many countries are working on transitioning cities into smart cities that are eco-friendly in an effort to preserve crucial resources.

The integration of IoT has been progressively on the rise over the years, and its importance in real-world applications is becoming increasingly evident. Numerous statistical studies were made on the IoT adoption trend. According to Statista, the global IoT market size is projected to reach $1.5 trillion by 2030, while there were approximately 10 billion IoT devices in use in 2020, this number is estimated to grow to over 25 billion by 2030. The extensive use of connected devices illustrates the widespread adoption of IoT particularly within the industrial sector, where the IIoT (Industrial IoT) market is projected to expand from $77.3 billion in 2021 to $110.6 billion by 2025. As regards other IoT applications, a recent report by IoT Analytics predicts that smart cities will spend $124 billion on IoT technologies by 2025. These investments aim to enhance urban infrastructure, transportation, and public services. According to another report by IoT Analytics, the worldwide consumer IoT market is anticipated to achieve a value of $104.4 billion by 2023. Moreover, in a study conducted by MarketsandMarkets, it was asserted that as of 2019 86% of healthcare organizations were already using IoT technology. This research anticipates that the global healthcare IoT market will surge to $188.2 billion by 2027. Additionally, research conducted by Grand View Research forecasts that the global IoT is set to achieve a value of $94.44 billion by 2025. In parallel, the global smart agriculture market is projected to reach $15.3 billion by 2027, says Research and Markets.

These promising statistics clearly indicate that IoT is becoming a significant catalyst in technological and industrial development. It will continue to play a crucial role in the future of humanity at all the levels. As a result of this promising success, prevalence and decentralization become the guiding principles of this technology. This is why many security and privacy concerns have risen since the emergence of the IoT. More specifically, the IoT field is more vulnerable to security threats that target its service availability. This include DoS (Denial of Service) and DDoS (Distributed DoS) attacks, which constitute the focal point of this study. Indeed, one of the most significant attributes of smart objects is their ability to automate tasks, streamlining processes to increase efficiency, collect data and transfer it through the network to be managed remotely. Thus, the failure to deliver a service can cause a host of significant issues for applications that rely on it, specifically in the case of critical real-time IoT systems [2].

Furthermore, the domain of IoT introduces an entirely new level of hyper connectivity, leading to the generation of massive amounts of data that require processing, filtering, transfer, and storage. These operations, and hence the obtained results, are closely related to the objects that generate the bigdata [3]. This has created a new set of larger-scale security challenges that need to be addressed and which, in some cases, contradict the core philosophy of the IoT that emphasizes expandability, ease of use, and interoperability.

Add to that, IoT devices have unique constraints that set them apart from conventional computing systems. Many IoT devices mainly operate on battery power and need to be energy-efficient to reduce costs and environmental impact. Moreover, as IoT devices often use MCs and small size memories, their processing and storage capabilities are limited compared to conventional computers, thus restricting their ability to process complex algorithms or perform resource-intensive tasks, making them reliant on cloud or edge computing for more extensive processing. Furthermore, using low-power, low-bandwidth communication protocols like Zigbee or LoRa could restrict the volume of data exchanged and elevate latency. IoT devices are also concerned by heterogeneity which impedes their interoperability, and by environmental constraints (extreme temperatures, humidity), which can affect their performance and longevity.

Therefore, standard threat prevention and mitigation methods are not as effective in the IoT as they are in the traditional computing and networking areas, mainly because of the constraints that IoT objects face, which makes the implementation of these methods challenging, resource-consuming, and, in some cases, unachievable. Furthermore, due to the heterogeneous nature of the IoT, new kinds of threats and attack vectors were investigated and exploited by malicious actors to compromise IoT security. Hence, the development of dedicated security protocols and techniques tailored for the IoT field has become essential to mitigate the imminent risks associated with the widespread adoption of this technology in our daily lives. As a result, significant researches have been devoted to this topic thus far. Indeed, multiple studies were conducted in the literature to accurately assess the risk of DoS and DDoS in IoT and propose prevention and countermeasure techniques to those attacks in order to ensure uninterrupted availability of connected object resources and services. These approaches are, spanning from simple methods that rely on traffic analysis and statistical measures of network traffic to more advanced and intricate solutions that utilize ML (Machine Learning) techniques. Furthermore, for the verification and assessment of the effectiveness of those algorithms, a variety of validation methods, tools, and metrics has been used. However, despite these endeavors, there is still substantial work to achieve to effectively address and tackle

the entirety of DoS and DDoS attack types, as they have not been adequately covered and addressed in the existing body of the literature.

To gain a better understanding of the tendencies and gaps in the current state of the art dealing with DoS and DDoS threats in the context of IoT, the identification and the classification of all the aspects involved in the design, the implementation, and the validation of the existing approaches is an essential step to carry out. The current study endeavors to respond to this research problematic, by conducting, first, a comprehensive analysis and classification of recent and significant works that address the detection of DoS and DDoS attacks. Secondly, a complete taxonomy of the various validation methods and tools used in works is presented, proposing detection, identification, and mitigation approaches for DoS and DDoS attacks, that were mainly published between 2018 and 2023. A comparative analysis of the detection approaches on technical and theoretical aspects is also conducted. Finally, different research questions that strive to correlate the usage of each algorithm with the most suitable validation method and tools as well as the appropriate set of metrics to consider, accordingly is discussed.

The rest of this paper is organized as follows. Section 2 reviews and discusses the most recent and interesting surveys in the literature that address DoS and DDoS attacks in the IoT domain. In Section 3, we define and present several essential background concepts that are relevant to the proposed study. As part of Section 4, we present three taxonomies that classify approaches related to DoS attacks along with their theoretical and systematic aspects. In Section 5, we conduct a comparative analysis of the surveyed literature. Section 6 introduces some research questions, and outlines the main contributory points of this study. Finally, Section 6 summarizes the main contributions of this paper and highlights the future research directions.

2. Related work

Although several surveys have been conducted in the IoT field, only few of them have focused primarily on DoS and DDoS as the main and sole research topic of their studies. We discuss hereafter the most recent and interesting reviews covering either IoT DoS and DDoS attacks, or alongside some other IoT security topics.

One of the first surveys addressing this topic has been carried out by Mosenia et al. in [4]. The authors proposed a reference model and provided definitions of fundamental security concepts and requirements in the context of IoT. Additionally, they reviewed the threats in the edge-side layer of the proposed model and investigated the existing countermeasure techniques for each threat in the literature. The authors observed that the literature did not cover two emerging threats at the time of writing the article: "exponential increase in the number of weak links" and the "unexpected uses of data". In [5], Alaba et al. provide a general introduction to IoT security concepts and proposed a taxonomy in which they divided IoT security issues into four major contexts: Application, Architecture, Communication and Data. Then, for each of the latter, they listed different types of threats and vulnerabilities and mentioned their impact on each aspect. In a similar vein, Kouicem et al. enumerated in [6] the security solutions found in the literature and classified them into two main classes: *classical approaches* which are further distinguished based on three of major security aspects that are confidentiality, privacy and availability; and the class of *new emerging approaches* regrouping solutions based on software defined networking (SDN) and those using blockchain. In [7], Lu et al. surveyed major IoT cybersecurity research topics. The authors first introduced relevant concepts and provided statistics on the number of research articles related to IoT cybersecurity from 2013 to 2017. Subsequently, they put forward a comprehensive taxonomy for classifying attacks targeting the IoT.

In 2019, Chaabouni et al. surveyed Network Intrusion Detection Systems (NIDS) for IoT [8]. Their specific focus was on ML based NIDS (Non Intrusive Detection Systems), as they assert that these systems provide better outcomes when compared to traditional approaches. In a more general aspect, Sengupta et al. explored in [9] IoT and IIoT security threats. They established a vulnerability-based classification of attacks and reviewed these attacks as well as their corresponding countermeasures. They additionally proposed a taxonomy of IoT/IIoT security research areas and highlighted the potential of blockchain-based security solutions in handling IoT/IIoT threats. Moreover, in [10], Arshad et al. identified several countermeasures used against sybil attacks in Wireless Sensor Networks (WSN). These countermeasures are based on different techniques such as encryption, received signal strength indicator (RSSI), trust, artificial intelligence and others. The authors evaluated the advantages and drawbacks of the methods and assessed their compatibility with IoT architecture. In [11], Mohamad Noor et al. conducted a survey in which they analyzed the state of IoT security research from 2016 to 2018, they classified IoT security mechanisms according to five main aspects on which they are based: authentication, encryption, trust management, secure routing and other new technologies. In a survey published in 2020 [12], Lounis et al. focused on attacks targeting IoT wireless infrastructure, especially short-range technologies like Wi-Fi, ZigBee, Bluetooth and RFID. They proposed a taxonomy classifying these attacks and presented some of the existing mitigation solutions and countermeasures to the most common surveyed attacks. In a recent survey [13], Bahaa et al. conducted a systematic literature review on real time IoT security attacks in which, along with enumerating the attacks, they tried to investigate which datasets, ML techniques and variables are used in this context. They also evaluated and discussed which of the current models monitoring real-time IoT security are using DevSecOps. In [14], Krishna et al. provided a comprehensive taxonomy of different IoT threats based on a seven-layer IoT architecture. They also focused on security solutions based on trending technologies like blockchain, fog/edge computing and machine learning that have as a purpose to prevent, detect or mitigate the attack risk.

Regarding DoS and DDoS attacks in the IoT domain, to date only a limited number of surveys have been published. We discuss next the most interesting and recent studies.

Lohachab et al. were among the first to publish a survey [15] in which they assessed various DDoS attacks and classified them into three classes which are the volumetric attacks, protocol attacks, and application attacks. They also inventoried some of the countermeasures proposed in the literature to tackle such threats. In 2020, Dantas et al. proposed another classification of IoT DDoS attacks [16]. The latter contains three main classes of attacks: application layer, resource exhaustion, and volumetric attacks.

Regarding countermeasures, they described SDN-based mitigation strategies and discussed the existing solutions. They also proposed a taxonomy that categorizes them into collaborative and non-collaborative solutions. In [17], Vishwakarma et al. discussed several IoT botnets and their exploitation in recent DDoS attacks. They classify defense mechanisms into three categories: prevention, detection, and mitigation. Al-Hadhrami et al. provided in [18] another classification of DoS and DDoS attack countermeasures, consisting of four main categories: IDS-based solutions, protocol-based solutions, trust-based solutions, and other solutions. For each class, they pointed out the advantages of each solution, as well as the constraints and limitations associated with their use. In [19], Shah et al. investigated the applicability of blockchain as a promising technology to deal with IoT DDoS attacks. They discussed some of the existing blockchain solutions, and concluded their work by presenting some unexplored research topics that could help developing DDoS-resilient IoT networks. In [20], Singh et al. presented a general analysis of DDoS attacks and defense mechanisms in web-enabled computing platforms including IoT, SDN, and blockchain. The authors proposed a taxonomy of DDoS attacks and discussed detection strategies found in the literature while presenting the most common used tools and evaluation metrics for those approaches.

They also provided a comparative analysis of the surveyed detection approaches. The work was concluded by presenting some of the open challenges and future research directions as the authors deduced that there are still many perspectives to be explored in dealing with DDoS attacks. In the last survey [21], Kadri et al. proposed a novel taxonomy that classifies the metrics and the variables used to evaluate DoS and DDoS attack detection approaches. Different research questions were addressed to find out a relationship between each class of metrics, attack types, and also the validation method adopted in the surveyed solutions. The key findings suggest that for some attacks, the existing solutions are mainly shifting towards using specific validation methods as well as a dedicated class of variables and metrics.

To the best of our knowledge, we have not come across any survey in the literature that has conducted a holistic study to identify and classify all the theoretical, technical, and validation aspects connected to approaches used to detect and mitigate DoS and DDoS attacks in the IoT domain. Moreover, another contribution of this study is to explore possible relationships between DoS and DDoS attack types, and aspects connected to detection methods, validation tools, and evaluation metrics used in this context. The main contributions of this research can be summarized as follows:

1. We conducted an analysis of recent literature that resulted in the selection of 80 papers spanning the period 2018–2023.
2. We identify in detail the theoretical aspects used in the design of the detection approaches of DoS and DDoS attacks, then we propose a taxonomy for their classification.
3. We identify and classify in detail the practical and technical aspects used in the validation of both detection and mitigation approaches defined in the surveyed literature. This includes methods, tools (software and hardware), datasets, and libraries used in this context.
4. We provide a comparative analysis of the surveyed solutions based on their technical and theoretical aspects.
5. We conduct a statistical analysis (to gain a deeper understanding of the prevailing trends in this field) to determine the usage rate of each detection algorithm, validation, method, tool, and metric employed in the classification of the surveyed literature relative to each type of attack.
6. We provide a comprehensive study to enhance the comprehension of the most suitable methods, algorithms, and tools to implement in the context of each attack. Furthermore, we investigate the relationships between attack types, detection approaches, validation methods, and tools, as well as the metrics employed in addressing DoS and DDoS attacks in the domain of IoT.

Table 1 summarizes the features of the previous discussed surveys and compares their contributions with the current study.

3. Concepts and background

To ease the understanding of the topic of our paper, we reiterate the definitions of concepts and paradigms that we judged relevant to the context of this study.

3.1. List of acronyms

Table 2 presents the list of alphabetically sorted acronyms cited in this paper.

3.2. Iot environment and cybersecurity

3.2.1. Iot architecture

One of the most common characteristics of IoT is the lack of standardization of its conceptual architecture. As a result, many architectures were proposed in the literature, so far. However, we focus solely on describing the three layer IoT architecture as it is the primary architectural framework considered in the existing surveyed solutions. As its name indicates, it is made of three separated layers that are:

- **Perception layer:** The main function of this layer is to provide the physical infrastructure for communication between connected devices and also to gather information from and about the environment surrounding those objects. For that, it involves the communication technologies needed to transfer information, in addition to the different types of sensors required to perform data sensing and collection. As shown in Fig. 1, a wide range of wireless technologies is used in IoT at this level, including:

Table 1
Comparison of recent surveys in IoT Security.

Reference	Year	Main contribution	DoS/ DDoS	DoS/DDoS attack types	Research questions	Paper selection	Taxonomy	Statistics	Main paper's classification criteria	Main attack's classification criteria	Evaluation of detection approaches	Evaluation of mitigation approaches
[21]	2022	Evaluation metrics for DoS & DDoS attacks detection	✓	20 Attack types	✓	✓	Metrics used in DDoS attacks detection approaches	✓	–	–	–	–
[19]	2022	Blockchain based DDoS mitigation	✓	Packet Flooding	✓	✓	Blockchain DDoS mitigate solutions	–	Following the taxonomy	–	–	✓
[20]	2022	DDoS attacks & defense in web enabled computing platforms	✓	16 Attack types	–		DDoS attacks DDoS defense mechanisms	–	Following DDoS defense mechanisms	Volumetric, protocol, and application	✓	–
[14]	2021	IoT threats and attacks	–	–	–	–	Threats in IoT Attacks in IoT	✓	–	IoT Layer	–	–
[13]	2021	SLR on monitoring real time attacks for IoT	–	–	✓	✓	–	✓	dataset, ML technique Attack Type, Metrics	IoT Layer	–	✓
[18]	2021	SLR on DDoS attacks detection	✓	13 Attack Types	✓	✓	–	–	Solution Type	IoT Layer	✓	✓
[10]	2021	Sybil attack countermeasures in IoT-based WSN	✓	Sybil Attack	✓	✓	–	✓	Countermeasure methods and types, Method frequency, Publication Year	–	–	–
[16]	2020	DDoS mitigation approaches using SDN	✓	Application, Resource Exhaustion, Volumetric.	–	✓	DDoS mitigation solutions in SDN-IoT	✓	Mitigation scenarios, Distribution of solutions.	–	–	✓
[12]	2020	Attacks and defenses in short-range wireless technologies	–	6 Attack types	✓	–	Attacks on short-range communication	–	–	Wireless technology, Attack family	–	–
[17]	2020	DDoS attacking and defense techniques	✓	8 Attack types	–	–	DDoS attacks, DDoS defense solutions.	✓	–	Following the taxonomy	–	✓
[9]	2020	IoT security issues blockchain Solutions	–	5 attack types	–	–	Attacks in IoT IoT security issues	–	–	Physical, Network, Soft & data attacks	✓	✓
[22]	2020	Security, privacy and trust in different IoT layers	–	–	–	–	–	–	–	–	–	✓
[23]	2020	NB-IoT Security	–	3 Attack types	–	–	–	–	–	IoT Layer	✓	–
[8]	2019	IDS for IoT based on ML	–	10 Attack Types	–	✓	–	✓	IDS Type	By Design challenges	–	✓
[11]	2019	IoT Security	–	–	–	–	–	✓	Authentication, Encryption, Trust Secure Routing	IoT Layers	–	–
[7]	2018	A review of IoT cybersecurity topics	–	–	–	–	Cybersecurity attacks on IoT	✓	–	By IoT Layer	–	–
[15]	2018	DDoS in IoT	✓	11 Attack types	–	–	DDoS attacks categories	✓	–	Volumetric, protocol & application attacks IoT layers	–	✓
[6]	2018	IoT Security	–	–	–	–	IoT security solutions	–	Solution Type	–	–	✓
[5]	2017	IoT Security	–	7 Attack types	–	–	IoT security taxonomy	–	IoT Security Architecture type Application domains.	Hardware, Network Infrastructure, Smart Application	✓	–
[4]	2016	IoT Security	–	3 Attack types	–	–	–	–	–	Edge-side Layer, Security requirement	–	–
This work	**–**	***Classification of DoS & DDoS detection and validation approaches in IoT***	✓	***25 Attack types***	✓	✓	***DoS & DDoS attacks detection approaches Validation methods***	✓	***Following the proposed taxonomy***	***IoT layer & attacks class***	✓	**–**

Table 2
List of cited acronyms and their related concepts.

Concept	Acronym
IPv6 over Low-Power Wireless Personal Area Networks	6LowPAN
Accuracy	ACC
ACKnowledgement	ACK
Association for Computing Machinery	ACM
ADAptive Boosting	ADABOOST
Averaged Dependence Estimator	ADE
ML Adversarial attacks	ADV
Artificial intelligence	AI
Application Layer	AL
Application Layer Packet Flooding attacks	ALPF
Artificial Neural Network	ANN
Artificial Neural Network based Self Organizing Map	ANN-SOM
AutoRegressive Integrated Moving Average	ARIMA
BlackHole attack	BH
Bluetooth Low Energy	BLE
Beta Probability Distribution Function	BPDF
Battery Exhaustion Attack	BX
Communication	C
Complete Autoencoder	CA
Center for Applied Internet Data Analysis	CAIDA
Command and Control Server	CCS
Cumulative Distribution Function	CDF
Computational Effort	CE
Intrusion Detection Evaluation Dataset	CICIDS-2017
Confusion Matrix	CM
Clone Node attack	CN
Convolutional Neural Network	CNN
Constrained Application Protocol	CoAP
Continuous Ranked Probability Score	CRPS
Disconnection Attack	DA
DataBase systems and Logic Programming	DBLP
Deep Belief Network	DBN
Distributed DoS	DDoS
development plus security plus operations	DevSecOps
DODAG Information Object	DIO
Spam DIS attack	DIS
Deep Learning	DL
Deep Neural Network	DNN
Destination Oriented Directed Acyclic Graph	DODAG
Denial of Service	DoS
Detection Performance Metrics	DPM
Data Rates	DR
Desynchronization Attack	DS
Decision Tree	DT
Deceptive traffic redirection	DTR
Energy Indicators	EI
Energy Overhead	EO
Exponentially Weighted Moving Average	EWMA
F1-Score	F1S
Fragmentation-Based buffer exhaustion attacks	FBA
Correlation-based Feature Selection	FCBFS
Fuzzy C-Means Clustering	FCM
Fusion Ensemble-based	FE
Federated Learning	FL
False Negative	FN
Feedforward Neural Network	FNN
False Positive	FP
False Positive Rate	FPR
Generative Adversarial Network	GAN
Gradient Boosting Decision Tree	GBDT
Graphics Processing Unit	GPU
Greddy node attack	GR
Gated Recurrent Unit	GRU
Generalized Total Variation metric	GTV
Graphical User Interface	GUI
Hello Flood	HF
Hilbert-Huang Transform	HHT
Host IDS	HIDS

(*continued on next page*)

Table 2 (*continued*).

Hypertext Transfer Protocol	HTTP
Integrated Circuit	IC
Internet Control Message Protocol	ICMP
Industrial Control system	ICS
IDentification	ID
Intrusion Detection System	IDS
Institute of Electrical and Electronics Engineers	IEEE
Industrial IoT	IIoT
Iterative Model Averaging	IMA
Iterative Model Averaging-Gated Recurrent Unit	IMA-GRU
Intrinsic Mode Function	IMF
IMPersonation Attack	IMP
Internet of Things	IoT
Internet of Things-Device Under Investigation	IoT-DUI
Internet of Things-Device Under Test	IoT-DUT
Internet Protocol	IP
Intrusion Prevention System	IPS
Jamming Attacks	JA
Jøsang's Subjective Logic	JSL
K-Nearest Neighbors	KNN
Kolmogorov Smirnov test	KS
Link Flooding	LF
Low power and Lossy Networks	LLNs
Logistic Model Tree	LMT
LOng RAnge wireless modulation	LoRa
Long Range Wide Area Network	LoRaWAN
Local Repair	LR
Low Rate DDoS attacks	LRDDOS
Logistic ReGression	LRG
Long Short-Term Memory	LSTM
Media Access Control	MAC
MicroController	MC
Multidisciplinary Digital Publishing Institute	MDPI
Machine Learning	ML
Maximum Likelihood Estimation	MLE
MultiLayer Perceptron	MLP
Multiclass Neural Network	MNN
Message Queuing Telemetry Transport	MQTT
Memory & Storage	MS
Naïve Bayes	NB
NarrowBand-Internet of Things	NB-IoT
Numeric Computing Environment	NCE
NeiGHbour attack	NGH
Network Indicators	NI
Network IDS	NIDS
Network Layer	NL
Network Simulator	NS
Network Security Laboratory - Knowledge Discovery in Databases	NSL-KDD
Other Metrics	O
Packet Capture and Generators Tools	PCGT
Packet-Drop Attacks	PDA
Packet Flooding	PF
Perception Layer	PL
Precision	Prc
Physical Unclonable Function	PUF
Quality of Service	QoS
Random Access Memory	RAM
Random Cut Forest	RCF
REpresentational State Transfer	REST
Random Forest	RF
Radio Frequency IDentification	RFID
Rank attack	RK
Recurrent Neural Network	RNN
Rider Optimization Algorithm	ROA
Read Only Memory	ROM
Routing Performances	RP
Routing Protocol for Low Power and Lossy Networks	RPL
Robust Random Cut Forest	RRCF

(*continued on next page*)

Table 2 (*continued*).

Received Signal Strength Indicator	RSSI
ReplaY attack	RY
Signal Based Indicators	SBI
Supervisory Control and Data Acquisition	SCADA
Software Defined IoT	SD-IoT
Software Defined Network	SDN
Stacking Ensemble	SE
Selective Forwarding	SF
SinkHole	SH
Systematic Literature Review	SLR
System On a Chip	SoC
Self-Organizing Map	SOM
Support Vector Machine	SVM
Sybil attack	SY
SYNchronize	SYN
Transmission Control Protocol	TCP
THRouGhPuT	THRGPT
Time Indicator	TI
TIme petri Net Analyzer	TINA
True Negative	TN
True Positive	TP
Trusted Platform Module	TPM
True Positive Rate	TPR
User Datagram Protocol	UDP
Utility Function	UF
Vehicular Ad hoc Network	VANET
Virtual Machine	VM
Version Number	VN
Wide-Area Network	WAN
WormHole	WH
WIreless FIdelity	WI-FI
Worst Parent	WP
Wireless Sensor Network	WSN
eXtreme Gradient Boosting	XGBOOST

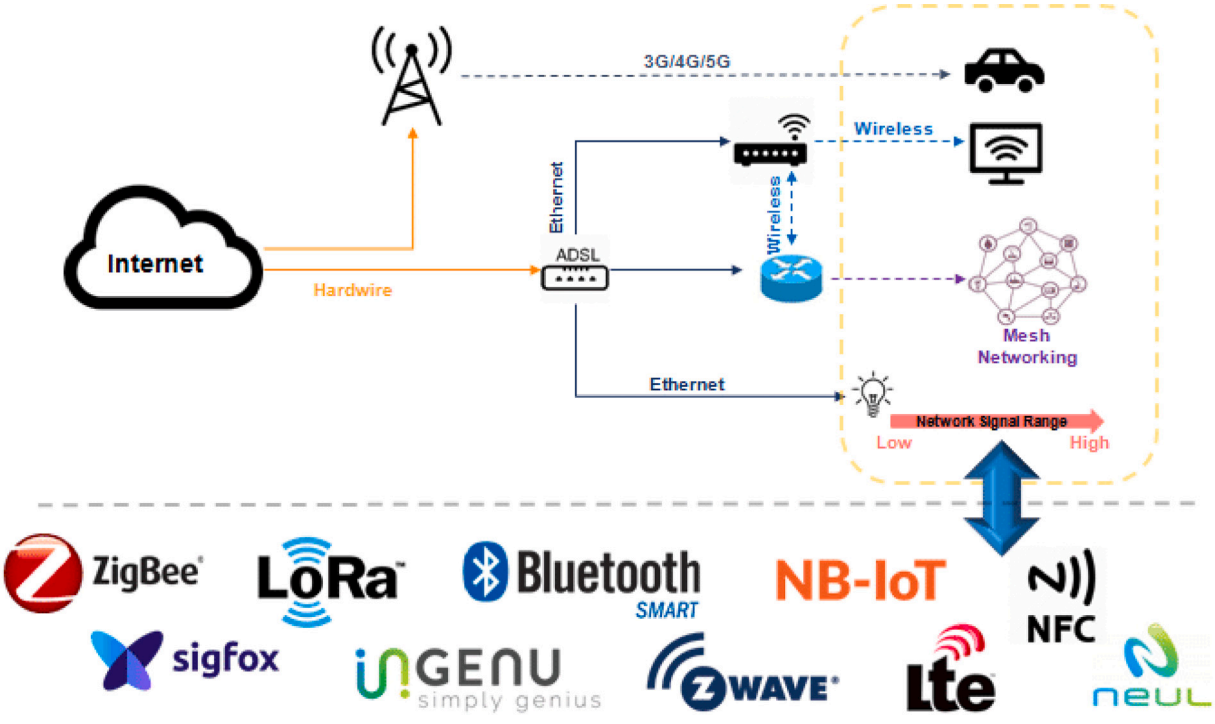


Fig. 1. IoT network environment and technologies [24].

-Wi-Fi: The most commonly used wireless technology for IoT devices that connect to the Internet.

-Bluetooth: A short-range wireless technology that is commonly used for connecting devices such as smartphones, wearables, and smart home devices.

-LoRaWAN: A long-range, low-power wireless technology used for IoT devices such as sensors, smart meters, and other industrial applications.

-ZigBee and Z-Wave: Low-power wireless technologies that are commonly used in smart homes and industrial automation.

-Cellular networks: IoT devices use cellular networks (like 4G and beyond) to connect to the Internet. The 5G, which refers to the fifth-generation, is the latest generation of wireless technology for cellular networks. It represents a significant advancement over its predecessor, 4G, in terms of speed, capacity, latency, and overall performance. 5G networks are designed to provide faster data transfer rates, lower latency, and support a massive number of connected devices, allowing for the deployment of large-scale IoT applications and services.

- **Network layer:** This layer has as a core function to enable the coordination and the connection between smart objects that use different communication technologies and bind them into one single network. This is done using higher-level network protocols such as RPL (Routing Protocol for Low-Power and Lossy Networks). It can offer an additional function which consists in preprocessing the collected data.

RPL which is becoming a standard for IoT, is a distance-vector routing protocol that is based on Directed Acyclic Graph (DAG) topology and multiple metrics for routing path selection. Moreover, RPL is designed in a way that each node maintains a routing table that is updated based on information received from its neighbors, in addition to route optimization and localized repair mechanisms that make this protocol scalable and suitable to be used in IoT networks that are in consistent change. There were, however, a number of security concerns associated with RPL, and different attacks have been identified, such as the *local repair*, the *version number*, and *the spam dis* attacks.

-**Application layer:** This layer works as an interface between the user and the IoT network. It involves the running applications that provide the network's services and resources to the user using specific protocols, like, HTTP, REST, MQTT and COAP. MQTT is becoming the standard protocol to transport data applications in IoT networks. It is designed to efficiently transport data between IoT devices and applications with minimal network bandwidth and processing requirements. MQTT operates on a publish–subscribe model, where devices or applications can publish messages to a central broker, which then distributes the messages to any subscribers who have expressed interest in them. This allows devices and applications to communicate with each other in a scalable and efficient way, without requiring a direct connection. MQTT is widely used in a variety of IoT applications, including home automation, industrial automation, and remote sensing. Its lightweight nature and support for QoS make it ideal for use in low-power, low-bandwidth devices, where minimizing network usage and power consumption are critical factors. In spite of this, MQTT has been the target of several potential DDoS attacks in the literature.

3.2.2. Security challenges in the IoT

With the advancement of IoT technology, the scope of devices and applications that can be interconnected has expanded, posing significant security challenges:

- **Lack of standardization:** IoT devices come from various vendors and use different operating systems, communication protocols, and data formats. This lack of standardization makes it challenging to develop a unified security framework and increases the risk of vulnerabilities.
- **Weak authentication and authorization:** Many IoT devices come with default credentials that users often fail to change, leaving them vulnerable to hacking. Additionally, authorization policies may be inadequate, allowing unauthorized access to sensitive data.
- **Insecure communication:** IoT devices often communicate over unsecured channels, making them vulnerable to eavesdropping, interception, and man-in-the-middle attacks.
- **Data privacy and protection:** IoT devices collect and transmit large amounts of sensitive data, such as personal and financial information, health data, and location data. The lack of proper encryption and data protection mechanisms can result in data breaches and privacy violations.
- **Firmware and software vulnerabilities:** IoT devices have firmware and software that may contain vulnerabilities that can be exploited by attackers.
- **Physical security:** IoT devices can be physically tampered with, stolen, or destroyed, leading to data loss, financial losses, and potential safety risks.
- **Lack of security updates and patches:** Many IoT devices lack security updates and patches, leaving them vulnerable to emerging threats and attacks.

Overall, the combination of factors such as the complexity of device interconnections, limited computing power, lack of security standards, and exposure to physical and network-based attacks makes IoT more vulnerable to security breaches. With billions of IoT devices in use, each with its unique hardware, software, and network configurations, managing and securing them all is a daunting task. As a result, attackers are more interested in targeting IoT devices because they are often less secure than traditional computing devices such as laptops or servers. IoT devices, such as smart home appliances, wearables, and industrial control systems, are typically designed to be low-cost and easy to use, which often means sacrificing security features. Additionally, many IoT devices are connected to the Internet and communicate with other devices, which creates a larger attack surface for attackers to target.

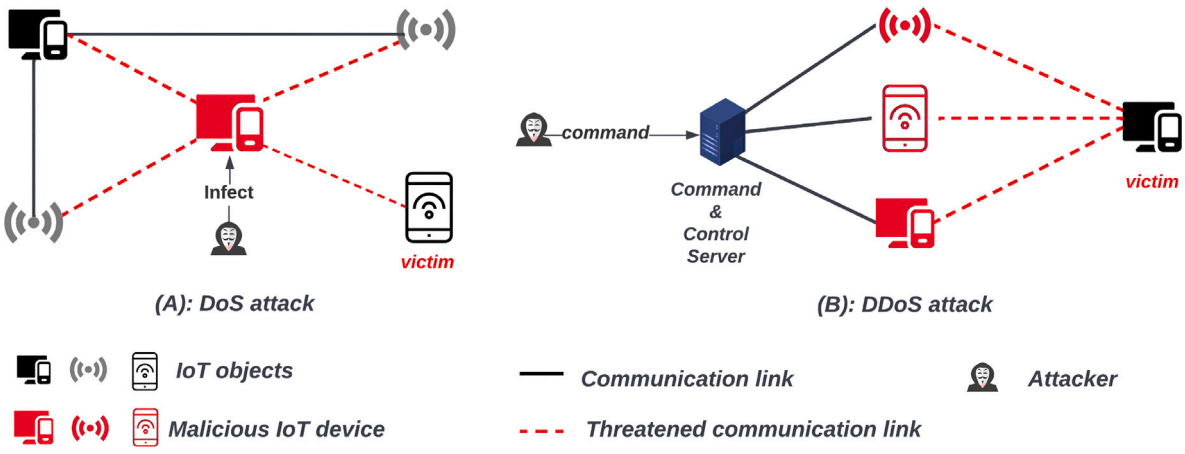


Fig. 2. DoS and DDoS attack in IoT environment.

Another reason is that they can use such devices as a stepping stone to launch larger attacks, such as DDoS attacks, to overwhelm servers and take down entire websites or networks. Finally, IoT devices often store sensitive data making them an attractive target for attackers who can use this information for identity theft, fraud, or other malicious purposes. Since its emergence, the IoT has witnessed several harmful attacks over the years that have highlighted the security challenges faced by this emerging technology. In what follows, we recall some of them:

- **Mirai Botnet:** The Mirai botnet is a malware that infects IoT devices and turns them into a network of bots that can launch distributed DDoS attacks. The botnet was responsible for some of the largest DDoS attacks in history, including the attack on DNS provider Dyn in 2016.
- **BrickerBot:** BrickerBot is a malware that targets IoT devices with default credentials and permanently disables them by overwriting their firmware. The malware has caused significant damage to IoT devices, with estimates of over 10 million devices affected.
- **Stuxnet:** While not specifically targeted at IoT devices, Stuxnet is one of the most infamous cyber attacks in history, known for targeting industrial control systems (ICS) and causing physical damage to Iran's nuclear program. The attack has raised concerns about the security of IoT devices used in critical infrastructure.
- **BlueBorne:** BlueBorne is a set of Bluetooth vulnerabilities that can be exploited to take control of IoT devices, including smartphones, laptops, and smart TVs. The vulnerability allows attackers to execute arbitrary code, steal data, and spread malware to other devices.

The next subsection discusses the most prevalent attacks against smart devices, namely DoS and DDoS attacks, which constitute the main attacks covered in this study.

3.3. DoS and DDoS attacks in IoT environment

A DoS attack targets a victim (which can varies from an individual device to an entire network) with the primary goal of impeding its intended function. This is achieved through various techniques, including communication blocking, flooding the victim with an excessive volume of service requests, overwhelming it with traffic or data, ultimately rendering it unavailable. As shown in Fig. 2.(A), a DoS attack in the context of IoT typically consists in an infected node that belongs to the network that can cut off completely the victim node's ingoing and outgoing communications.

DDoS attack is a more complex variant of DoS in which the attack is launched from multiple sources simultaneously. In most cases, those sources are mainly smart objects that were compromised (often called zombie devices) to be controlled remotely by a malicious entity using a Command and Control Server (CCS) which is often another compromised device. This forms what is called a *botnet*, used by the CCS to launch an attack against a specific target (See Fig. 2.(B)).

In addition to the aforementioned IoT security challenges, DoS and DDoS attacks in an IoT environment can have some specific characteristics that differ from traditional network attacks even from other IoT attacks. These make them particularly harmful in IoT environments. Here are some notable characteristics:

- *Limited processing power:* Many IoT devices have limited processing power, memory, and bandwidth, which makes them easy targets for DoS and DDoS attacks.
- *Distributed nature of the attacks:* DDoS attacks are carried out using multiple devices, often in different locations. In IoT environments, this can include thousands of compromised devices, which can cause a much more significant impact than a single attack.

- *Hard to detect:* IoT devices are often located in remote or unmonitored locations, making it challenging to detect a DoS or DDoS attack in a timely manner. This delay in detection can allow the attack to continue for longer, causing more damage.
- *Cascading effect:* IoT devices are often interconnected and can share data with each other. When a device is taken down by a DoS or DDoS attack, it can cause a cascading effect, taking down other connected devices as well, leading to more significant damage.
- *Botnets:* IoT devices can be easily compromised and added to botnets, which can be used to launch DDoS attacks and generate large volumes of traffic to overwhelm a target system.

While some DoS and DDoS attacks are directly inherited from classical networks (i.e flooding attacks), there are many attacks that are specific to smart object constraints (e.g. battery exhaustion), used protocols like RPL-specific attacks (e.g., version number and rank attacks) and those related to application layer protocols, like MQTT flooding attacks. In the subsequent section, we explore, for each layer of the system, the prevalent and commonly identified DoS and DDoS attacks in the surveyed literature.

3.3.1. Perception layer attacks

The main objective of such attacks is to hinder smart objects from carrying out their sensing function (i.e. data collection) and/or disturb communications on the medium to obstruct the transfer of the gathered data to the upper layers. We expose hereafter the main perception layer attacks addressed in the surveyed literature:

- **Jamming Attacks (JA):** In an IoT jamming attack, an attacker might use a device called a jammer to flood the airwaves with radio frequency noise, preventing any IoT devices in the area from receiving or sending any data. This attack can also be carried out on the MAC layer. We can distinguish three forms of jamming, which are: the *constant jamming* (i.e. the attacker emits the jamming signal continuously), the *reactive jamming* (i.e the attacker emits the jamming signal only when a transmission is detected on the medium), and the *random jamming* (i.e the attacker emits the jamming signal randomly on the medium). In the context of IoT, JA can be particularly damaging, as many IoT devices rely on wireless communication to function. This can cause disruptions in critical services such as healthcare, transportation, and energy management systems, as well as in everyday applications such as smart homes.

- **Battery exhaustion (BX):** As it name indicates, this attack targets the energy-constrained objects in an IoT network for which an attacker tries to drain their batteries using different techniques like sleep-prevention and power-consuming service requests; ultimately resulting in their shutdown. BX is a common issue in IoT devices, particularly those that are designed to be low-power and operate on battery power for extended periods. It can be particularly harmful in cases where the device is critical to a system's operation or is used for monitoring purposes. For example, in a medical device, such an attack could have life-threatening consequences.

- **Greedy node attacks (GR):** The attack is called "greedy" because it involves an attacker consuming more than their fair share of the device's resources, leaving little or nothing for legitimate users or processes. A greedy behavior attack targets IoT devices, particularly those with limited network bandwidth. A malicious node (called a greedy node) tries to exploit vulnerabilities in the medium communication protocols in order to circumvent the fair-network-throughput-sharing mechanisms applied in the network, and consequently reserve all or most of the network's throughput to itself, thereby preventing the other nodes from getting enough throughput to communicate properly [25]. The specificity of this attack is that it is hard to detect because it does not disrupt an ongoing communication, and in a distributed domain, it is often perceived as a network overload. In a large-scale IoT deployment, a greedy behavior attack can have a cascading effect, affecting other devices and the entire network, leading to widespread disruption and potentially causing significant financial and reputational damage.

- **Disconnection Attacks (DA):** A disconnection attack in the context of IoT refers to a type of cyberattack that aims to disrupt or to sever the connection between IoT devices and the network or the Internet. This attack can be carried out by an attacker who gains unauthorized access to the communication channel between the devices and the network and blocks or interferes with the transmission of data packets. The impact of a DA can be severe, for example, if IoT devices are used for monitoring and controlling critical infrastructure such as power grids or transportation systems, a DA can cause widespread disruption and damage.

-**DeSynchronization attacks (DS):** In the context of IoT security, DS attacks refer to a type of cyberattack that aims to disrupt the synchronization of devices within an IoT network. In IoT, devices often rely on time synchronization protocols to communicate with each other effectively. These protocols ensure that all devices in the network have the same understanding of the current time, allowing them to coordinate their actions and exchange data seamlessly. A DA seeks to disrupt this synchronization by sending false time signals or manipulating the time information in the network, by putting for example, the RFID tag in a desynchronized state resulting in errors, or even system crashes.

3.3.2. Network layer attacks

At this level, attackers attempt to exploit existing vulnerabilities in the routing protocols to disrupt data transfer in the IoT network. As there are many types of attacks targeting this layer, we group them into four main classes based on common aspects between those attacks. We provide in what follows a definition of the main characteristics of each group:

-1-**Packet-drop attacks:** In this type of attacks, malicious nodes intentionally discard some of or all of the packets they are supposed to forward to their intended destination. They are particularly hard to detect in the context of IoT because of the lossy nature of the connection links in this domain. The main attacks identified within this class are:

- *Selective forwarding (SF):* In a typical IoT network, devices mainly communicate with each other over wireless links. As these devices are often resource-constrained, they rely on other devices to forward their messages to the appropriate destination. In SF attack, an attacker who has gained access to one or more IoT devices selectively forwards some packets and drops others, thereby compromising the security and the reliability of the IoT.
- *BlackHole attack (BH):* A BH attack in IoT refers to a type of attack where an attacker intercepts and drops all the packets passing through the compromised IoT device.
- *NeiGHbour attacks (NGH):* In this attack, an attacker gains access to a device by being physically close to it. In the context of RPL, for example, the attacker impersonates neighboring nodes to disrupt routing paths by forwarding unmodified DIO (DODAG Information Object) messages in the network.

-2-**Flooding attacks:** In this class, malicious nodes try to exhaust nodes' computational resources and/or network links capacity by generating a huge amount of packets. Flooding is particularly effective in IoT networks because many IoT devices have limited processing power and memory, and may not be able to handle the large volume of traffic generated by this attack. Additionally, many IoT devices are connected to the network through wireless channels, which is more vulnerable to congestion and interference. We identify the following attacks within this group:

- *TCP/UDP/ ICMP Packet Flooding (PF):* TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are both commonly used transport protocols in IoT devices. In a TCP flooding attack, the attacker sends a large number of TCP SYN packets to the target device, which then responds with a SYN-ACK packet. However, since the attacker does not respond with the final ACK packet to complete the handshake, the target device ends up waiting for this packet and becomes overwhelmed with the number of incomplete connections. In a UDP flooding attack, the attacker sends a large number of UDP packets to the target device, causing it to consume resources in processing each packet. ICMP (Internet Control Message Protocol) is another protocol used in IoT devices, primarily for diagnostic and error reporting purposes. In an ICMP flooding attack, the attacker sends a large number of ping requests (Echo Request) to the target device, which responds with an Echo Reply packet. This consumes a significant amount of the target device's resources, leading to a DoS situation.
- *Link Flooding (LF)*: In LF attacks, the attacker sends a large number of data packets to the network, with the goal of consuming the available bandwidth and overwhelming the network infrastructure.
- *Fragmentation-based Buffer exhaustion Attack (FBA)*: In the FBA attack, the attacker sends fragmented packets to the IoT device, with each fragment containing a small portion of the overall payload. The device is designed to reassemble the fragments into the original message, but if the device's buffer is not large enough to handle all the fragments, it can become overwhelmed and can crash.
- *Low Rate DDoS attacks (LRDDOS):* . In a low-rate DDoS attack, multiple compromised IoT devices are used to generate a relatively small volume of malicious traffic to overwhelm a target system or network. Unlike high-rate DDoS attacks (PF) that aim to flood the target with a massive amount of traffic, low-rate attacks focus on consuming resources more subtly. The attack traffic is usually generated at a rate that is below the threshold of traditional DDoS mitigation techniques, making it harder to detect and mitigate. The malicious traffic generated by compromised IoT devices can take various forms, including simple connection requests, bogus requests for resources, or specific attack vectors targeting vulnerabilities in the target's infrastructure.
- *Spam DIS attack (DIS):* In RPL, a device called a "DODAG root" is responsible for managing the routing paths between devices in the network. The DODAG root periodically sends out "DODAG Information Solicitation" (DIS) messages to discover new devices in the network and maintain the routing topology. In a spam DIS attack, an attacker floods the network with a large number of fake DIS messages, overwhelming the DODAG root and causing it to become unresponsive. This can result in the network being unable to properly route data.
- *ReplaY attack (RY):* The RY attack is a type of cyberattack in which an attacker intercepts and records a legitimate data transmission in an IoT network and then replays that same data to impersonate a legitimate device. This type of attack can be used to gain unauthorized access to a system or to trick the system into performing unintended actions. In IoT, RY attacks can be particularly dangerous because many devices communicate using wireless protocols that are vulnerable to interception. For example, an attacker could capture the signal sent by a smart lock when a legitimate user unlocks the door, and then replay that same signal to unlock the door at a later time. Although RY is not a typical DoS attack, it is worth noting that in some cases, if a RY floods a system with a large number of repeated messages, it might cause a temporary DoS due to resource consumption. But this scenario would be more of an unintended consequence rather than the primary goal of the RY attack.

-3-**Deceptive traffic redirection attacks:** In this class, malicious nodes utilize various techniques to redirect traffic to themselves, such as cloning the identity of other nodes, using multiple fake identities, or manipulating other nodes' selection of the optimal path. In overall, we have identified the following attacks within this category:

- *Version Number (VN), Local Repair (LR), RanK (RK) and Worst Parent attack (WP):* VN, LR, RK, and WP are specific DoS attacks that have been identified in the context of RPL. In VN attack, an attacker sends RPL messages with a higher version number than the one currently in use by the network. The attacker hopes to disrupt the routing of the network by causing inconsistencies in the version numbers used by the nodes to build and maintain the routing of the network. This can result in packets being dropped or sent to the wrong destination. The LR attack is a type of attack where a malicious node tries to disrupt the routing process by creating a temporary routing loop. The attack works by sending a fake routing message that claims that

the malicious node has a better route to a particular destination than the existing route. This causes the neighboring nodes to update their routing tables and start sending traffic to the malicious node, which drops the traffic, creating a blackhole in the network. The WP is an attack in which a malicious node intentionally chooses a sub-optimal path to the root node to forward its data packets, resulting in a loss of routing performance. In RK attacks, a malicious node falsely increases or decreases its rank value in the RPL network. The rank value is used to determine the position of a node in the network and the preferred path towards the root. By increasing its rank value (Rank increment attack), a malicious node becomes the preferred parent of its neighbors and can redirect traffic to a compromised node. By decreasing its rank (Rank decrement attack), a malicious node becomes the non-preferred parent of its neighbors, thus isolating certain nodes and cause a network partition.

It is also possible to identify attacks within this class that are not typically DoS or DDoS, but are nevertheless able to cause a DoS, as for example:

- *Hello Flood attack (HF):* The HF attacks can be caused by a node that broadcasts a Hello packet with very high power, so that a large number of nodes even far away in the IoT network choose it as the parent node. The malicious node can then corrupt or redirect the data to an illegitimate destination.
- *SYbil attack (SY)*: SY attack refers to a situation where a malicious actor creates multiple fake identities or nodes to manipulate the behavior of the network. The attacker can use these fake identities to gain control over a significant portion of the network, overwhelm the network with false data, or steal sensitive information.
- *SinkHole (SH)*: In an SH attack, the attacker aims to intercept or modify the data being transmitted by traffic from legitimate IoT devices to a malicious device controlled by the attacker by typically exploiting vulnerabilities in the IoT devices or their communication protocols. The attacker may be able to steal sensitive data from the compromised devices or use them as a foothold to launch further attacks against other systems.
- *WormHole attack (WH):* In WH attacks, an attacker creates a shortcut between two distant points, thus creating a tunnel through the network, which bypasses normal communication channels and allows it to eavesdrop on or modify data passing through the network. WH attacks can be particularly dangerous in IoT networks, where devices may be spread out over a large area and may not be closely monitored.
- *Clone Node attack (CN):* A CN attack refers to the process of creating a replica of an IoT device with the intention of gaining unauthorized access to a network or system. The attacker creates an identical copy of a legitimate IoT device, including its unique identifier or MAC address, and uses it to impersonate the original device. Once the clone device is set, the attacker gains the ability to intercept or modify data transmitted between legitimate IoT devices and the network.

-4- **ML adversarial attacks *(ADV)*:** This attack is generally used against ML-based detection approaches, in which an attacker tries to induce an ML model failure in detecting the attack by either interfering with the training process itself by introducing deceptive false data or crafting specially designed data to mislead a particular model.

3.3.3. Application layer attacks

At this level, the attacker tries to hinder the IoT objects from providing their services to the end user. The attacks identified within this layer are similar to the network's flooding attacks, with the difference being in the used protocol (e.g. MQTT, CoAP, and HTTP in the context of the application layer). In the sequel, we identify them as *"Application Layer Packet Flooding attacks" (ALPF)* to distinguish them from those related to the network layer.

4. Proposed taxonomies and methods classification

This section starts with an overview of the methodology used to select the papers considered in our study. Then, we proceed to the presentation of the different taxonomies that identify and classify the different theoretical and practical aspects used in the design and the validation of the approaches defined in the surveyed literature.

4.1. Paper selection methodology

The selection of the papers considered in this survey has been conducted by following three rigorous steps:

- **Keywords selection:** Before looking for relevant papers, we produced a list of keywords to use in the paper collecting phase, then we created multiple combinations using those keywords in order to get more accurate results. The list of keyword includes the following: *Attack; BlackHole; Bluetooth; DDoS; Denial of Service; Distributed Denial of Service; DoS; Flood; Greedy; IDS; Internet of Things; IoT; Jamming; LoRa; NB-IoT; RFID; Selective Forwarding; Selfish; SigFox; SinkHole; Sybil; Vanet; WormHole; WSN; ZigBee.*
- **Paper collecting:** During this phase, we started by gathering papers indexed and published by reputable venues such as ACM, IEE Explore, MDPI, Science Direct, and Springer, based on the list of defined keywords. We used DBLP as our main search engine along with main editors' engines. In addition, we mainly focused on papers published between 2018 and 2023. However, we also included some earlier papers that we deemed relevant to our survey's context. In total, we collected approximately 126 papers by the end of this phase.

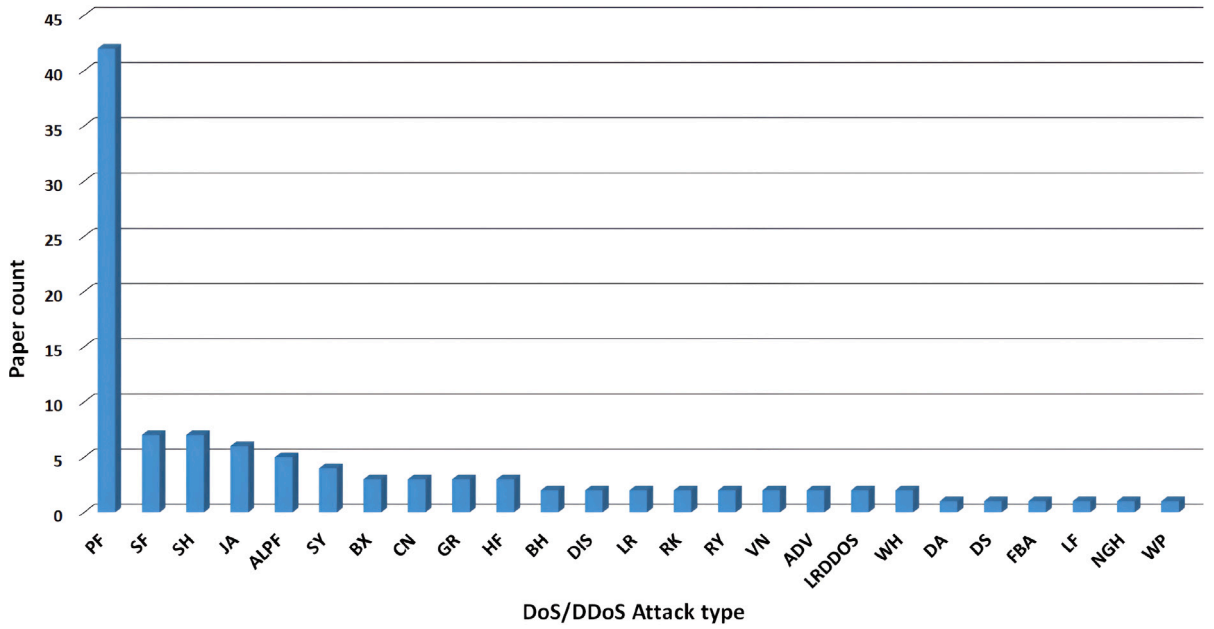


Fig. 3. Attack types count in the surveyed papers.

- **Papers filtering and selection:**
  In this phase, we proceeded to select the most relevant papers for our study. After reviewing carefully the text of all the collected papers, we kept only papers that have as a main topic IoT DoS and DDoS attacks or attack types that can result in a DoS or DDoS in the IoT domain. The chosen papers covered novel attack methods and prevention techniques related to DoS or DDoS attacks to one or multiple types of those said attacks. As a result, we ended up with a total of 80 papers addressing 25 different attack types: 13 papers addressed 5 perception layer attacks, 20 network layer attacks have been addressed through 64 research papers, while ALPF attack was covered by 5 papers. To give a general idea about the content of the surveyed papers, we have found that:

  – 32 papers came up with a detection only approach for the addressed DoS and DDoS attacks.
  – 7 papers proposed both detection and identification approaches to the attacks.
  – 19 papers presented a detection and a remediation (which can be either preventive or mitigation) solution to the attacks.
  – 12 Papers proposed only a remediation solution.
  – 7 papers proposed extensive approaches that can detect, identify, and mitigate the addressed attacks.
  – Finally, 3 papers proposed a new DoS and DDoS attack approach or highlighted protocol-specific vulnerabilities to existing DoS and DDoS attacks.

Fig. 3 presents the counts of papers that covered each type of attack. As one can notice, the PF attack is the most addressed in the surveyed literature followed by the SF attack.

4.2. Detection techniques taxonomy

In this section, we identify all the algorithms used in detecting DoS and DDoS attacks in the context of IoT. Then, we proceed to their classification according to common theoretical aspects. Hence, a general taxonomy is elaborated and discussed. Note that only papers that addressed detection approaches have been included in this classification. As shown in Fig. 4, the developed taxonomy intends to list, categorize, and organize in-depth the theoretical aspects to better understand the tendencies and techniques in use in the surveyed literature. As soon as all the aspects used in the detection solutions were inventoried, we were able to categorize them into two main classes, which are: the class of *Not ML-based approaches* that regroups all the solutions that do not use any form of ML to achieve the detection, and the second one that gathers all the solutions based on running ML algorithms. Note that one approach may combine different aspects in its solution design.

4.2.1. Not ML-based detection approaches

Not ML-based detection approaches refer to classical algorithms that do not use ML in detecting DoS and DDoS attacks. This primarily involves analyzing network traffic to identify patterns or anomalies that serve as indicators of an ongoing attack. As attacks are constantly evolving, and attackers are finding new ways to bypass traditional detection methods. Therefore, the trend is

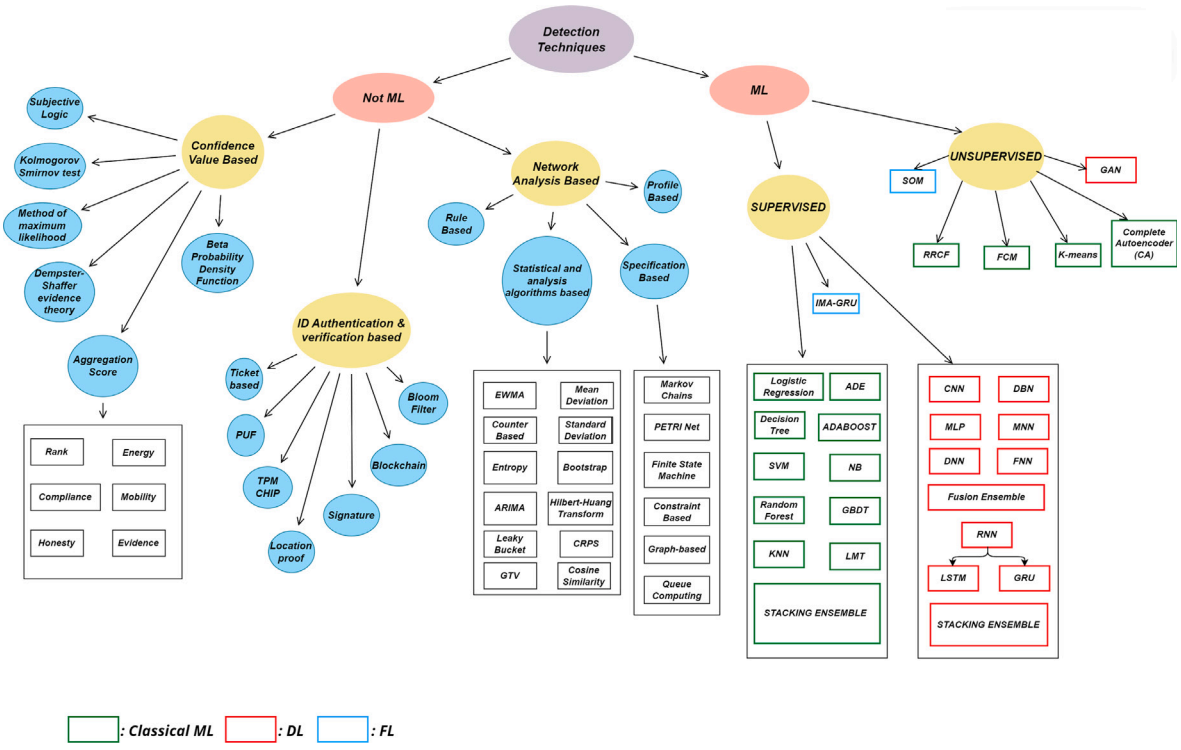


Fig. 4. Detection approaches taxonomy.

to combine different aspects in the design of each solution to improve the detection. In our taxonomy, Not ML-based methods are classified into three main classes, which are:

- **Confidence value based:** This category of methods includes all detection techniques that rely on calculating confidence or trust parameters to determine afterward whether a node is considered sane or malicious based on predefined threshold values. A confidence value refers to the level of certainty that a given detection algorithm has in its assessment of whether a particular event, behavior, or a node represents a genuine attack. This value can be expressed as a probability, percentage, or another numerical score, and is typically based on a variety of factors including the type and frequency of network traffic, the behavior of nodes, and the presence of known indicators of compromise. A high confidence value indicates that the detection algorithm is highly certain that an attack is underway, while a lower value may suggest that the activity in question could be due to other factors, such as legitimate network traffic or system glitches. Different algorithms in the surveyed literature have been used to compute these confidence parameters:

  - **Beta Probability Distribution Function (BPDF):** is a probabilistic function that is used in the security context mainly to predict the sanity of a node based on its behavior, as the parameters of the function are constantly updated according to the change of the node's behavior. Specifically, to compute the confidence value, the BPDF is integrated over a range of values that correspond to an attack. For example, if we know that attacks typically produce values of x in the range [0.9, 1], we can integrate the BPDF over this range to compute the probability of an attack occurring, used then as a confidence value. The advantage of using the beta distribution in DoS detection is that it can model the uncertainty associated with the occurrence of an attack, and can provide a more nuanced measure of confidence than simple threshold-based methods. Additionally, the BPDF can be updated as new data is collected, allowing for real-time detection of attacks. However, BPDF has limited applicability and flexibility while being sensitive to prior assumptions. The solutions presented in [26,27] to detect SH attack used BPDF on the compliance degree history for a node in the network to establish its reputation.
  - **Method of maximum likelihood:** It is a statistical approach that can be used to evaluate the confidence value in attack detection. This approach involves determining the probability of observing the data that is obtained from the system being monitored, assuming that it was generated by a specific distribution. Generally, we use the log-likelihood to calculate a confidence value for the attack detection. Then, a threshold for the confidence value is set to determine when an attack is detected. However, this method is known to be prone to overfitting and very sensitive to outliers in the data, which can skew the results. Moreover, it lacks of robustness comparatively to BPDF, for instance.

- **Dampster-Shaffer evidence theory:** is a mathematical framework for reasoning with uncertain information that enables the calculation of a value that represents a certain degree of "belief" by combining evidence from different sources, such as intrusion detection systems, network logs, and user behavior analysis, following a specific set of combination rules as defined by the theory. Th theory has been shown to be effective in evaluating the confidence value of attack detection systems. This theory assigns a belief function to each source of evidence. These belief functions can then be combined using a mathematical formula to calculate a single belief function that represents the overall confidence in the attack detection system. However, in addition to its complexity, this theory is sensitive to the choice of prior knowledge, which can significantly impact the results. Moreover, it assumes to collect sufficient independent evidence sources to make an accurate assessment of the confidence value. This may be challenging in the case of real-world cybersecurity scenarios. In our context, this theory has been used in [26,27] to detect SH attacks by computing a value specific to each node that can serve as a proof of its sanity (or its maliciousness).
- **Jøsang's Subjective Logic (JSL):** is a probabilistic logic that provides the necessary rules and guidelines to mathematically formalize and represent situations where the uncertainty and the unreliability of information sources should be taken into account before taking any kind of decision, making this approach suitable for the detection of malicious nodes in certain DoS and DDoS attacks (e.g. SF, SH and VN attacks) [28]. JSL uses a belief function, that takes into account both objective and subjective evidence to calculate a measure of confidence, called the degree of belief. Objective evidence could include factors such as the type and severity of the attack, while subjective evidence could include factors such as the reputation of the attacker or the quality of the attack detection system. However, in addition to its complexity, using JSL may lead to inconsistencies and uncertainties in the evaluation of confidence values. Moreover, this theory requires a significant amount of data to be effective.
- **Kolmogorov Smirnov test (KS):** It is a statistical hypothesis test used to determine the likeliness of two samples originating from the same probabilistic distribution. The KS test works by comparing the Cumulative Distribution Function (CDF) of the algorithm's output to the CDF of a reference distribution. The test calculates the maximum absolute difference between the two CDFs, determining the confidence value. If the two CDFs are significantly different, then the confidence value is low. The KS test is useful for evaluating the confidence value when the probability distribution of the data is unknown or cannot be modeled accurately. However, the KS test does have limitations, such as being sensitive to sample size and being less powerful than other hypothesis tests in some cases. In the context of our survey, KS was applied in [29] to resolve a trust-based problem in relation to the detection of BH and CN attacks.
- **Aggregation score:** Some approaches found in the literature combine multiple trust parameters to get an aggregated confidence score for each node of the IoT network [27,30]. There are several aggregation functions that can be used for this purpose, including, the maximum, the mean, the median, the weighted average. The choice of the function to use depends on the specific requirements of the attack detection system and the characteristics of the detectors. The aggregated values may include the rank, the compliance, the honesty, the energy, the mobility and the evidence values. For example, in [30], to detect SY, a node i computes a confidence score for a node j at time (t) which is mainly a weighted sum of its calculated mobility (generated based on a predefined list and the RSSI), honesty (Number of bad experiences occurred between i and j) and the energy (proportional to j's remaining energy percentage) values at (t).

- **ID authentication and verification based methods:**
  ID authentication and verification techniques are not typically used to detect DoS and DDoS attacks but can help to improve the detection by completing the main detection algorithm. It is common for detection approaches to consider the verification of authenticity. This may include the identity of the nodes forming the IoT network, the information presented by the nodes on their own state, or the global network state. A further distinction between these approaches can be made based on the type of verification and authentication mechanism used:

  - **Ticket based:** This approach relies on the generation of a cryptographic token that is used to verify and authenticate nodes in the IoT network. For example, the solution defined in [31] uses ticket generation and verification to distinguish sane nodes and detect malicious nodes participating in BX attacks.
  - **Physical Unclonable Function (PUF):** Based on the principle that each integrated circuit (IC) has its unique physical proprieties that makes it different from any other IC, PUF takes advantages of this principle to generate, for each input it gets, a response that depends on the previously mentioned proprieties of the IC that this function is implemented on. The solution presented in [32] uses PUF to detect SY attacks, while the approach in [33] combines PUF with blockchain, signature and ML for PF detection.
  - **Trusted Platform Module (TPM) Chip:** TPM is a micro-controller that has as a main function to offload the cryptographic calculations from the main device's CPU as it is specifically designed to deal with the latter. Much like the previously discussed approaches, the author in [30] make use of this chip to create a unique ID for the nodes in the IoT network to enable the detection and mitigation of SY attacks.
  - **Location proof:** Location proof can be used to detect and identify the source of the attack. To implement this approach, IoT devices can be equipped with location-aware sensors, to provide geographic location. This latter is used to establish a baseline of expected traffic patterns for the device. When a DoS attack occurs, it typically involves a flood of traffic from a particular source. By comparing the incoming traffic patterns against the expected patterns based on the device's location, abnormal traffic patterns can be detected and flagged as potential DoS attacks. Some solutions in the surveyed

literature [34–37] used IoT nodes location as a tool to detect DoS attacks like BX, CN and SY. As an example, in [34], the authors proposed to detect and approximately locate an attacker (that tries to exhaust a node's battery by sending multiple request messages using different identities), by verifying that the requests are sent from a constant range of positions relatively to that of the victim node. Two approaches are used: either by deploying some receivers in the network or by performing the distance calculation internally on each node receiving the requests. Both approaches rest on computing the RSSI value for that communication. The works in [33,38] have adopted this technique to detect a variety of DoS attacks like BH, PF and CN attacks.

- **Blockchain:** is a technology that provides enhanced features for authentication like decentralization, efficiency and transparency. Its distributed operating principle makes it well-suited for enhancing the reliability and performance of protecting IoT resources from DDoS attacks. Blockchain technology can help to detect DoS attacks by establishing a distributed ledger of all the IoT devices in a network. This ledger can store data about each device, including its IP address, MAC address, and other identifying information. Whenever an IoT device sends or receives data, the blockchain can be used to track the traffic and detect any unusual patterns. This can include sudden spikes in traffic, repeated requests for the same information, or other anomalies that may indicate a DoS attack. Despite its potential benefits in detecting and mitigating DoS attacks in IoT, blockchain technology also introduces several potential disadvantages, such as scalability issues, cost and complexity of the solution, and high energy consumption. As part of our survey, we quote the following works [33,39–41] related to PF attacks. For example, in [39], a blockchain-based verification mechanism is used to run on a server to detect a potential misbehaving node that tries to flood the IoT network with messages. As message transmission history is saved in the blockchain alongside the node's ID information, the server proceeds in case of misbehavior detection to add the malicious node to a blacklist on the blockchain that is propagated to all the other IoT nodes in the network.
- **Bloom filters:** A Bloom filter is a probabilistic data structure that can be used to efficiently test whether an element is a member of a set. It offers a space-efficient representation of a set by using a bit array and a set of hash functions. It is important to note that while Bloom filters can aid in detecting DoS attacks in IoT, they are not foolproof and have limitations. False positives can occur, where legitimate traffic is flagged as malicious. Additionally, Bloom filters cannot provide detailed information about the attack or its source, but rather act as an initial filter for identifying potential threats.

- **Network analysis based:**
  Network analysis can be an effective approach for detecting DoS attacks in IoT systems as IoT devices are typically designed to interact with each other and with other systems over the Internet. To detect DoS attacks, such algorithms monitor network traffic and look for anomalies in its patterns. For example, if there is a sudden spike in traffic volume or a large number of requests coming from a single point, it might suggest the presence of a DoS attack. Network analysis can also identify unusual packet patterns, such as malformed packets or those with incorrect headers, which are commonly used in DoS attacks. This analysis results in the calculation of different parameters characterizing the network state, making it possible to decide the occurrence of an attack. However, there are several potential drawbacks of using network analysis as an increase in network traffic, storage and computation costs, and false positives. Moreover, the lack of standardization can make it difficult to develop uniform approaches. This can lead to a fragmented security landscape, with different devices and networks using different security measures and tools. Among network analysis algorithms, we recognize four categories:

  - **Rule based:** Rule-based methods involve setting thresholds and monitoring traffic for anomalous patterns that exceed these thresholds. The rules include traffic volume threshold, protocol anomaly detection, traffic pattern analysis, geolocation blocking, blacklist/whitelist, behavior analysis, and so on. These rules can be implemented using various tools, such as IDS, firewalls, and traffic analysis tools. However, rule-based methods suffer from scalability as they require manual updates and maintenance for each new attack vector. Moreover, they are time and resource consuming to settle in and-consuming, while they generate high false positives in addition to their inability to detect unknown attacks. Some approaches in the surveyed literature have considered such methods by defining a set of rules that the network traffic should meet [42–45].
  - **Profile based:** This approach consists in the pre-profiling of each IoT node in the network by the IDS node. After the creation of nodes profile, the IDS actively checks the current profile of the nodes and compares it with the previously established ones to detect misbehaving nodes [46]. Node profiling can help to identify nodes that are consuming excessive resources, such as CPU or memory, or that are experiencing high network traffic. However, using profiling has several disadvantages as for example a communication overhead, limited visibility on DDoS attacks. More specifically, there could be a difficulty in defining normal behavior for each device in the network in dynamic IoT environments where devices may behave differently depending on the context. Furthermore, node profiling can itself be vulnerable to adversarial attack, as malicious actors may be able to modify the behavior of devices in the network to evade detection.
  - **Specification based:** Such techniques consider formal or semi-formal models to specify the behavior of the network, and hence compute key parameters used in the detection process. Formal specifications can help in ensuring the correctness, reliability, and robustness of the detection mechanisms. When applied to DoS detection in IoT, such methods imply defining the system under consideration by specifying the components involved in the IoT network, including the devices, sensors, actuators, and network infrastructure and identifying the potential sources of DoS attacks and the desired

properties of the detection system. Subsequently, the latter are transformed into formal requirements that need to be precise, clear and capable of being verified. For example, a requirement could be "The system should detect and mitigate a DoS attack within 30 s with 95% accuracy". To this aim, a formal specification language is used to describe the expected behavior of the IoT system and its components. This could involve defining the states, events, and transitions in the system, as well as the communication protocols and data flows. The descriptions of various types of DoS attacks that the system may encounter are then modeled, and the behavior of the IoT system against the specified requirements and the modeled attacks is then analyzed using formal verification techniques. This involves techniques such as model checking or theorem proving to ensure that the system satisfies the desired properties and remains resilient to known attack patterns. However, using formal specifications is very challenging, as modeling formally the complete IoT system is not easy, in addition to the complexity and the costs of running such approaches. Furthermore, the latter are known to not respond well to scalability, and to dynamic systems. In the context of our survey, we quote few works that took advantage of this approach. For example, in [47] *Petri nets theory* has been considered to model and analyze Zigbee networks, in presence of normal and abnormal behaviors. Hence, thresholds values have been determined for some traffic parameters (Energy, collision rate, packet drop ratio) to detect greedy behavior attack. Moreover, the authors in [48] used *Markov chains* to analyze, detect and predict DDoS attack in 5G IoT networks based on the evolution of a device's state. In [42], the authors considered *finite-state machine* to detect RK, SH, LR, NGH and DIS attacks. Furthermore, both [49] and [50] utilized *graph theory* to model traffic flows in their ML model development. Also, approaches based on *queue computing* were proposed in [51] and [38], both deal with PF and ALPF attacks. Finally, [27] has implemented constraint based specification in the context of SH detection.

– **Statistical and analysis algorithms based:** Statistical methods can be used to detect anomalous traffic patterns and identify potential DoS attacks. Quantitative or statistical measures are used to determine thresholds for network parameters using algorithms ranging from simple to complex. This involves monitoring metrics such as packet rates, packet sizes, traffic volume, and network flow characteristics. Deviations from normal traffic behavior can be identified using techniques like anomaly detection, entropy-based analysis, statistical modeling, and time-series analysis. However, such approaches can, depending on the method, involve collecting and analyzing large volumes of data, which can be challenging in resource-constrained IoT devices. Furthermore, such techniques can hardly adapt to new attacks patterns as they rely on threshold calculation, leading to the increase of false alarms. Finally, as statistical analysis typically involves collecting and analyzing data from IoT devices, this raises privacy concerns, as sensitive information may be involved. Multitude statistical techniques have been employed in the literature ranging from simple to more sophisticated algorithms. However, in most of the papers, these techniques have been combined either together or with algorithms from other classes to provide comprehensive detection frameworks.

  For example, *counter based algorithms* were applied in the detection of the PF attacks [52,53], the SF attack [54], the SH attack [27], and the ALPF attack [38]. A counter-based algorithm is a common approach consisting in counting various metrics to detect abnormal patterns or behaviors that may indicate an ongoing DoS attack. In other solutions, *mean deviation based algorithms* have been exploited in [38,55] and [56] to respectively detect GR, ALPF, and PF attacks. Other works [38,46,50,55] have considered the *standard deviation* to detect GR, PF, ALPF and PF attacks. These algorithms are used to calculate detection thresholds based on multiple parameters, such as throughput and transmission power.

  In [31] the *leaky bucket algorithms* has been considered in the detection of BX attacks. The bucket being represented as *a counter* incremented by *the energy required to serve an incoming request* and decremented by a *predefined amount at a fixed time interval*. This counter is afterward compared to a threshold to determine whether the node is under an attack. Moreover, the authors proposed in the same paper another approach based on the *Exponentially Weighted Moving Average* (EWMA). EWMA is a statistical method used for smoothing data by assigning exponentially decreasing weights to older observations while giving more importance to recent data points. When it comes to detecting DDoS attacks in IoT, an EWMA-based approach needs to select the key parameter, such as network traffic volume or packet rate, to consider as an indicator of normal IoT system behavior. When the variation between two data points for this parameter exceeds a predefined threshold, then a potential DDoS attack alert is triggered. In [31], EWMA has been applied taking as a key parameter the energy. The authors concluded that both EWMA and the leacky bucket were equally suitable for the detection of BX attacks.

  Some interesting works [57–59] have considered the *Entropy* to detect PF attacks. Entropy is used for traffic analysis by measuring the randomness or the unpredictability of data. In the context of network traffic, normal traffic patterns tend to exhibit certain levels of entropy, while DoS attacks often introduce unusual patterns. Entropy can consider various network traffic features, such as packet size, inter-arrival times, source/destination IP addresses, protocol fields, or payload characteristics. For example, the authors of [57] proposed to actively analyze the entropy of packets' source and destination IP and also their source and destination ports. The obtained entropy is afterward compared to a threshold previously defined during the initialization phase of the deployed solution in the IoT network.

  The *Continuous Ranked Probability Score* (CRPS), is another statistical analysis algorithm that has been considered in DoS detection. CRPS is a statistical metric commonly used in the evaluation of probabilistic forecasts. It measures the accuracy of probabilistic predictions by comparing the predicted cumulative distribution function (CDF) to the observed data. While CRPS is not typically used directly for detecting DoS attacks, it can be applied indirectly as part of a broader anomaly detection framework. For example, in [60], the authors used the CRPS to compare between real-time traffic and normal traffic distribution that was established in a pre-deployment phase to detect PF attacks.

The *Autoregressive Integrated Moving Average (ARIMA)* is another statistical analysis model commonly used for time series analysis and forecasting. It relies on time series data to either better understand the dataset or to predict future trends. ARIMA is well known to be good only for short-term forecasting while being poor at predicting turning points and computationally expensive. ARIMA can be used to analyze certain aspects of IoT data that might be indicative of DoS attacks. In the context of our survey, ARIMA has been considered only in [44] to detect multiple DoS attacks (BH, SF, SH, HF, CN, SY, WH, VN, RK, RY, WP, DIS, and LR) for which it has provided good results that were above 80% detection rate for most attacks with the exception of BH (60%) and LR (40%).

Furthermore, *Bootstrap*, which is a statistical technique for estimating quantities about a population is considered in [61] for traffic analysis and dataset generation in the context of an ML-based detection approach of SH attacks. The *Bootstrap* method has gained popularity due to its simplicity and flexibility. It does not require strong assumptions about the underlying data distribution, making it applicable in situations where traditional parametric methods may not be suitable. However, *Bootstrap* relies on the assumption that the original dataset is representative of the population from which it was drawn.

Other algorithms have been considered in the literature. In [62], the authors took advantage of the *Generalized Total Variation* metric (GTV) to detect LRDDOS attack. The GTV metric is a mathematical concept used to measure the variation or changes in a signal or a dataset. In our context, it is used to analyze network traffic patterns and identify abnormal or anomalous behavior by capturing changes in traffic volume, packet size and inter-arrival times, etc. In [63] the authors consider the Cosine similarity in a SDN based solution to detect PF attacks. The Cosine similarity is a metric commonly used in natural language processing and information retrieval to measure the similarity between two vectors. In our case, the cosine similarity of the packet-in message rate at boundary SD-IoT switch ports is computed to determine whether DDoS attacks occur in the IoT. Finally, the *Hilbert-Huang Transform* (HHT) was applied in [29] to decompose data signals in WSN-IoT networks into Intrinsic Mode Function (IMF) components to enable the detection of BH and CN attacks. The HHT is a signal processing technique that is used for analyzing non-linear and non-stationary signals. The obtained IMFs represent different scales or oscillatory modes present in the data. HHT can be applied to detect anomalies or patterns in IoT, including DoS attacks. These features could include energy distribution across different frequency bands, modulation patterns, or any other characteristic that is indicative of normal or malicious activity.

4.2.2. ML-based detection approaches

ML is a sub-domain of artificial intelligence that covers all the algorithms that allow automated learning by a machine without it being explicitly programmed. This is usually done by feeding a large amount of data to a specific set of ML algorithms to generate a model based on computed features from that data. This, applied to our context, enables the approaches based on ML to have a better ability to adapt and detect multiple variations of DoS and DDoS attacks. Deep Learning (DL) is a branch of ML that is based on artificial neural networks (ANNs) and advanced multi-layer learning techniques that can provide better results than classical ML in suitable cases. Federated Learning (FL) is a distributed variant of ML in which multiple devices collaborate for the training of a specific ML model. Compared to classical approaches, ML can provide several advantages when it comes to detecting DoS and DDoS attacks in IoT environments. Indeed, ML models provide scalable real-time detection and fast response and mitigation while reducing false positives. Moreover, they can learn from historical attack data or retrained with the latest data to enhance their accuracy and effectiveness in identifying, preventing, and adapting to new attack patterns. Although ML can be a valuable tool for detecting and mitigating DoS and DDoS attacks in IoT systems, it also has some drawbacks:

- Data scarcity: ML models require large amounts of qualitative data to learn effectively. In the context of IoT security, acquiring sufficient labeled data for training ML models can be challenging. This scarcity of data can hinder to accurately detect and respond to new or evolving attack patterns.
- Data quality and noise: Data used in ML algorithms can be subject to noise, inconsistencies, missing values, and outliers impacting negatively their performances. Cleaning the data can be complex and time-consuming.
- Adversarial attacks: ML models are susceptible to adversarial attacks, where malicious actors intentionally manipulate input data to deceive the model or cause it to make incorrect predictions.
- Computational requirements: Implementing complex ML algorithms can be computationally intensive for IoT devices. Furthermore, ML model training is a time consuming task requiring an accurate tuning of the ML model parameters to achieve an effective detection.

Many solutions in the literature have considered ML in the detection of DoS and DDoS attacks, by using classical algorithms, DL ones, or either FL based techniques. However, regardless of the type of ML used, these solutions can be further categorized based on whether the approach is supervised or not. In the following, we discuss some interesting and recent solutions that have been proposed for each of these two classes of models.

1-Supervised ML approaches:

In supervised ML, the data given as input to the algorithms to train upon are labeled, which means that each set of input is mapped to a known output. Therefore the goal here for the algorithm used in the detection approaches is to learn from that data and to generate a regression/classification model to use afterward for real-time DoS and DDoS attack detection. The majority of the ML-based solutions proposed in the literature heavily rely on supervised ML algorithms. Primarily because of their effectiveness in intrusion detection solutions and promising results achieved by advancements in the field of AI domain. [8]. Indeed, supervised ML

is more effective in identifying and classifying DoS attacks accurately as it has been trained to detect specific types of DoS attacks that are prevalent in IoT networks. This targeted learning approach enables the model to focus on specific attack characteristics, enhancing its ability to identify and mitigate DoS attacks. By leveraging labeled training data, these models can capture the complex relationships between input features and the corresponding output labels (normal or attack). This accuracy is crucial in DoS detection, where timely and accurate identification of attacks is essential for maintaining the availability and integrity of IoT systems. In the papers we surveyed, a multitude of supervised techniques has been applied and tested. For each category of ML (classic, DL, FL) we discuss in the sequel the most interesting approaches.

Regarding classical supervised ML approaches, we quote the use of the following algorithms:

- **Decision Tree (DT):** DT is a popular algorithm used for both classification and regression tasks. It is a flowchart-like tree structure where each internal node represents a feature. Each branch represents a decision rule, and each leaf node represents the outcome. DT learns from the training data by recursively partitioning the data based on the values of the features. It selects the most informative feature at each step to split the data into smaller subsets, aiming to maximize the information gain or decrease the impurity of the subsets. DTs are easy to understand and can capture complex nonlinear relationships between features while handling missing values. However, DT can be prone to overfitting and bias in certain cases. DT was tested in plenty of approaches [49,64–71] for the detection of PF attacks; mainly alongside other ML classifiers where it provided good results, especially in [67] where it came out as the best performing classifier.
- **Random Forest (RF):** RF is a widely adopted ensemble learning method used in ML for both classification and regression tasks. The approach is rooted in the concept of aggregating predictions from multiple DT to generate accurate predictions Each DT is constructed by selecting a subset of the training data through a process called bootstrapping (random sampling with replacement). Once all DT are built, predictions are made by aggregating the results from individual trees. RF model is less prone to overfitting and robust to outliers compared to single DT. However, RF can be memory-intensive and computationally expensive, especially when dealing with large datasets and a large number of trees. Based on our survey, RF algorithm appears to be the most commonly used ML algorithm in this context. Many researches have tested its efficiency either solely [61], or combined with other approaches to select the most important features to be fed to other classifiers [70], or compared with other algorithms, as in [40,41,49,56,64–69,71–73]. RF was mainly tested to detect PF and SH attacks. According to [61], the authors argued that RF is well suited for their approach to detect SH attack because it has more probability to give higher accuracy than SVM.
- **Support Vector Machine (SVM):** SVM has been widely used in various domains. It is a powerful algorithm, but it can be computationally expensive for large datasets. SVM is particularly effective in dealing with complex datasets and can handle high-dimensional feature spaces. The main idea behind SVM is to find an optimal hyperplane that separates the data points of different classes in the feature space. SVM works under the assumption that the data can be linearly separable. Support vectors are the data points that are closest to the decision boundary (hyperplane). These points play a crucial role in defining the hyperplane and are used to make predictions. SVM has been adopted in several solutions through the literature to detect mainly PF. In [74], the authors based their approach exclusively on SVM. They claimed that SVM is better suited for the detection of attacks launched from multiple sources than other ML classifiers such as NB and KNN. They showcased a comparison of the accuracy of those classifiers to support this claim. Other works such as [49,64,65,67,68,73,74] have conducted experiments comparing SVM with other ML classifiers. While SVM generally performed well, these studies reported slightly inferior performances of SVM compared to the alternative classifiers.
- **Naive Bayes classifier (NB):** NB is a classification algorithm based on the application of Bayes' theorem with the "naive" assumption of independence among the features. Bayes' theorem describes the probability of an event based on prior knowledge or conditions related to the event. In the context of classification, it helps to calculate the probability of a certain class label given the observed features. During the training phase, the algorithm learns the probabilities of each feature belonging to each class label based on the frequency of each class label in the training data. NB is generally recommended for large volumes of data. It is a fast and uncomplicated algorithm. It was considered in PF detection approaches where it was mainly used in comparison with other classifiers, as in [64,66,68,69,71] where it performed poorly due to its attribute-independence hypothesis.
- **Averaged Dependence Estimators (ADE):** ADE was introduced to address the attribute-independence problem of the NB classifier to develop substantially more accurate classifiers, at the cost of a modest increase in the amount of computation. In our context, ADE has been used in [75] to deal with PF attacks. Its evaluation showed that those estimators improved drastically the performance of the NB based classifier with the averaged two-dependence estimator and the multi-scheme (combination of averaged one-dependence and two-dependence estimators) having the best results in overall for the proposed approach.
- **K-Nearest Neighbor (KNN):** KNN is used for both classification and regression tasks. It does not make any assumptions about the underlying data distribution. When making a prediction for a new data point, KNN looks for the K nearest neighbors to that point in the feature space. The distance metric used determines the neighbors. The predicted label for the new data point is the majority class among the K neighbors. In KNN, the selection of the appropriate value for K is crucial. A small K can make the model sensitive to noise, while a large K can introduce bias. Moreover, the choice of distance metric affects how the algorithm measures the similarity between data points. KNN is relatively simple to understand and implement. However, its performance can be affected by the curse of dimensionality and the need for appropriate preprocessing and parameter selection. KNN was mainly tested to detect PF attacks where it was compared to other classifiers. Its performance was found to be good in those approaches [65,69–71]. However, [65] states that this algorithm is not recommended to be used in the context of IoT DDoS attack detection due to its compute-intensive characteristics.

- **Logistic ReGression (LRG)**: LRG is a widely used algorithm in ML for binary classification problems. It models the relationship between a set of input variables and a binary output variable. As LRG assumes no feature selection and a linear relationship between variables, it limits its ability to capture complex relationships that may exist in the data. In addition, it is sensitive to outliers, unable to handle missing values and prone to overfitting or underfitting. Despite these limitations, LRG remains a useful and interpretable algorithm. In our context, LRG has been explored in [65,70,71] to address PF attacks where it has been compared with other ML classifiers. In [70], LRG outperformed both KNN and SVM in accuracy comparison.
- **ADABOOST:** Boosting is a technique used in ML to reduce prediction and classification error rate. It works by combining multiple weak learning models (models with poor classification accuracy) into one model called a strong learning model; it is generally used with DT. ADAptiveBOOST (ADABOOST) is an ML algorithm that implements the boosting principle by re-calculating the parameters of the tree with the goal being the minimization of the exponential loss function; this is done at each iteration. Hence the "adaptive" ability of the algorithm. ADABOOST was used solely in [76] to detect ALPF attacks where it provided promising results (95.13% accuracy, 90.97% recall, 99.14% precision, and 94.88% F1-score for the MQTTset dataset). In another work [69], ADABOOST was implemented alongside other ML/DL algorithms to detect PF attacks where it outperformed DT, KNN, RF, NB and MLP classifiers.
- **Gradient Boosting Decision Trees (GBDT):** GBDT is another implementation of the boosting technique based on DT that was used in our context to detect PF attacks. It was considered in [72] where it was compared to other ML classifiers. Results were encouraging as it outperformed RF and SVM. Interestingly, a variant of this classifier called *XGBoost* performed better than the original GBDT in this paper. It was also considered in other solutions [40,41,49,56,66,69] where it presented good results in overall. More particularly, in [66], the authors proposed a new feature selection method called Fast Correlation-based Feature Selection (FCBFS) that resulted in 99.84% accuracy when using the XGBoost classifier on NSL-KDD dataset that contains PF attack samples alongside other non-DoS/DDoS attacks. However, one of the drawbacks of XGboost and boosting in general is the longer training times especially when dealing with very large datasets.
- **Logistic Model Tree (LMT):** LMT is an ML algorithm that combines DT with LRG. It offers a unique approach to classification problems by providing interpretable models with a good balance between accuracy and transparency. However, LMT lacks of scalability when dealing with large datasets with a risk of overfitting the training data. LMT was used in [77] in the context of PF attack detection where it provided 99.21% accuracy and 99.99% for each of the precision, recall, and F1 score metrics.
- **Stacking Ensemble (SE):** Stacking ensemble involves training multiple individual models and then combining their predictions using another model called a meta-model or a blender. SE not only combines the predictions of the base models but also learns how to best weight or combine them based on the training data. The SE follows a two-step process. In the first step, the base models are trained on the training data. Then, in the second step, the predictions of the base models are used as features to train the meta-model, which makes the final predictions. The meta-model learns to find the optimal combination of the base models' predictions to make the best overall prediction. SE has been tested to detect PF attacks by combining different classifiers (LR, KNN, SVM, RF and DT) in [33,71,78]. In those works, SE shows its effectiveness by outperforming standard ML classifiers. However, as one may guess, its time complexity is significantly higher than the other ML techniques.

Regarding supervised DL algorithms used in the surveyed literature, we recognize the following models:

- **Deep Neural Networks (DNN):** DNN is one of the most popular model used in DL. DNNs are designed to mimic the structure and function of the human brain, specifically the interconnected network of neurons. DNNs are composed of multiple layers of artificial neurons. These layers are organized in a hierarchical manner, with an input layer, one or more hidden layers, and an output layer. The hidden layers between the input and output layers enable the network to learn complex representations and patterns from the input data. DNNs are especially effective in domains with large amounts of data. However, training DNNs can be computationally intensive and requires a large amount of labeled training data. DNN has been considered in [79] to train an adversarial model used to attack and bypass the DL-DNN based IDS named *Kitsune*. The authors showcased that their attack can successfully evade the IDS, reduce its accuracy and also increase its false alarm rate.
- **Deep Belief Networks (DBN):** DBN can be seen as a class of DNN, composed of multiple layers of latent variables with connections between the layers but not between units within each layer. DBN are computationally efficient but needs huge volume of data to train and can be used for temporal data analysis. DBN has been tested in [37] to detect HF attacks with success by combining it with a nature-inspired optimization algorithm, namely Rider Optimization Algorithm (ROA), for path re-selection after the removal of the malicious node.
- **Feedforward Neural Network (FNN):** FNN is a type of Artificial Neural Network (ANN) where the flow of information moves in one direction, from the input layer through one or more hidden layers to the output layer. In a FNN, the connections between the layers do not form cycles which distinguishes it from RNNs. FNN was tested in [73] to detect PF attacks in an SDN-based IoT, where it was marginally outperformed by RF, with the authors mentioning that FNN would perform better if provided with a larger training dataset.
- **Recurrent Neural Network (RNN):** RNNs are commonly used in DL for processing sequential data. They are designed to handle inputs of variable length by introducing connections between nodes in a directed cycle, allowing the network to persist information over time. Unlike FFN, which process input data in a strictly sequential manner, RNNs have an internal memory that enables them to capture dependencies and patterns. This memory enables the network to retain information from previous inputs and leverages it to impact the processing of subsequent inputs. However, RNNs can be computationally expensive to train and are sensitive to the choice of hyperparameters. Additionally, they struggle with capturing long-term

dependencies, as the influence of early inputs tends to diminish over time due to the vanishing gradient problem. One of the most widely used recurrent unit architectures is the Long Short-Term Memory (LSTM) cell, which addresses the vanishing gradient problem that can occur in traditional RNNs. LSTMs have gating mechanisms that control the flow of information within the network, enabling it to selectively remember or forget information over long sequences. Another popular recurrent unit is the Gated Recurrent Unit (GRU), which is similar to LSTM but has a simplified architecture with fewer gates. GRUs are computationally less expensive than LSTMs and often achieve comparable performance on many tasks. The work defined in [67] considered RNN, GRU and LSTM to address PF attacks alongside other classifiers in 3 different feature sets, where in average GRU performed better than LSTM while this latter bettered classical RNN. Interestingly, in all cases those three DL classifiers were outperformed by DT. Moreover, LSTM has been also tested in [80,81] to deal with PF attacks. In [81] the proposed solution performed better on the CICDDoS2019 dataset than RF, NB and LR classifiers.

- **Convolutional Neural Network (CNN):** CNNs are a type of DL model commonly used for image recognition, computer vision tasks, and other applications involving structured grid-like data. CNNs are inspired by the organization and functioning of the visual cortex in the human brain. The main idea behind CNNs is to automatically learn hierarchical representations of data through multiple layers of interconnected nodes, known as neurons or units. These layers typically include convolutional layers, pooling layers, and fully connected layers. However, CNN has not the ability to process temporal information. In the context of this survey, CNN has been tested in [80] and [81] to address PF attacks. The results in [80] report a 94.80% mean accuracy for the N_BaIoT dataset, whereas in [81] it was found less efficient to detect the same attacks.
- **Multi Layer Perceptron (MLP):** MLP is a type of ANN that consists of multiple layers of artificial neurons, or perceptrons. It is one of the simplest and most commonly used neural network architectures. In MLP, each layer is made up of multiple neurons that are interconnected with the neurons in the adjacent layers. MLPs can learn complex non-linear relationships and are mainly used for non-sequential data and lack the ability to capture temporal dependencies, unlike RNNs. MLP has been applied and tested in [65,67] to deal with PF attacks. It presented an average detection performance in both papers when compared to other ML and DL classifiers. Nevertheless, it came out in [67] as the second-best classifiers in classification time performance just behind DT; outperforming classical RNN, GRU, SVM, LSTM and RF. In [82], the authors explored the use of the "Looking-Back" concept on different classifiers including MLP to detect PF attacks. This typically refers to incorporating past information or historical context into the learning process. It involves considering previous data points or events to make predictions or decisions about future outcomes. The results were not promising when using this concept with MLP as the accuracy dropped with each of the five looking-back steps (from 99.14% accuracy on the basic approach to 98.06% using 5 steps).
- **Multiclass Neural Networks (MNNs):** MNN is used in DL to obtain a more sophisticated response (Not binary). MNN has been tested in [71] to detect PF attacks where it provides good results having the second-best accuracy (99.529%) just behind SE (LR, KNN, RF, DT). However, the authors stated that MLP is better suited for systems with a large amount of resources.
- **Fusion Ensemble-based (FE):** Fusion ensemble-based, also known as model averaging or model combination, involves training multiple individual models independently and then combining their predictions using a predefined fusion method. The fusion method can be as simple as averaging the predictions based on the performance of each model. Unlike SE which involves using a meta-learner to combine their predictions, FE combines predictions directly without the use of a meta-learner. SE allows for more complex relationships between models than FE. FE can be used in both classical ML and DL depending on the combined methods. In [83] the authors used this concept in the form of a concatenation of RNN, LSTM and GRU hidden layers features; which ultimately provided very promising detection results (99% for each of the accuracy, precision, recall, and F1 score) and also for attacks classification (97% for each of the accuracy, precision, recall, and 96% for F1 score).
- **Stacking Ensemble** In [84], CNN and LSTM classifiers have been combined to detect LF attacks. The resulting stacking-based model provided 94.38% detection accuracy and 92.95% attack blocking accuracy which, according to the authors, is 60.81% higher than traditional methods.

Regarding supervised FL algorithms, the work in [85] considered the *Iterative Model Averaging based Gated Recurrent Unit protocol* (IMA-GRU) to detect PF attacks where it provided an improvement in download/upload/server response time, uplink queuing delay, bandwidth utilization, and throughput when compared to classical "victim-centric" solutions. It also provided a better accuracy than the latter when tested on UNSW NB-15 dataset (around 97.5%).

2-Unsupervised ML approaches: Unsupervised learning or unsupervised classification can be seen as the use of ML methods to analyze and cluster unseen data into groups. Unsupervised learning techniques, such as clustering and anomaly detection algorithms, can be valuable for DoS detection in IoT networks. Instead of relying on pre-labeled data, these algorithms learn the normal behavior of the system and flag any deviations from that behavior as potential attacks. This can be useful for identifying unknown or evolving attack patterns that may not be captured in labeled training data. Unlike supervised models, unsupervised learning algorithms can handle large amounts of unlabeled data efficiently. This scalability is advantageous when dealing with real-time streaming data or when labeled training data is scarce or expensive to obtain. However, they may induce high false negatives. The choice between supervised and unsupervised approaches depends on the specific requirements, available data, and the nature of the IoT environment being monitored. In the context of our survey, we identified the following algorithms:

- **Robust Random Cut Forests (RRCF):** RRCF is an extension of RCF which is based on the idea of creating an ensemble of randomized binary trees and using the properties of those trees to detect anomalies. The RRCF algorithm enhances the original RCF algorithm by incorporating robust statistics to make it more resilient to the presence of outliers and anomalies. The main

idea behind RRCF is to assign anomaly scores to the data points based on their depth in the binary trees of the forest. RRCF has been tested in [49] to detect PF attack where it yielded promising results achieving a precision of 91% in accurately identifying the top 100 traffic anomalies.

- **K-means** : It is a simple algorithm used for data clustering into K clusters. This algorithm was tested in [50,86] for PF attack detection, with mixed results in the first paper, and promising ones in the second.
- **Complete Autoencoder (CA):** CA is an unsupervised DL model that is commonly used for dimensionality reduction, feature extraction, and data generation. It consists of an encoder network that maps the input data to a lower-dimensional representation, often referred to as the ''latent space'', and a decoder network that reconstructs the original input data from the latent representation. A complete autoencoder includes both the encoder and decoder components, which are typically symmetric in structure. This algorithm is used in [87] for the detection of PF attacks where it averaged a 97% percentage in each of recall, precision, and F1 score metrics, outperforming other unsupervised ML models such as K-means and ANN-SOM.
- **Artificial Neural Network based Self Organizing Map (ANN-SOM):** ANN-SOM is an unsupervised learning class of ANN. ANN-SOM aims to organize and represent high-dimensional data in a lower-dimensional space. It is inspired by the biological processes that occur in the brain. The SOM comprises a grid of nodes or neurons, where each one is associated with a weight vector of the same dimension as the input data. Throughout the training process, the SOM acquires the ability to map the input data onto the grid, while maintaining the topological relationships among the input vectors. Clusters of similar input vectors tend to be represented by adjacent neurons on the SOM grid. This property makes SOMs useful for tasks such as data visualization, clustering, and exploratory data analysis. ANN-SOM has been used as an FL model in [88] to deal with PF attacks in fog-based IoT. Compared to other approaches, this algorithm provided an optimized CPU utilization of the controller along with an improved detection rate (around 99%) and accuracy (around 98.7%).
- **Generative Adversarial Networks (GAN)**: It is an unsupervised DL technique based on two competing neural networks a generator and a discriminator. GANs are specifically designed to generate new data that closely resembles the distribution of given training sets. They have gained significant attention due to their ability to generate realistic and high-quality samples across various domains, such as images, text, and audio. The generator in a GAN takes random noise as input and generates synthetic data samples. The discriminator, on the other hand, acts as a binary classifier that tries to distinguish between real and fake samples. It is trained using both real data samples from the training set and fake samples generated by the generator. The training process of GANs involves a game between the generator and the discriminator. This adversarial training process continues iteratively, with the generator and discriminator improving their respective abilities until an equilibrium is reached. GAN was used in [56] to detect PF attacks where it was compared with Gradient boost and RF. The approaches were tested on two datasets (CAIDA, personal IoT generated dataset). The results show that although GAN was slightly outperformed by the two other supervised classifiers, in terms of precision and recall, it was by far the fastest to make inferences by using only benign data in the training phase.
- **Fuzzy C-means (FCM)**: FCM is a data clustering technique in which a dataset is grouped into N clusters with every data point in the dataset belonging to every cluster to a certain degree. It was tested in [50] to detect PF attacks, using the CICIDS-2017 dataset where it clearly outperforms K-means.

4.3. Validation tools and methods taxonomy

The second taxonomy produced in this study aims at classifying technical and practical aspects used in the validation of the surveyed solutions. This includes the considered validation methodology as well as the hardware and the software resources (Datasets, tools, libraries...) used to validate and evaluate the performances of DoS and DDoS attack detection, identification, or mitigation methods. We believe that this taxonomy can provide a clearer perspective, firstly by facilitating the identification of the current state of art of tools and hardware used in this context, and secondly, by aiding in the recognition of correlation between the type of attacks, existing detection approaches and validation tools. The proposed taxonomy is depicted in Fig. 5. The first level of the taxonomy considers the type of validation used: *empiric validation*, *analytical validation*, or *simulation* methods. Note that some approaches have combined two or even all three of these methodologies together to validate their proposed solutions. The concepts used in each category are detailed next.

4.3.1. Empiric validation methods

This covers all the concepts used in experimentation and testbeds. In our survey, a limited number of papers has conducted empirical validation due to the challenges of achieving scalability and to deploying a testbed solution.

Used hardware: This section describes all the hardware concepts used to empirically validate the surveyed literature's testbed frameworks.

- *IoT-DUT:* The IoT-DUT (Device Under Test) or IoT-DUI (Device Under Investigation) is a crucial component of any testbed. The latter is designed and deployed to support the examination of a variety of IoT devices and sensors, including: mote sensors (e.g. TelosB), smart sensors (e.g. Pir motion infrared, CCS811 Air quality, DS18B20 temperature, MQ2 Gas smoke detector, TVOC 2, window/door sensor, Wemo motion sensor etc.), smart cameras (e.g. Belkin, TP-Link NC200, Netatmo, Nest, Yi, Samsung, etc.), smart lamps and bulbs (lifx), smart locks, smart watches (e.g. Fitbit, LG), printers, and other smart objects (e.g. Amazon echo dot, Smart plug, Amazon dash button, Nest thermostat, smart meter, WEMO smart crock-pot, Arlo security system, WEMO power switch, British gas hive, doorbell etc.).

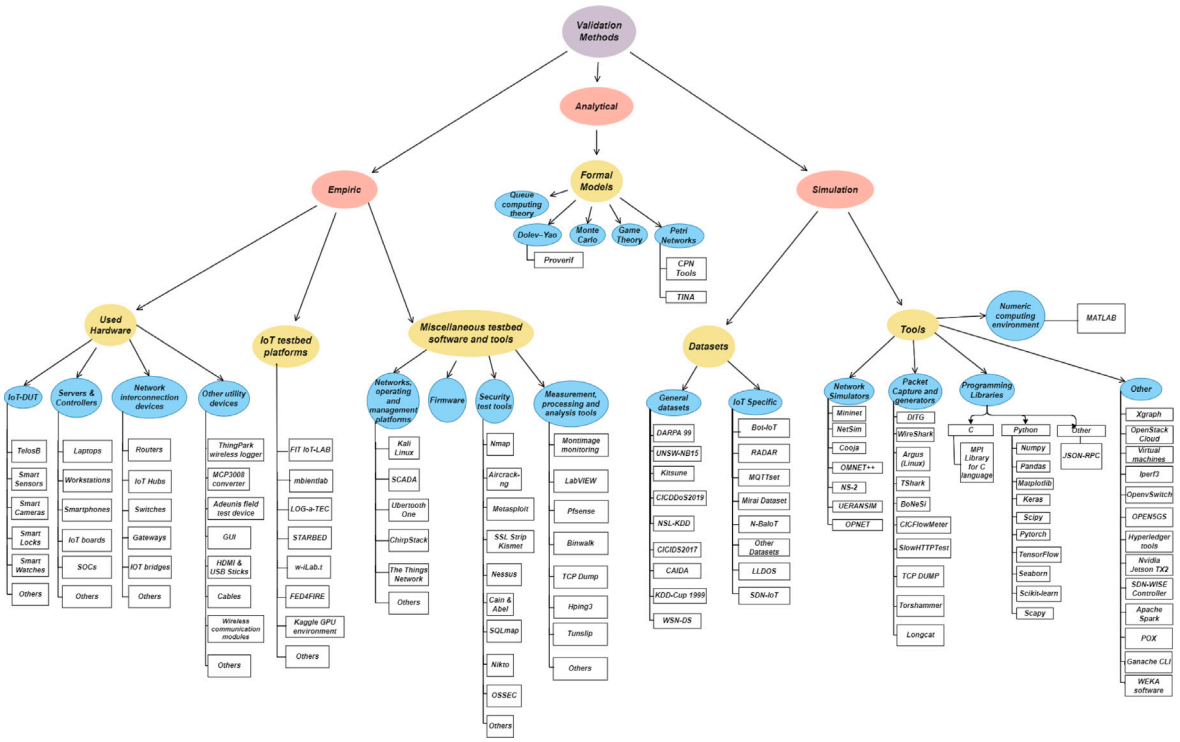


Fig. 5. Validation methods taxonomy.

- *Servers and controllers:* This includes laptops, workstations, smartphones, IoT boards (Arduino Uno, SODAQ Explorer,[1] BLE, Raspberry Pi-3, ESP8266, TI-CC2650, TI-CC2530), or specifically designed SOCs (e.g. nRF51422 BLE SoC, Wemos D1 Mini, Metawear chip). This hardware is used to monitor and control the DUT by running specific tools and embedded firmware.
- *Network interconnection devices:* The testbeds often require the use of communication modules and network equipment like routers (e.g. Netgear DGN 2200 Wifi router), IoT Hubs (e.g. Samsung thing hub, Xbee etc.), switches, gateways (e.g. Kerlin[2]), IoT bridges (Philips hue bridge), etc.
- *Other utility devices:* Other devices can be considered in the testbeds, when needed, like ThingPark wireless logger to record data transmitted from sensors, the MCP3008 to convert analog to digital values, the Adeunis field test Device[3] to test the IoT network (LoRaWan), GUI for visualization, HDMI and USB sticks, cables, wireless communication modules (e.g. Zigbee, LoRawan, WiFi, Bluetooth) etc.

Miscellaneous testbed software and tools: In addition to hardware, different software and tools are needed to run a testbed. In the following, we examine the most commonly utilized ones in the context of the surveyed solutions.

- *Networks, operating and management platforms*: Plenty management and operating systems can be deployed when needed to run a testbed. For example, many solutions run *Kali Linux*[4] which is a Debian-derived Linux distribution designed for digital forensics and penetration testing. It comes with approximately 600 penetration-testing tools. Moreover, the *SCADA server* is often deployed to monitor and control the IoT devices embedded with the controllers. We identify also *Ubertooth One*[5] which is an open source wireless development platform used for Bluetooth experimentation. Some testbeds [64,89] took advantage of *ChirpStack*[6] and *The Things Network*[7] network servers to test their approaches in the context of LoRaWAN.

1 (https://support.sodaq.com/Boards/ExpLoRer/)
2 Wirnet iFemtoCel, https://www.kerlink.com/product/wirnet-ifemtocell.
3 https://www.adeunis.com/
4 https://www.kali.org/
5 https://ubertooth.sourceforge.net/
6 https://www.chirpstack.io/network-server/
7 https://www.thethingsnetwork.org/

- *Security test tools:* In DoS and DDoS related testbeds, designers often utilize several security testing tools available online, including the *Nmap*[8] which is a security scanner tool for the network discovery and security auditing; *Aircrack-ng*[9] which is a complete suite of tools to assess Wi-Fi network security[10]; *Metasploit*[11] which provides reports about security vulnerabilities and assists penetration testing; *SSLStrip* to transparently hijacks HTTP traffic on a network; *Kismet*[12] which is an open source sniffer, wiresless IdS, wardriver, and packet capture tool for Wi-Fi, Bluetooth, and more. Other security and testing tools like *Nessus*,[13] *Cain & Abel*,[14] *SQLmap*,[15] *Nikto*,[16] and *OSSEC*[17] can be considered in testbed design.
- *Measurement, processing and analysis tools*: Security testbeds use a variety of other specific tools, including: data and traffic monitoring and analysis tools, virtual routers and firewalls. For example, the *Montimage monitoring tool*[18] has been exploited to capture and analyze network traffic in [90]. It is utilized to figure out how the protocols, applications and users are operating and detecting potential security and performance incidents. We can quote besides the *LabVIEW*[19] which is a system-design platform and development environment using the visual programming language "G", which is commonly deployed for data acquisition, instrument control, and industrial automation. Some testbeds run *pfsense*,[20] which is a free and open source firewall and router that also features unified threat management, load balancing, and more. Some testbeds used *Binwalk*[21] for reverse engineering, and extracting firmware images. Other popular tools are often used, as for instance: *TCPdump*[22] to analyze data-network packet; *hping3* to send custom ICMP/UDP/TCP packets and to display target replies, and *Tunslip*[23] to bridge IP traffic between a host and another network element over a serial line.
- *Firmware:* is the customized software/code embedded into each device involved in the testbed, and which provides instructions on how the device is supposed to operate during the test. Generally, programming code is composed using one of several languages, such as: Python, Java, C#, JavaScript, REST, and so on.

IoT testbed platforms: While a variety of network testbeds have been designed for specific purposes, only a limited number of them offer a complete solution that addresses all the IoT requirements. The literature review reveals that only a small number of papers have proposed testbeds or considered existing ones to authenticate their methodologies. Actually, IoT testbeds are typically created and utilized with the primary goal of generating datasets that are specific to IoT, or to test the security of IoT devices or platforms. However, they are seldom utilized to validate methods. Notwithstanding the lack of enthusiasm for testbed use in academic researches, many well performing general solutions have gained interest and popularity in the last decade, such as: *Emulab*,[24] *DETERLab*,[25] *Kansei*.[26] For example, we can quote *FIT IoT-LAB*,[27] *Mbientlab*[28] and *LOG-a-TEC*[29] which have been designed to provide testbed systems for sensors networks, and *STARBED*[30] which is a large scale general purpose network simulation environment based in Japan.

However, as the Industrial Internet Consortium approved testbeds must encompass all the significant technologies, domains, and platforms for industrial IoT environments, including the cloud, networks, mobile, sensors, and analytics, more specific solutions have been therefore put forward recently to cope with IoT requirements [91], as for instance: The *T-City Friedrichshafen*[31] which is a multidomain testbed that integrates several IoT devices, enabling the test of many IoT requirements and services; *FIESTA-IoT*[32] which provides an experimental infrastructure for heterogeneous IoT technologies; The *SmartSantander* testbed[33] which proposes a city scale experimental research framework for general smart city applications and services. Other projects are gaining interest as

8 https://nmap.org/
9 https://www.aircrack-ng.org/
10 https://www.aircrack-ng.org/
11 https://www.metasploit.com/
12 https://www.kismetwireless.net/
13 https://www.tenable.com/products/nessus
14 https://cain-and-abel.fr.malavida.com/
15 https://sqlmap.org/
16 https://cirt.net/Nikto2
17 https://www.ossec.net/
18 https://www.montimage.com/
19 https://www.ni.com/en-lb/shop/labview.html
20 https://www.pfsense.org/
21 https://www.kali.org/tools/binwalk/
22 https://www.tcpdump.org/
23 https://github.com/contikios/contiki/blob/master/tools/tunslip6.c
24 https://www.emulab.net/
25 https://deter-project.org/
26 http://kanseiproject.com/en/
27 https://www.iot-lab.info/
28 https://mbientlab.com/
29 https://log-a-tec.eu/
30 https://starbed.nict.go.jp/en/equipment/index.html
31 http://www.telekom.com/dtag/cms/content/dt/en/395380
32 http://fiesta-iot.eu/fiesta-project/project-objectives/
33 https://www.smart-circle.org/portfolios/smartsantander/

the *w-iLab.t* platform[34] which allows its users to run remote experiments in a fully automated way. Also, the *SLICES* project[35] the new version of the *FED4FIRE+*[36] portal which provides a flexible platform designed to support large-scale, experimental research focused on networking protocols, radio technologies, services, data collection, parallel and distributed computing and in particular cloud and edge-based computing architectures and services. Finally, the *Kaggle GPU environment*[37] provides a GPU environment for data scientists and ML practitioners to run code that requires accelerated computing. The GPU environment is primarily used for training DL models and performing computationally intensive tasks.

Regarding the surveyed literature, some interesting works have evaluated the efficiency of their approaches using testbeds [31,43,49,56,75,77,89,92,93]. For example, in [92], the authors validate a DA attack mitigation solution on LoRaWAN by implementing their own testbed using different network server implementations. The testbed comprises an Adeunis field test device used as a victim, a LoRaWAN gateway to receive and transmit packets and an Arduino with LoRaWAN EU module registered to a local instance of the network server. Continuing with the topic of LoRaWAN, the authors in [89] designed a testbed to validate a JA attack mitigation solution. The setup for this solution involves two SODAQ ExpLoRer boards, one acting as the attacker an the other as a victim and an indoor LoRaWAN gateway connected via Ethernet to a basic home router, which then connects to a ThingPark network server. In [49], the authors implement a testbed to generate real stream data in the context of a PF detection approach for IoT smart home. The testbed includes 5 IoT devices which are: a TP-Link plug, a Netatmo camera, a Samsung camera, a WEMO power switch, and a WEMO motion sensor. Likewise, a physical IoT testbed is deployed in [75] to collect real IoT traffic using different temperature sensors interfaced with two data capturers (Raspberry Pi-3 and ESP8266). Two WiFi routers were integrated into the platform to connect the latter. Legitimate network traffic was collected using the TCPdump tool, while the HPING3 and Wireshark tools were used to generate the PF attack. In [43], a testbed is set up to validate a SF IdS based detection solution. The IdS node is placed near a network of heterogeneous real-world IoT devices, including a small WSN of 6 TelosB nodes, a Nest thermostat, an August smart Lock, a Lifx smart light bulb, an Arlo security system, and an Amazon dash button. Furthermore, to deal with BX attacks, a detection solution is validated by implementing a testbed in [31] consisting of two Linux workstations, a BLE board as a provider and an IoT border router connecting two separate IPv6 networks. In [56], the authors design a testbed to evaluate their detection approach based on GAN to detect PF attack. The testbed includes Wi-Fi based access point routers and webcam digital video recorders systems, a Netgear DGN 2200 Wi-Fi router and IP Cameras. Two virtual machines are run to generate the PF attack. *Pfsense* is setup to manage the testbed networking and to capture data. This benign traffic is then used to train the GAN. In [94] the authors took advantage of the *StarBed* testbed to validate their remediation approach against PF attacks using the Mirai dataset.

4.3.2. Analytical validation methods

Such a category inventories concepts used in formally validated approaches, laying on analytical modeling and verification. Concepts are arranged according to the formal framework they are considering. Additionally, for each of the latter we list the resolver used to realize the analytical validation. In overall, five models have been identified:

- *Queue computing theory:* Queue computing, also known as queue-based computing or queue-oriented computing, is a theoretical model of computation that focuses on the concept of queues as a fundamental data structure. It involves the organization and manipulation of data elements in a queue-like fashion to perform computational tasks. For example in [51], the authors took advantage of this model to validate their multi-flow queues counter-based approach for PF attack detection as they performed the theoretical analysis of this approach using this model. The results of this analysis were confirmed using simulation.
- *Dolev-Yao formal model:* This model [95], provides a framework for cryptographic protocols' proprieties formal verification and validation. It was exploited in [31] to verify its newly proposed DoS-resilient authentication protocol against BX attack using *Proverif* [96] that implements this model.
- *Monte Carlo methods:* Monte Carlo methods are "a collection of computational techniques for the solution of mathematical problems". Works in the literature used *Monte Carlo* methods to mathematically prove and validate their DoS and DDoS attacks proposed countermeasures [85,97] against respectively, jamming and PF attacks. Models are generally implemented using personal resolvers or *Matlab*.[38]
- *Game theory models: Game theory* models were considered in researches as a method to validate the approach's adaptability to JA that rely on dynamic attack strategies [98,98–100]. The competition between a legitimate node and the reactive jammer is formulated as a hierarchical game where the legitimate node tries to reduce the probability of its detection by the reactive jammer. The game equilibrium is therefore formally evaluated and analyzed and it was concluded that the legitimate user can improve its profit (i.e communication security) by exploiting the first mover advantage.
- *Petri Nets models:* Petri Nets is a mathematical modeling language widely used for formal validation of event-based processes. In our context, [47,101] used this validation approach to verify their proposed solutions against Gr attack and for the exploitation of specific LoRa vulnerabilities to induce DoS, respectively. The used tools in each of those two researches were respectively, the TINA software [102] and the CPN Tools [103].

[34] https://doc.ilabt.imec.be/ilabt/
[35] https://slices-ri.eu/
[36] https://www.fed4fire.eu/
[37] https://www.kaggle.com/
[38] https://www.mathworks.com/products/matlab.html

4.3.3. Simulation validation methods

Simulation based validation methods refer to concepts used in DoS and DDoS attack handling approaches validated using simulation tools. It can also refer to datasets or libraries that have been considered in the validation.

- **Datasets:** This part of the taxonomy inventories all the datasets used in simulating DoS and DDoS attack detection and mitigation solutions.
  -*General datasets:* This refers to datasets that were generated for classical networks architecture but still suitable for IoT DoS and DDoS attack scenarios. In the surveyed papers, we have noticed the usage of the following well know common datasets: *UNSW-NB15* used in [83,85,104]; *Kitsune* in [78,80]; *DARPA 99* in [60]; *DARPA 2009* in [88]; *CICDDoS2019* in [33,81]; *NSL-KDD* in [66,69,88]; *CICIDS2017* in [50,72,83]; *CAIDA2019* in [56,62,88]; *KDD-Cup-1999* and *WSN-DS* in [83].
  -*IoT Specific datasets:* Datasets that are specifically generated for IoT scenarios have been employed in many solutions: The *BoT-IoT* is the most used dataset in simulating the surveyed solutions [40,41,57,65,67,82,104].
  Other datasets that are not widely used have been considered, such as: *LLDOS* in [62]; *RADAR* in [44]; *MQTTset* in [76]; *Mirai dataset* in [56,79], *N-BaIoT* in [80], *SDN-IoT* in [83]. Some works have fashioned personalized IoT oriented datasets to validate their solutions, as in [49,56,64,67,70,71,73–75,77,81,86,87,94].
  Note that almost all the identified datasets were employed in the context of flooding attacks, except for *RADAR* that was created and exploited in [44] to deal with multiple DoS attacks (BH, SF, SH, HF, CN, SY, WH, VN, RK, RY, WP, DIS, and LR) in the context of RPL. Moreover, we identified three works [37,61,84] that dealt with SH, LF and HF respectively, which, despite using datasets in the context of ML, did not mention any information about the nature of data used in the training process.
- **Simulation tools:** This class identifies all the tools and software resources used in simulation-based validation of the surveyed papers. We further distinguish the tools according to their nature and their functionalities:
  -*Network simulators:* A multitude of network simulators were used in the surveyed solutions. However, the *Cooja* simulator appears to be the most popular, mainly because it offers cross-level simulations (Networking, OS, and machine code levels) in addition to its ability to be adapted for simulating heterogeneous networks, which is an important feature of the IoT domain [105]. *Cooja* was used in different approaches [26,34,42,52,53,55,61,62,106–109] to simulate solutions against almost all the DoS attacks (BX, GR, SH, PF, RY, RK, LR, NGH, DIS, FBA, LRDDoS and SF). *Mininet*[39] is another tool that is gaining popularity mainly in the context of simulating SDN-based approaches. It has been considered in [57,63,72,73,88] to validate solutions against PF attack. Traditional network simulators were also considered in many researches. For example, *OMNET++ simulator*[40] is run in [54] and [38] to address SY and SF attacks, respectively. Moreover, *NS-2*[41] has been employed in [27,45,46] to validate approaches against PF, SH and LRDDoS attacks, respectively. In addition, *OPNET*[42] is used in [85] to deal with PF attacks, whereas *NetSim*[43] is exploited in [44] to deal with multiple DoS attacks. Other specific network simulators were run, as for instance, *UERANSIM*[44] for 5G networks in [78] to address PF attacks.
  -*Packet-capture and generator tools*: Traffic tools are used in different simulation works across the surveyed literature mainly to analyze traffic parameters (nature, origin, and destination...) and to generate customized datasets or attacks used in simulations for approaches' evaluation. We identified the following tools used in the validation of the surveyed solutions. Among them, *Wireshark*[45] is the most used tool, as it has been considered in [60,70,75,77,81,86] to address PF attacks. Other tools like *Argus*,[46] *TCP DUMP*[47] and *Tshark*[48] have been respectively, used in [71,74,75] in the context of PF attacks. *D-ITG*[49] which is a platform capable to produce traffic at packet level accurately replicating appropriate stochastic processes, is run in [73] to simulate IoT devices traffic in the evaluation part of a PF attack detection approach. *BoNeSi*[50] is another network traffic generator for different protocol types. BoNeSi allows the generation of ICMP, UDP, TCP and HTTP flooding attacks. *BoNeSi* has been utilized in [77,81,87,88] to simulate PF attack traffic coming from a botnet in their ML-based PF attack detection approaches. During the dataset creation part of the solution presented in [81], the *SlowHTTPTest*[51] is used to generate a kind of PF attack traffic. This latter was then captured and sent to another tool called *CICFlowMeter*[52] for feature extraction. Moreover, *TorsHammer*[53] is used in [62] to generate LRDDoS attack on web servers.

[39] http://mininet.org/
[40] https://omnetpp.org/
[41] https://ns2simulator.com/
[42] https://opnetprojects.com/opnet-network-simulator/
[43] https://netsim.boson.com/
[44] https://github.com/aligungr/UERANSIM
[45] https://www.wireshark.or
[46] https://openargus.org/
[47] https://www.tcpdump.org/
[48] https://tshark.dev/
[49] https://traffic.comics.unina.it/software/ITG/
[50] https://github.com/Markus-Go/bonesi
[51] https://github.com/shekyan/slowhttptest
[52] https://github.com/ahlashkari/CICFlowMeter
[53] https://github.com/Seabreg/Torshammer

-*Numeric computing environment:* In the literature, the only used numeric computing environment in papers covered by this study is *MATLAB*.[54] The latter is exploited in [29] as a main tool to validate BH and CN attacks detection solutions, while it has been used with other tools in [31,35,45,56] to simulate solutions against respectively, BX, SY, LRDDoS and PF attacks.

-*Programming libraries:* In the surveyed solutions, many works used Python programming libraries mainly in the context of ML approaches. For example, we note the usage of *Pandas*[55] library in [65,69,80,82], *Numpy*[56] library in [65,69,80] for data manipulation and analysis, and *Scikit-learn*[57] library for ML utilities in [65,67,69,80,82,83]. Also, the solutions presented in [67,79] used the *PyTorch*[58] library to take advantage of DL resources. Other libraries like *Tensorflow*[59] were considered in [56,80,83] for ML functionalities. *Scipy*[60] library was used in [80] for mathematical functions integration. Moreover, *Matplotlib*,[61] and *Seaborn*[62] libraries were exploited in [65] to provide drawing tools and interactive visualizations, whereas *Keras*[63] is run in [65,82,83,87] to provide a Python interface for *TensorFlow*. The *Scapy*[64] library which is a powerful interactive packet manipulation library written in Python has been used in [63,73]. The solution in [110] used the JSON-RPC to invoke methods and functions on a remote server over a network.

As concerns *C language libraries*, we identify the *MPI Library*[65] that has been used in [36] to simulate communication and message transfer between the network nodes in the context of CN attack detection. It is noteworthy that some works did mention the utilization of C language [75] or Python [40,75] in their approaches validation process, but did not state which libraries they ran.

-*Other tools:* Different works took advantage of a set of specific simulation tools to validate their solutions. For example, *Virtual Machines* (VM) were run in [58] and [81] in the context of PF attack mitigation. In [58] they were used within the framework of a cloud-based IoT simulation wherein multiple malicious VMs were run to experiment data center's performance in the presence of an attack. The solution in [58] operated in addition the *OpenStack Cloud*[66] to manage and monitor the VMs used in the proposed PF mitigation approach's evaluation. In [81], VMs were used to create a scenario where a malicious VM attacks a victim VM to collect traffic needed for dataset generation. Moreover, *Iperf3*[67] is considered in [72] for bandwidth measurement in the evaluation part of the proposed PF attack mitigation approach. In [46], the authors took advantage of *Xgraph*[68] as a performance analysis tool for a PF attack detection solution. *Nvidia Jetson TX2*[69] which is an AI computing device is used in [80] to train the proposed approach against PF attacks. For the same attack, two works based on SDN [52,53] have taken advantage of the *SDN-WISE*[70] solution for network management. Furthermore, the Apache Spark[71] which is an open-source unified analytics engine for large-scale data processing is run in [87] to validate an ML based solution for PF detection.

Regarding blockchain-based approaches, the solution in [110] used the following tools to detect and mitigate PF attack: the *Ganache CLI*[72] is used to emulate a local Ethereum blockchain network for development and testing purposes, while the *Remix IDE*[73] which is an integrated development environment for Ethereum smart contract development is used to design, test and deploy the application. For the same purpose, the *Hyperledger tools*[74] which is a modular and extensible architecture for developing enterprise-grade blockchain solutions, are used in [33,39] to deal with PF attacks.

The *WEKA* software[75] which provides an ML framework for data preparation, classification, regression, and clustering, is utilized in [77] to train and test the proposed LMT-based solution. In other regards, in [73] the authors used *POX*[76] and *Open vSwitch*[77] to validate their SDN-based solution for PF detection. *POX* which is an OpenFlow controller, can be used as an OpenFlow switch, or to write networking software in general, whereas *OpenvSwitch* is a multilayer virtual switch. *OpenvSwitch* is also run in [88] to implement PF attack detection agents within a FL-based approach. Finally, the *Open5GS*,[78] which is an

54 https://www.mathworks.com/products/matlab.html
55 https://pandas.pydata.org/
56 https://numpy.org/
57 https://scikit-learn.org
58 https://pytorch.org/
59 https://www.tensorflow.org/
60 https://scipy.org/
61 https://matplotlib.org/
62 https://seaborn.pydata.org/
63 https://keras.io/
64 https://scapy.net
65 https://www.open-mpi.org/
66 https://www.openstack.org/
67 https://iperf.fr/
68 https://www.xgraph.org/
69 https://developer.nvidia.com/embedded/jetson-tx2
70 https://sdnwiselab.github.io/
71 https://spark.apache.org/
72 https://docs.nethereum.com/en/latest/ethereum-and-clients/ganache-cli/
73 https://remix.ethereum.org
74 https://www.hyperledger.org/
75 https://www.cs.waikato.ac.nz/~ml/weka/
76 https://github.com/noxrepo/pox
77 https://www.openvswitch.org/
78 https://open5gs.org/

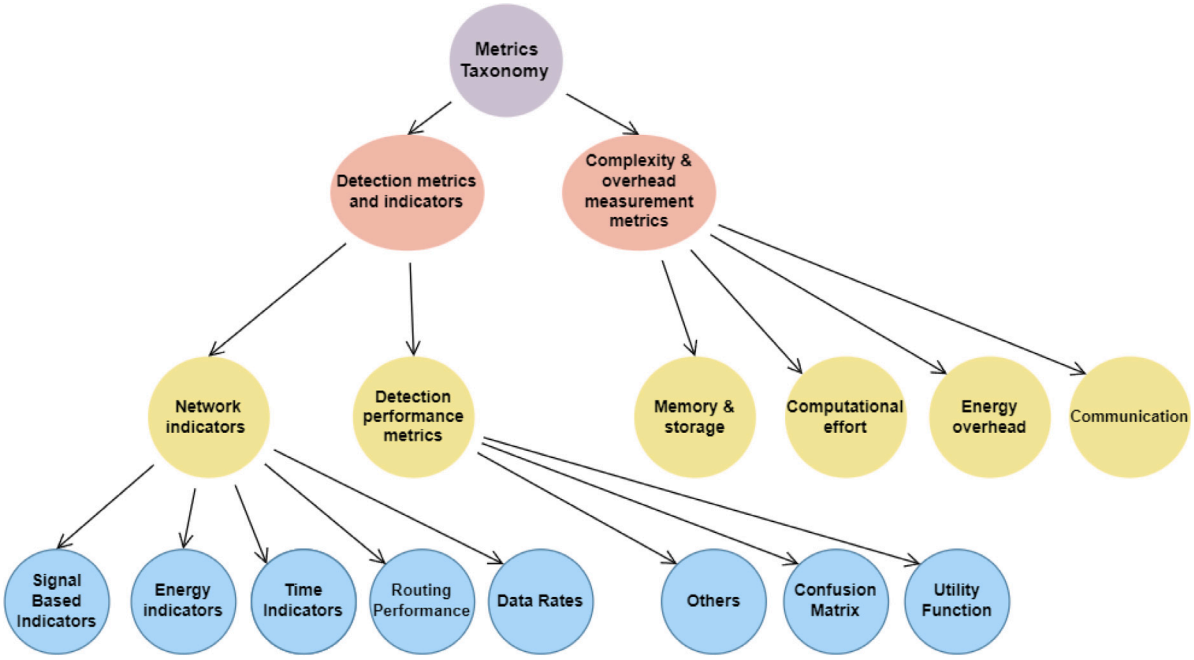


Fig. 6. Taxonomy of evaluation metrics used in DoS attacks detection and mitigation approaches [21].

open implementation of 5G core networks, is considered in [78] to simulate 5G-based IoT networks in the context of a PF mitigation approach.

4.4. Evaluation metrics classification

In a previous work [21], we produced a taxonomy that identifies and classifies all the evaluation metrics and parameters used in the validation process of DoS and DDoS attack detection, identification and mitigation approaches. The primary objective of this study was to correlate the classes of metrics with the different types of attacks and validation methods. This taxonomy is recalled in Fig. 6 to showcase the different classes of metrics that have been recognized. Metrics were categorized into two main classes according to the intended goal of their usage in the validation process. The first category, *detection metrics and indicators*, contains all the metrics used to assess the effectiveness of the proposed approach; which can further be distinguished into two classes: *Detection performance metrics (DPM)* that are used to measure the approach performance in proportion to the instances number of the attack; and *network indicators* that are used to evaluate the network performances as indicators of a potential occurrence of an attack. As regards *complexity & overhead measurement metrics*, they refer to variables used to evaluate the overhead induced by the proposed approaches, which we divided into 5 classes: *memory & storage* that assess an approach's requirement in terms of devices RAM and ROM capacity, *computational overhead* that regroups metrics that measure an approach impact on the IoT object CPU, *Energy overhead* that covers metrics used to evaluate an approach's energy consumption, and finally, *communication metrics* which measure the overhead induced by the approach in terms of network resources consumption.

Subsequently, we take advantage of this taxonomy to investigate the relationship that stands between DoS and DDoS attacks, classes of evaluation metrics, and the detection approaches, as well as the validation methods and tools used in the surveyed literature. All the three taxonomies are correlated for the purpose of answering some research questions that are raised and discussed in Section 6. Before, we proceed to a technical comparative analysis of the surveyed solutions.

5. Comparative analysis of the detection approaches

To gain a better understanding of the detection approaches reviewed in this study, Tables 3 and 4 present a comparative analysis based on the targeted attack, the used technology in the detection scenario, the main algorithm(s) / concept(s) on which the approach is based, the strength and limitation of each approach as well as the main reported results. Table 3 is dedicated to papers addressing packet flooding attacks (PF and ALPF) as they are the most discussed attacks in the literature, while Table 4 is for the rest of the solutions.

Regarding packet flooding attacks, some interesting patterns can be identified: First, in what concerns environment technology, we can notice that blockchain is used alongside fog computing in most of the works, this may be due to the high energy requirement engendered by blockchain calculation that need to be done on devices with higher resources than most IoT objects, in addition to

Table 3
Comparative analysis of detection approaches against packet flooding attacks.

Paper	At-tack	Environment technology	Tested algorithms	Strengths	Weaknesses	Main results
[60]	PF	Cloud Fog	CRPS	-Fast detection -Low FPR	-Anomaly detection -General dataset	100% detection
[58]	PF	Cloud, VM	Entropy	-Lightweight and fast	-Strong assumptions -Anomaly detection -Limited simulations results.	-Abnormal entropy when the system is under attack -Attack successfully detected
[57]	PF	SDN-IoT 5G	Entropy	-Lightweight and fast	-High FPR -Anomaly detection	-100% DR -20% FPR
[59]	PF	SDN-IoT	Entropy	-Dynamic threshold calibration to reduce false alarms -Low overhead on SDN controller	-Tested using only one POX controller	-98.2% DR with tolerance factor = 2
[63]	PF	SDN-IoT	Cosine similarity	-Reduces the communication overhead	-Parameters are manually set up -Anomaly detection	-12% overhead throughput -<20 s latency.
[48]	PF	5G	MARKOV chains	-Proactive approach -High detection rate -Fast detection	-Low DR when having few activity logs -High FPR -Anomaly detection.	-98% DR -(10%–40%) error rate
[52]	PF	SDN-IoT	Counter based	-Extensive simulation -Easy to install and parameter	-High detection time -High CPU usage	-12% CPU ↑ -Detection time ≤150 s
[53]	PF	SDN-IoT	Counter Based	-Low detection time and memory usage	-High CPU usage -Anomaly detection.	- 98% to 100% DR - 50%CPU usage - 2.5 s detection time
[51]	PF ALPF	Cloud, Edge	Counter based Queue computing	-Lightweight -Analytical validation	-Manually defined thresholds -Limited simulation results	-Identification of more than 100 flows using 7 queues
[38]	ALPF	Fog, MQTT	Signature Counter based Queue computing	-Lightweight.	-Limited simulation results -Anomaly detection	-85% of flows are satisfied -20% of traffic is dropped
[39]	PF	Cloud, Fog Blockchain Edge	Signature	-Improved performances compared to other blockchain schemes	-High CPU usage -Scalability issues -Anomaly detection	13% throughput ↑ 11 ms latency 34% CPU usage
[46]	PF	General-IoT	Profile based Signature	-Lightweight	-Not flexible -Limited simulation results -Anomaly detection	Bandwidth consumption ↓ PDR ↓
[33]	PF	Smart-City Blockchain	PUF, Signature, ML:SE(DT,LR) (DT,RF,SVM,LR) SE(DT,RF,LR,SVM) SE(SVM,LR) SE(DT,SVM)	-Strong authentication scheme -Multiple ML approaches comparison	-General dataset -Overhead is not evaluated - Scalability issues -Anomaly detection	SE(DT,RF,LR,SVM) -97.39% accuracy -98.53% detection -07.46% FPR
[40]	PF	Fog, Cloud Blockchain	ML: (RF, XGBoost)	-Distributed architecture -Fast training time -Efficient feature selection	-Energy consuming -Complex deployment -Scalability issues	RF: 99.99% detection rate
[70]	PF	General-IoT	ML: (LR, DT, SVM, KNN)	-Multiple ML approaches comparison	-Limited simulation results -Overhead is not evaluated	DT, LR: 99.99% detection rate

(*continued on next page*)

Table 3 (*continued*).

[41]	PF	Fog Blockchain	ML: (RF, XGBoost)	-Binary and Multiclass classification -Distributed detection	-Complex deployment - Scalability issues.	-RF is better in binary detection -XGBoost is better in Multiclass.
[78]	PF	5G	ML:(SE(DT, LR, KNN), DT, LR, KNN)	-Multiple ML approaches comparison -Efficient feature selection. -Reduced training times -High detection.	-High detection time with SE.	SE(DT,LR,KNN): 97% accuracy and precision
[72]	PF	SDN-IoT Edge	ML:(XGBoost, RF, GBDT)	-Low detection latency -Multiple ML approaches comparison.	-Accuracy ↓ with ↑ attack rates	XGBoost: 94% accuracy 93% precision and recall, 0.2% FPR
[69]	PF	General IoT	ML:(XGBoost, KNN, AdaBoost, DT, NB, RF) DL:(MLP)	-Multiple ML approaches comparison -Uses a data mining model -High detection	-General dataset	XGBoost: 99.58% accuracy 0.99 F1-score
[66]	PF	WSN	ML:(XGBoost, DT, RF, NB,)	-Improved feature selection (FCBFS) -High detection -Multiple ML approaches comparison	-General dataset -Overhead is not evaluated	XGBoost: >99.8% accuracy, precision, recall, and F1-score
[68]	PF	General-IoT	ML: (DT, SVM, NB, RF) Feature selection: (AUC, TOPSIS, Entropy)	-Improved feature selection -Multiple ML approaches comparison -High detection	-Overhead is not evaluated	DT and RF: >97% on accuracy and precision
[71]	PF	General-IoT	ML:(SE(DT,RF,KNN), KNN,DT,RF,NB) DL: (MNN)	-Empirically generated dataset -Multiple ML approaches comparison -High detection	-Overhead is not evaluated	SE:(DT,RF,KNN) 99.611% accuracy
[49]	PF	Smart Home	ML:(DT, RF, SVM, XGBOOST) Graph-based	-Multiple ML approaches comparison -High detection rate	-Graph model lacks of flexibility. -Anomaly detection.	DT: 91% F1 score RF: 98% precision
[50]	PF	IIoT	Unsupervised ML: (FCM, K-means) Graph-based	-Multiple ML approaches comparison -High detection rate	-Overhead is not evaluated -General dataset -High false negative	FCM: 100% accuracy 30% false negatives 1.05% false positives
[87]	PF	ISP	Unsupervised ML: (CA)	-Dynamic feature selection -Fast detection.	-Unknown dataset -Overhead is not evaluated -Anomaly detection	97% average recall, precision and F1 score
[86]	PF	Zigbee	Unsupervised ML: (K-means)	-Empirically generated dataset -Fast detection	-Anomaly detection -Very high FPR -High energy consumption	99.94% accuracy 63% FP 3.7% TN
[77]	PF	Smart-home	ML:(LMT)	-Empirically generated dataset -High performances	-Overhead is not evaluated	99.99% accuracy, TPR, precision, Recall, F1 score
[76]	ALPF	MQTT	ML:(ADABOOST)	-High detection -Deals with MQTT	-Overhead is not evaluated -No comparison is conducted	>95.7% accuracy and F1 score 98.29% precision 93.28% recall
[65]	PF	General-IoT	ML:(DT, RF, KNN, LR, SVM, MLP)	-Multiple ML approaches comparison -Binary and multiclass detection	-General dataset	-RF, DT: >99% accuracy and F1-score
[82]	PF	General-IoT	ML:(DT, RF, KNN) DL: (MLP, LSTM)	-High detection -Uses the looking back concept	-Overhead is not evaluated	RF: 99.81% accuracy

(*continued on next page*)

Table 3 (*continued*).

[67]	PF ALPF	General IoT	ML: (SVM,DT,RF) DL:(RNN, LSTM, GRU, MLP)	-Multiple ML approaches comparison -Binary and multiclass detection -Different feature selections tested	-Overhead is not evaluated	DT: 99.9% accuracy
[75]	PF ALPF	WSN	ML:(ADE, NB) DL:(MLP, RNN)	-Empirically generated dataset -Tested on two datasets -Different feature selections -Multiple ML approaches comparison	-Overhead is not evaluated	ADE: 99.9% accuracy, precision, recall and F1 Score
[74]	PF	General-IoT	ML:(SVM, NB, KNN) DL:(MLP)	-Empirically generated dataset -Predictive approach -Multiple ML approaches comparison	-Anomaly detection -High FPR	SVM: -71% of attacks are predicted -94% accuracy
[104]	PF	General-IoT	DL:(LSTM)	-Trained on two datasets -Binary and Multiclass detection.	-Overhead is not evaluated -Confusion in multiclassification	-96.3% accuracy
[80]	PF	Cloud	Distributed DL: CNN on IoT device LSTM on Cloud	-Distributed detection -High detection accuracy -Extensive comparison with the literature.	-Long training phase -LSTM overhead is not provided.	94.8% accuracy
[81]	PF	General-IoT	DL:(CNN, LSTM)	-Trained on a combined dataset -High detection .	-High training time -High CPU usage -No feature selection	-99.9% accuracy -99.3% precision, recall and F1 score
[56]	PF	General-IoT	Unsupervised DL: (GAN) ML: (RF, Gradient boost)	-Trained on combined datasets -High recall -Low latency	-Overhead is not evaluated -Anomaly detection	GAN: 100% precision 93%recall 6.05ms latency
[73]	PF	IoT-SDN Edge	DL:FNN ML:(SVM,RF)	-High detection -Low FPR	-Unknown dataset Overhead is not evaluated	SVM: 100% precision FNN: 99.93% TPR <0.02% FPR
[83]	PF	CPS SDN-IoT	DL: (FE(RNN,LSTM,GRU), LSTM,GRU,RNN) ML:(NB,LR,KNN, DT)	-High detection -Tested on multiple datasets -Multiple ML approaches comparison	-Overhead is not evaluated	FE(RNN,LSTM,GRU): >0.97% accuracy, precision and recall >0.96% F1-score
[85]	PF	IIoT, Fog	FL:(IMA-GRU)	-Improved security due to FL -Attack centric approach Analytical validation	-High implementation cost -General dataset	98% accuracy
[88]	PF	HIoT, Cloud, Fog	Unsupervised FL: (ANN-SOM)	-Improved security due to FL -Trained on combined datasets -Uses an orchestrator.	-General datasets -Communication overhead	99.3% detection 99.4% accuracy

the distributed nature of this technology that fits the fog architecture. It is also noticeable that blockchain and SDN were not used together in the literature as the distributed aspect of the former technology conflicts with the centralized, controller-based, working paradigm of the SDN-IoT. On the other hand, SDN solutions and edge computing, which fit well together, are often combined in the literature.

Furthermore, the majority of the approaches relying on classical algorithms can be regarded as "Lightweight", especially for entropy and queue computing-based approaches, while having as a downside limited simulation results as these approaches were not extensively validated in most of the cases. Furthermore, almost all these solutions have been designed for anomaly detection, thus resulting in a high error detection rate.

In what concerns ML-based solutions, we can notice that most of the papers compared multiple ML/DL/FL models and presented detection results in the form of confusion matrix values and percentages (accuracy, precision, recall, F1-score ... etc.), which were mostly very high. A noticeable downside in those papers is, for most of them, the absence of overhead evaluation. A brief analysis of the primary findings indicates that DT, RF and XGboost models presented consistently demonstrated promising results

Table 4
Comparative Analysis of detection approaches against other attacks.

Paper	Attack	Environment technology	Main algorithm	Strength	Weakness	Results
[64]	ADV	General-IoT	ML: (DT,RF,NB,SVM)	-Used their own generated dataset -Improved performance. -No additional overhead.	-Strong assumptions on the dataset and selected features	RF: 99.9% precision, recall and F1-score
[79]	ADV	General-IoT	DL:(DNN)	-Tested on a state-of-the-art NIDS,	- Strong assumptions on feature extraction	94.31% of attack success rate
[54]	SF	WSN	Counter based	-Lightweight and energy-efficient -Low detection latency	-Requires static routing paths	-90% PDR ->95% detection. -<7% FPR.
[43]	SF, PF, HF,WH SH	General-IoT	Rule based	-Low CPU and memory usage -Empiric validation	- FPR and FNR are not evaluated. - Lacks of theory and details	-91% average detection -100% Accuracy -0,19% CPU overhead -14 MB RAM overhead
[61]	SH	RPL	ML:(RF), JSL	-Extensive analysis and evaluation -Combining ML and Not ML techniques. -High detection. -Low latency.	-Unknown dataset	-95% PDR -<3% FNR and FPR. -98% Accuracy
[42]	SH, RK, LR, NGH, DIS	RPL	-Finite State machine. -Profile based	Scalable	-Not flexible.	-Up to 100% TPR -<6.8% FPR -6.3% energy overhead
[27]	SH	RPL 6LoWPAN	Aggregation score Maximum likelihood DSE theory, BPDF Constraints based	-Energy efficient	-Strong assumptions -No detection evaluation	-94% PDR, -Overhead <10%
[26]	SH	RPL 6LoWPAN	Constraints based DSE theory, BPDF Counter based	-Considers nodes mobility	-High packet drop ratio - High FNR - Overhead is not evaluated	-92% detection rate (static nodes) -75% detection rate (mobile nodes) -28% FNR, 8% FPR
[32]	SY	RPL	PUF Bloom filter	-Lightweight -Authentication -Energy efficient -Intensive simulations -Analytical validation	-Requires hardware implementation of the PUF	-95% detection rate -5% FNR -<30 s detection latency
[44]	SF,BH,SH, HF,CN,SY, WH,VN,RK, RY,WP,LR DIS	RPL	ARIMA Signature Rule based	-Generate a dataset considering 12 RPL attack types -Multiclass detection	-Low Detection rate for some attacks -Overhead is not evaluated -Limited simulation results	Detection rate 100%: SF,SH,HF,CN, SY,RK,RY,DIS 80%: BH,WH,VN,WP 40%: LR
[28]	SF, SH VN	RPL	JSL	-Proposed 3 confidence based algorithms -Energy efficient	-Communication overhead	-(80%–95%) detection rates -7% FNR and FPR
[30]	SY	RPL 6LoWPAN	TPM, Aggregation score	-Security computation offloaded to TPM	-Communication overhead -Cost of implementing TPM module	No validation
[35]	SY	General-IoT	Location proof Signature	-Simple yet efficient concept	-No evaluation of detection.	Improved throughput, number of alive nodes, and energy consumption
[29]	BH, CN	WSN	HHT KS test Signature Location proof	-Provides nodes authentication -Scalable	-No evaluation of detection.	95% average PDR under attack

(*continued on next page*)

Table 4 (*continued*).

[36]	CN	General-IoT	Location proof Signature	-Distributed architecture -Considers mobility -Intensive simulations	-High CPU usage -Assumes a symmetric routing pattern	-100% detection rate -Detection time <1 s per node.
[34]	BX	General IoT	Location proof	-Intensive simulations -Lightweight -Formal validation	-Deals only with static nodes	-Less than 12 cm locating error -The false alarm probability of 0.7%
[31]	BX	Restful IoT	Leaky BUCKET EWMA Ticket based	-Multiple detection algorithms -Lightweight -Formal verification	-Manually defined thresholds	-100% detection -<3% energy overhead
[55]	GR	LR-WPAN Wi-Fi	-Mean and standard deviations -Counter based	-Lightweight	-Evaluated with 1 malicious node	99.5% detection
[47]	GR	LR-WPAN Zigbee	PETRI Net	-Formal modeling and verification -Extensive analysis of the attack	-No detection evaluation	–
[62]	LRDDoS	6LoWPAN	GTV	-Distributed approach -Lightweight	-Overhead is not evaluated	-3.47 s detection time -5.41% FNR, -5.12% FPR
[45]	LRDDoS	General-IoT	Rule based	-Scalable -Formal modeling and analysis	-Challenging threshold setup -High FNR -Overhead is not evaluated -Anomaly detection	-FPR < 5% -FNR > 20%
[37]	HF	WSN	DL: (DBN) Location proof Constraints based	-Combine ML and not ML approaches	-Unknown dataset -FPR and FNR are not evaluated	-100% detection -Improved latency energy consumption and path selection
[93]	JA	Zigbee LoRaWAN	Counter based	-Extensive analysis of JA.	-High energy consumption on LoRa	-100% detection rate
[84]	LF	SDN-IoT	DL: SE(CNN,LSTM)	-Specific generated dataset	-Overhead is not evaluated -No results on FPR and FNR -Few details on the dataset	-94.38% accuracy

across the papers that tested those models. Moreover, when the former two models were combined into a stacking ensemble, even more favorable outcomes were observed. However, XGBoost was outperforming other models in multiclass detection. On the other hand, models like KNN, MLP, LSTM, NB, RNN, LR, and SVM consistently under-performed when tested before other classifiers. Interestingly, for the majority of papers comparing classical ML and DL models, the latter performed better overall, with however a lighter impact on IoT device resources. Another observation that can be made is that many papers proposed either new feature selection methods or improved existing ones, both of which can reduce training times and improve the overall accuracy of the generated model.

Regarding solutions listed in Table 4, a global perspective reveals that these solutions are for most of the cases targeting specific IoT environment technologies, such as RPL, 6LoWPAN, WSN, and others, with RPL being the most prevalent. The majority of the solutions addressing these attacks relies on the combination of multiple classical algorithms to facilitate a binary or a multiclass detection of the attacks. As for an example, approaches dealing with ADV attacks improved significantly the performance of ML/DL models while having as a common weakness the requirement of an extended knowledge on the targeted model and the extracted features. Approaches dealing with SF were lightweight but the FPR was either high or not evaluated in some cases. Papers addressing SH were almost all based on RPL, using various Not-ML approaches while reporting promising results overall, with each approach having its specific drawbacks. The same observation can be made for other RPL-specific attacks VN, LR, WP, DIS, RK, and NGH. Moreover, approaches dealing with SY and CN attacks have mostly either been based on location-proof and signature concepts (providing extensive authentication but generating overhead), or proposed hardware solutions such as PUF and TPM (dedicated security but high implementation cost). Approaches dealing with BX attacks have been formally verified and validated. They are lightweight and report good detection performances. Both papers dealing with GR attacks considered an LR-WPAN environment but used contrasting approaches as one provided extensive formal modeling and analysis using Petri net theory, while the second

presented a network analysis-based approach that was validated using simulations. Furthermore, both of the proposed LRDDoS detection approaches were based on network analysis, with one approach having the specificity of being distributed. Unfortunately, the overhead that these two approaches may induce was not evaluated. In the case of HF attack detection, two classical algorithms were combined with a DL model. However, the validation was performed on an unspecified dataset, which raised concerns about the reliability and credibility of the results. A DL stacking ensemble approach was proposed for LF attack detection, in the context of an SDN-based IoT network, for which a dataset was generated. This solution presented high-accuracy results, but the overhead that may be caused by the usage of this approach was not evaluated. Finally, only one approach was proposed for the detection of reactive jamming attacks which was based on performance metrics analysis for each of the Zigbee and LoRaWAN technologies. This approach presented promising detection results but a higher energy consumption was noticed in the context of LoRaWAN.

6. Discussion and research questions

In what follows, we undertake a comparative study to establish correlations between the various classes of our taxonomies. This examination considers several aspects, with the aim of addressing the following research questions:

- **RQ1:** Which classes of our taxonomy are most commonly used to detect DoS and DDoS attacks?
- **RQ2:** Are there correlations between a type of DoS or DDoS attack and the approach used to detect it?
- **RQ3:** What are the most popular tools used in the validation of solutions against DoS and DDoS attacks?
- **RQ4:** Does there exist a relationship between detection approaches and the validation methods, tools, and datasets used to evaluate them?
- **RQ5:** How do detection approaches and evaluation metrics relate?
- **RQ6:** Do validation methods and tools influence the type of evaluation metrics that should be considered during validation?

We believe that the responses to these research questions can provide researchers with the stepping stone into understanding the full picture of the existing solutions in detecting DoS and DDoS attacks in the IoT, and the way they are evaluated and validated.

6.1. RQ1: Detection approaches utilization

First, we compiled statistics on the utilization of every category class and subclass of DoS and DDoS detection approaches in the literature we surveyed to address our first research question. This analysis enables us to gain insights into the frequency and distribution of these approaches in the research landscape.

First of all, regarding the first level of the taxonomy presented in Fig. 4, we found out that ML and Not ML categories are almost equally represented in the papers we gathered, with 31 papers falling under the Non-ML category while 34 are under ML. Note that among the latter, few of them have further applied classical algorithms (e.g. Graph, standard and mean deviations, entropy, rule-based, etc.) in data pre-processing. However, at this level, we take into account only the main algorithm used in the detection to categorize each paper.

Upon examining the Not-ML category, we find out that solutions based on *network analysis* are the most used (23 out of 31 papers). This is because active network monitoring and profiling had already proven their effectiveness in classical networks which might have encouraged researchers to propose simple lightweight approaches based on these concepts to fit IoT constraints. Moreover, *ID authentication and verification* based techniques appear in 11 papers, in which they are often combined with other classical algorithms. Finally, 5 papers took advantage of *confidence value* based algorithms in designing their detection approaches.

Regarding ML-based solutions, *supervised ML* techniques are dominating by a far margin when compared to unsupervised ones (82.35% vs 17.81%), being also the most regarded when considering all the surveyed papers that presented a detection approach (43% presence in overall), followed by respectively *network analysis* solutions (34%), *ID authentication & verification* (17%), and both of*confidence value* based and *unsupervised ML* (9%).

In order to provide a more comprehensive analysis of the figures, we focus now on the most commonly used algorithms within the surveyed literature. *Classical ML* based detection algorithms are the most exploited algorithms in overall (40% of the surveyed papers proposing a detection approach), mainly because they mostly require less computational and storage requirement than DL and FL-based solutions. *Statistical and analysis algorithms* can easily be distinguished as the second top (26%). This shows a tendency towards using statistical analysis of IoT network data in the detection of DoS and DDoS attacks. *Supervised DL* approaches are considered in 13 articles (20%).[79] Then, follow *signature* and *counter based* algorithms which are used in 8 papers each (12.3%), 5 papers took advantage of RSSi-based *location proof* techniques, while *blockchain*, *entropy*, and unsupervised ML solutions are considered in 4 works, each. Finally, *constraint based*, and *rule based* solutions were utilized in three papers, each. The remaining algorithms have been exploited twice or less in the surveyed literature.

[79] Some ML papers have tested and compared conjointly DL and classical ML algorithms.

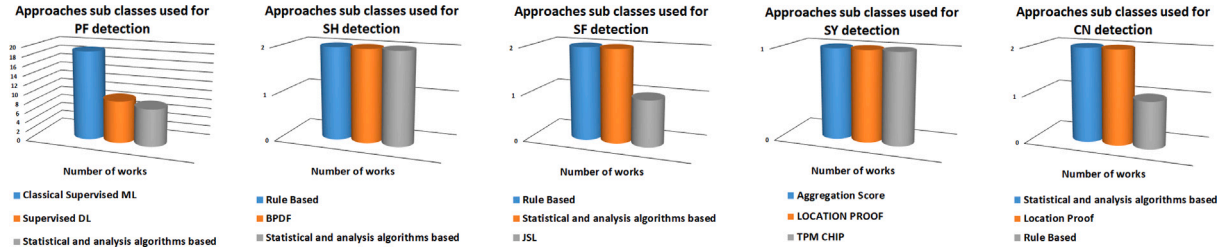


Fig. 7. Most used approaches in most addressed attacks.

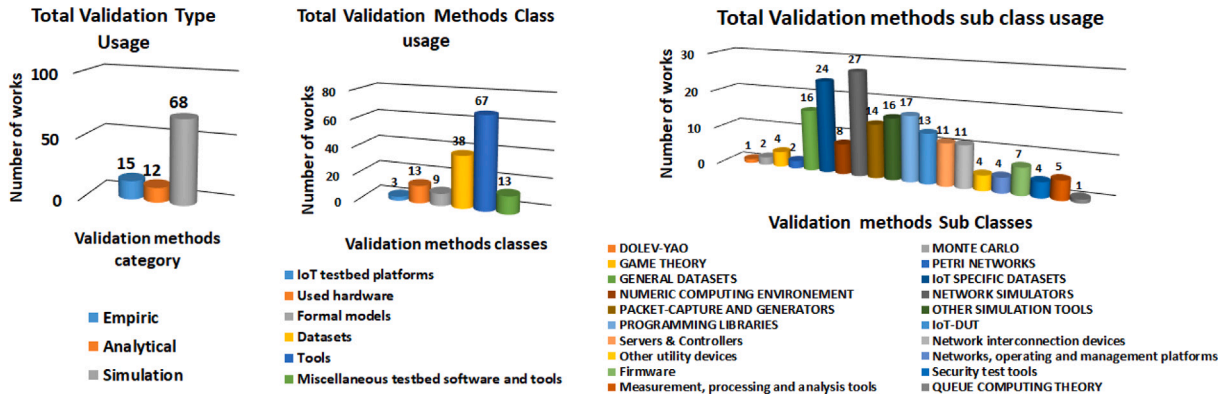


Fig. 8. Validation methods usage.

6.2. RQ2: Correlation between attack types and detection approaches

We extend our analysis to cover each individual detection approach in our classification to categorize the statistics by attack types and IoT layers concerned by those approaches. Starting with the PF attack, as it is by far the most discussed attack in the papers proposing detection approaches (40 papers). We notice that there is a noticeable leaning towards using ML approaches (27 works vs 13 for classical algorithms) to detect this attack. A supposition that may explain this patent is that PF is the most common type of DDoS attacks present in most network types including IoT ones, and also most datasets that ML approaches use are generated and trained upon are mostly, if not completely, composed of different variants of PF attacks (TCP, UDP, and ICMP flooding). It is also clear from the statistics that this is a two-way relationship. That is to say that ML category approaches are almost all focused on PF attacks as 27 of the 34 papers using ML-based algorithms have dealt with it.

In other respects, the SH attack is the second most discussed attack in the papers we gathered (7 papers). In almost all cases, classical algorithms were used (6 papers, implementing mainly *network analysis* and *confidence value* based algorithms). A similar assertion can be made regarding SY and SF attacks (with 4 research works proposing detection methods for each), as well as CN attacks (with 3 papers) for which classical algorithms were applied. We note that *network analysis* based algorithms have been used in all the papers dealing with SF, whereas the most proposed approaches for SY are based on *ID authentication and verification*. Fig. 9 shows the most three used subclasses of detection approaches in dealing with the most addressed attacks in the literature we had surveyed (see Fig. 7).

In overall, *network layer* attacks were the most targeted by papers proposing detection approaches; naturally as most identified attacks occur in this layer (58 papers, 33 of them are implementing ML). Regarding attacks that take place in the *perception layer*, they have been addressed in five papers all implementing classical algorithms (two of which focused on GR [47,55], two on BX attacks [31,34] and one on JA [93]). This could be due to the idea that classical approaches have already shown efficiency in dealing with these attacks, thus dismissing the use of ML techniques in this case. Moreover, ML-based techniques need a pre-deployment phase that consists of model generation and training which itself requires a pre-generated dataset containing instances and traces of this layer's attacks. Something that is challenging to prepare for attacks identified within this layer. In other respects, *application layer* is the least discussed with only 5 papers proposing detection approaches (3 of them use ML).

6.3. RQ3: Validation methods and tools utilization

In a similar manner to the previous research questions, we count hereafter the usage of tools in each of the three types of validation methods (empirical, analytical, and simulation). An overall look at the obtained statistics gives a clear idea of which validation type is preferred by researches dealing with DoS and DDoS attacks in the IoT (see Fig. 8). Simulation-based validation

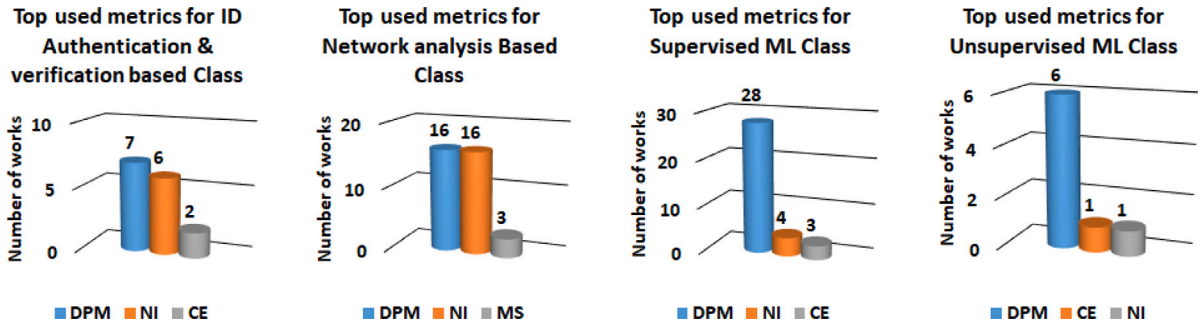


Fig. 9. Most used metrics in each class of detection approaches.

was conducted in more than 86% of the works that presented a validation and evaluation part of their algorithms. (53 works run simulation only, while 15 papers combine simulation with other validation methods). This proportion was expected because simulation-based validation is most of the time easier to implement, cost and time effective, and also gives a better scalability and diversity when compared to empirical-based validation, for example. Empirical validation methods were the second most considered validation type, being performed in 23.4% of the papers (4 papers consider empirical validation only [89,92–94], while 10 papers combine it with simulation [43,49,56,64,71,74,75,77,79,86], and 1 article uses the three validation methods [31]), with a marginal advantage over analytical validation (15. 4% of the works covered by our study).

Digging deeper into the classes of each validation type, as one can expect, *real hardware* and *software and tools* were deployed in all works using empirical validation. Moreover, only two papers have taken advantage of existing testbed platforms. Regarding analytical validation, *game theory* was the most used (4 papers) [45,98–100] followed by *Petri net theory* [47,101], *Montecarlo model* [85,97] and *Queue computing theory* [38,51] taking advantage of personal resolvers, CPN and TiNA tools.

In what concerns works using simulation-based validation methods, all of them took advantage of existing *Tools* to simulate their approaches. Furthermore, we notice that most of the identified datasets were processed in papers proposing ML approaches (34 out of 38 papers using datasets), mainly against PF attacks, while we identified the usage of four datasets in the context of classical approaches. A more detailed look shows that *network simulators*, are the most operated in simulation-based solutions (and also in overall; 27 papers). On the other hand *IoT specific datasets* were the most utilized among datasets (25 papers), whereas *General datasets* have been exploited in 16 works. Moreover, *Other tools* had a noticeable usage rate within simulation-based papers (18 papers).

When correlating between attack types and validation methods, it appears that among the 13 papers dealing with *perception layer* attacks: 5 are analytical [47,97–99,101], 4 used simulation [34,55,106,111], 3 run an empirical validation [89,92,93], and one used all of the three validation methods [31]. *Network simulators* and *numeric computing environment* (MATLAB) were the simulation tools most regarded in this context. The JA attack has the highest number of papers proposing validation methods within the *perception layer* (6 papers, 3 of them are analytical [97–99], two works follow an empirical validation [89,93], and one paper uses simulation [111]).

As for *network layer* attacks, among the 64 papers that validated their approaches (including 7 papers proposing other approaches than detection [94,100,106–110]), 62 works carry out the simulation (11 of them combined with an empirical validation and 4 combined with an analytical one [32,45,51,85]), one paper uses empirical validation [94], and the remaining paper follows an analytical validation [100]. When categorizing by attack class, 49 solutions targeted *flooding attacks*, 47 of them used simulation. The PF attack was the most addressed attack (43 papers), in approaches based mainly on using *datasets* (30 papers) and running *packet capture and generators* (14 papers), *Other tools* (13 papers) and *network simulators* (11 papers). Both articles dealing with LRDDoS attacks have run simulation, using *Network Simulators (NS)*.

Deceptive traffic redirection attacks are discussed in 12 papers, all of which used simulation (one paper combined with analytical, another paper combined with empirical validation) while identifying SH as the most targeted attack within this class (7 works). *Network simulators* and *numeric computing environments* are the most adopted tools to validate this attack class. All 9 papers dealing with *packet-drop* attacks used simulation, with one work that used, in addition, empirical validation [43], while SF is the most discussed attack within this class (7 papers), by considering *network simulators* in the validation process. Finally, ALPF being the only *application layer* attack, has been addressed in 5 papers [38,51,67,75,76], all of them have run a simulation-based validation method (1 paper combined with analytical validation [51] and another combined with empirical [75]) using BoT-IoT *dataset* [67,75] and Scikit-learn *python programming library* [67,76] (2 occurrences each).

6.4. RQ4: Relationship between detection approaches and validation methods and tools

To connect between the two taxonomies proposed in this study, we conducted a comprehensive statistical analysis to examine the usage of each validation method and associated tools across each class and subclass of detection approaches within our taxonomies. This analysis allows us to gain insights into the preferred validation techniques employed within specific categories and subcategories of detection approaches.

An overall analysis shows that the simulation validation methods were used in 97% of the works proposing detection approaches (62 of 64 works); either solely (47 works) or used besides other validation methods (10 works proposed simulation together with empiric validation, 4 combined simulation with analytical [32,45,51,85], and another work used all three [31]). Moreover, empiric validation methods were conducted in 12 papers proposing detection approaches, 11 papers have combined it with either simulation or analytical validation. Finally, analytical validation was considered in 6 works, the work in [47] used it exclusively.

6.4.1. Validation methods and tools usage in not-ML category approaches

Diving into the usage of validation methods and tools within classical approaches, we found out that empirical validation was conducted in 3 works, all of which implement a *network analysis* based approach [31,43,93], while one [31] considers in addition an ID authentication and verification method. Moreover, five papers perform an analytical validation. One considers a *network analysis* and *specification* based Petri Nets approach validated using the *TINA tool* [47], while another used the *Proverif* tool for *Dolev-yao* based formal validation of an approach based on *ID authentication & verification* [31]. Two papers consider *network analysis* based approaches. The one in [45] used a rule-based solution to detect LRDDoS attacks, while the work in [51] combines a counter based with *queue computing theory* to detect PF and ALPF attacks. The last analytical paper [32] conducts an *ID authentication & verification* based approach to detect SY attacks. Simulation-based validation methods were the most operated within the papers adopting classical approaches with 94% usage rate (29 papers). *Network simulators* came at the top (17 works), 14 of them being considered in, as one would expect, *network analysis* based approaches. Moreover, eight works took advantage of *other tools* in validating mainly *network analysis* based approaches. The use of *Programming libraries* was mentioned in one work based on *ID authentication & verification* [36]. Two works ran *packet-capture and generators* tools using a *network analysis* based approach [60,62].

As concerns datasets, they were seldom used, as only two IoT-specific datasets (*RADAR* and *BOT-IoT*) were used respectively in [44] for the detection of 13 routing attacks in RPL using *ARIMA* and [57] for their *entropy* based PF attack detection, while three general datasets (*DARPA99, CAIDA 2007 and CICDDoS2019*) have been exploited once each in the validation of three *network analysis* based approaches [57,60,62]. The work in [57] uses entropy to detect PF in an SDN-IoT/5G environment, while the solution presented in [60] adopted CRPS to detect PF attacks too in a fog/cloud environment. The approach in [62] uses GTV to detect LRDDoS in a 6LoWPAN.

6.4.2. Validation methods and tools usage in ML category approaches

As concerns ML-based detection approaches, all of them were validated by running simulations to program and train their classifier models. However, 9 works have combined simulation with empirical validation mainly to generate datasets: 4 works used *supervised classical ML* [64,74,75,77]; 2 works considered *supervised DL* [71,79]; 1 work implemented *unsupervised classical ML* [86]; 1 work took advantage of *unsupervised DL* [56]; and finally 1 work used both *supervised classical ML* and *unsupervised classical ML* [49]. Analytical validation was conducted once as *Monte Carlo* method was applied in the validation of IMA-GRU supervised FL approach [85].

For simulation-based papers, unlike for classical approaches, datasets were considered in all of the paper's works implementing ML approaches. This is consistent with the fact that training an ML-based detection model always requires pre-collected attack data that is provided in the form of datasets. IoT-specific datasets were considered in most of the works (23 papers), while general datasets were used in 13 papers; 3 of them implement an unsupervised ML [50,56,88] while another work [49] combines both unsupervised and supervised ML algorithms to detect PF attack in a smart home environment. Regarding simulation tools, 13 papers (39.4%) mentioned the usage of *programming libraries* for validating ML approaches. However, it is probable that nearly all of the ML approaches utilized this tool within their programming framework (e.g. Python and C) to implement their classifiers, even if this was not explicitly stated in the papers. The use of *packet-capture and generators tools* appears in 10 works, while *Other tools* are run in 7 papers. *Network simulators* have been considered in 7 ML papers (6 papers using *supervised ML* [61,71–73,78,85]), (1 paper using *unsupervised ML* [88]), where they were used to assess the performances of the generated ML model in a simulated network environment.

6.5. RQ5: Correlation between detection approaches and evaluation metrics

In a similar manner to the previous research question, we present in what follows our findings on the connection between evaluation metrics that we classified in the taxonomy presented in Fig. 6 and that of DoS and DDoS attacks detection approaches given in Fig. 4.

Regarding main metrics categories, it appears that *detection metrics and indicators* were calculated in all the papers proposing detection approaches, while *complexity and overhead measurement metrics* were only used in 13 works (20% usage rate), 8 of them implemented classical approaches and 5 ML solutions. In what concerns metrics classes, within papers proposing classical detection approaches, *Network Indicators* (NI) and *Detection Performance Metrics* (DPM) were utilized equally (21 papers, for each). Interestingly, for papers using ML-based detection approaches, DPM presented a 100% usage rate. This is understandable as the main evaluation measurement for ML techniques is their attack detection performances. While in contrast, we found that NI metrics class usage dropped drastically within this category with 4 papers only computing them [37,61,73,85]. It is also noticeable that ML approaches have evaluated the overhead of their solutions by only computing *computational effort metrics* (measuring CPU time performances).

Moreover, the analysis of metrics subclasses usage reveals that *Confusion Matrix* (*CM*) metrics are highly regarded, with a total of 50 papers utilizing them. Notably, CM metrics are predominantly adopted in ML detection approaches, with a 97% usage rate across 33 papers. This can be explained as most ML algorithms are evaluated using metrics like accuracy, precision, F1-score, and

so on, which are all part of the extended confusion matrix. Only one ML paper [37] that used *supervised DL* (DBN) has evaluated its approach by measuring the normalized energy, the latency and the length of the shortest path between nodes in its approach for the detection and the mitigation of HF attacks.

Furthermore, the *Data Rates* (DR) metrics were the second most computed in overall (20 papers). 19 works proposed a classical detection approach which makes these metrics the most adopted within this category of solutions, followed by the CM metrics (18 papers). Moving next to *Energy indicators* (EI) metrics which were computed in 10 papers, mostly in solutions proposing classical detection approaches. The only two ML papers that used EI in their approach evaluation targeted HF attack in the case of [37] which applied supervised DL (DBN), and SH attack in the case of [61] which implemented supervised classical ML (RF).

Regarding *other metrics*, they were computed in 9 papers, mainly in classical detection approaches dealing with specific environments, like RPL in [32,44]; SDN in [52,53,59]; 5G in [48]; blockchain in [39]; and 6LoWPAN in [62].

Time indicators, *Utility Functions* (UF) and *routing performances* metrics were seldom used (6, 4, and 2 occurrences, respectively). UF was measured in solutions based on *location proof* [34,37], *Entropy* [57,58] and *supervised ML* (DBN) [37]. *Time indicators* were calculated in the context of solutions implementing PUF [32], counter based algorithm [93], and ML approaches considering RF [61], GAN [56], and IMA-GRU [85]. Finally, *routing performances* metrics were measured in the solution presented in [37] combining DL(*DBN*) with *location proof* to detect HF attack, and in [93] using a counter-based algorithm to detect JA in the context of Zigbee/LoRaWAN networks.

6.6. RQ6: Relationship between validation tools and evaluation metrics

Using the same statistical analysis process, we describe in this section the key points that highlight the correlation between evaluation metrics and validation methods and tools that utilized those metrics.

Starting with main metrics categories, it appears that *detection metrics and indicators* were measured in almost all the papers proposing a validation process The *complexity & overhead measurement* metrics were computed in 16 papers (20.2%); all used simulation, while one rule-based solution was considered besides an empirical validation [43] to detect multiple attacks. If we look deeper at the usage of metrics classes, the DPM class was the most computed in overall (61 papers, 77%). It is also the most calculated class within papers using empirical and simulation-based validation (80% respectively, 82.4%), while being the second most used metric class in analytically validated papers (58.3%). Moreover, DPM metrics have been computed in almost all the papers using datasets, as the latter are tightly connected with the usage of ML-based approaches. As concerns *NI*, they record the highest usage rate among papers using analytical validation (8 out of 12 papers), and the second in works implementing empirical or simulation-based validations (5 of 15 respectively, 29 of 68). As one would expect, this class reported the highest usage rate within papers running *network simulators* (19 of 27 papers). Metrics connected to *complexity and overhead measurement* were almost all assessed in simulation papers. A more detailed statistics show the following usage for those metrics within simulation papers: *CE*, *MS*, *C* and *EO* have been measured in respectively 11, 5, 5, 4, and 1 papers. Furthermore, according to our statistics, it appears that *CM* metrics are the most computed in overall (50 papers, 63.2%), as well as in empirical and simulation papers with respectively 73.3% and 72% usage. In other regards, *DR* were the second most used metrics in overall (32 papers), while being the most computed in analytical validation (50%). Furthermore, *DR* metrics are the most calculated within papers using *network simulators* (59.2%). The following most evaluated metrics are *EI* (19% in overall; 13.3%, 33.4%, and 17.6% in papers using respectively empirical, analytical and simulation-based validation), with a noticeable 33.3% usage rate in works running *network simulators*. As concerns *Other metrics (O)*, they appear in 11 papers and were mostly used in solutions running simulation; six of them used *network simulators*. *UF* metrics were considered in eight papers, five of them were within papers using simulation and three among analytical papers (25%). A noticeable correlation for this latter validation type is that *UF* metrics are computed in all papers implementing *game theory* based mitigation approaches. Moreover, *TI* metrics were computed in 8 papers running simulation (2 works conducted in addition analytical validation [32,85]), with *network simulators* and *datasets* being used in respectively [32,59,61,61,73,73,85,85,107]. Finally, *RP* and *SBI* metrics have been evaluated in only 3 and 2 papers respectively. *RP* were used in the context of simulation papers [37,93,109], whereas *SBI* were calculated in one simulation paper [111] and one analytical paper (Monte Carlo method) proposing a mitigation solution for JA [97].

6.7. Synthesis

To summarize our study, we have compiled Table 5, which outlines the attacks covered in this survey, the corresponding papers in which they were discussed, the most commonly used approach classes to detect each attack, as well as the most frequently utilized validation methods, datasets, and metrics for the surveyed solutions related to each specific attack. As it can be observed, perception layer attacks are considering exclusively classical approaches. As a result, datasets are nowhere to be found in papers dealing with these attacks. Moreover, *network analysis* based detection approaches were the most used against these attacks. Furthermore, analytical validation was prevalent in this layer alongside simulation with network simulators and *MATLAB* being the most used tools; mainly to compute *NI* and *DPM* metrics. Regarding network layer attacks, supervised ML algorithms are the most commonly used methods to detect such attacks. These algorithms make use of IoT datasets, and focus on computing CM metrics, using mainly simulations for their validation. Classical approaches have been also adopted for this attack class, by running in overall simulation using network simulators (Cooja) and measuring computing NI metrics. The same observation can be made for ALPF attacks. In what concerns specific technologies and environments, as for example RPL, SDN-IoT, LoRaWAN, and Blockchain, we notice that most of the works targeting SDN-based IoT networks have dealt with PF attack detection [52,53,57,59,63,72,73,83] except in [84] which

Table 5
Synthesis: Attack, Detection approach, validation method, used tools and metrics.

Attack layer	Attack type	Papers	Main detection approaches classes	Main validation method	Main datasets	Main models & tools	Main metrics
Perception layer attacks	JA	[97,98,111] [89,99][93]	Network analysis	Analytical	–	Game theory Monte Carlo	NI(DR), DPM(UF)
	BX	[31,34,106]	ID authentication & verification, Network analysis	Simulation	–	NS (Cooja), NCE (Matlab)	NI(EI), CE, DPM(UF), EO
	GR	[47][55]	Network analysis	Simulation	–	NS (Cooja), Petri Nets	NI(DR, EI), DPM(CM)
	DA	[92]	–	Empirical	–	Arduino	NI(DR)
	DS	[101]	–	Analytical	–	Petri Networks	–
Perception layer attacks(overall)			Network analysis, ID authentication & verification	Analytical Simulation	–	NS, NCE	NI(DR), EO, DPM(UF), CE
Network layer attacks	Packet-drop attacks (SF-NGH-BH)	[42,44,54] [43,107,108] [28,109] [29]	Network analysis, Confidence value	Simulation	RADAR	NS (Cooja), NCE (Matlab), Other tools	NI(DR), DPM(CM), CE
	Deceptive traffic redirection attacks CN, SY, RK, ...	[32,35,36] [37,61][43] [28,30,42] [26,27,29] [44]	Confidence value, ID authentication & verification , Network analysis	Simulation	RADAR	NS (Cooja; Netsim), NCE (Matlab), PL (MPI)	NI(DR), DPM(CM), C
	LF	[84]	Supervised ML	Simulation	–	Other Tools	DPM(CM)
	PF	[60,65,94] [40,52,80] [46,57,75] [58,74,100] [53,67] [41] [49,78,104] [48,70,81] [72,85,88] [39,68,69] [50,56,71] [43] [73,87] [33,51,86] [77,82,110] [63,66,83] [59]	Supervised ML, Network analysis Unsupervised ML	Simulation	Bot-IoT, Other datasets,	Other tools, NS (Mininet), PCGT(Wireshark), PL (sci-kit learn)	DPM(CM), NI(DR), CE
	LRDDoS	[45,62]	Network analysis	Analytical Simulation	–	NS, PCGT, NCE	DPM(CM), NI(DR)
	Other flooding attacks (RY, DIS, FBA)	[42,44,106] [107]	Network analysis,	Simulation	RADAR	NS (Cooja, Netsim), NCE (MATLAB)	NI(EI), DPM(CM), CE
	ADV	[64,79]	Supervised ML	Simulation	Mirai	Generic IDS (Kitsune), PL (Pytorch), Other tools	DPM(CM)
Network layer attacks (overall)			Supervised ML, Network analysis, ID Authentication & verification	Simulation	Bot-IoT, Other datasets, UNSW NB-15	NS (Cooja), Other Tools, PL (scikit-learn)	DPM(CM), NI(DR), CE
Application layer attacks	ALPF	[38,51,76] [67,75]	Supervised ML, Network analysis	Simulation	Bot-IoT, Other datasets, MQTTset	PL (scikit-learn)	DPM(CM), NI(DR)
Attacks overall			Supervised ML, Network analysis ID Authentication & verification	Simulation	Bot-IoT, Other datasets,	NS (Cooja), Other Tools, PL (scikit-learn)	DPM(CM), NI(DR), CE

NS: Network Simulators; NCE: Numeric Computing Environment; PL: Programming Libraries; PCGT: Packet Capture and Generator Tools.

addressed LF attack. four works [52,53,57,63] have used network analysis-based approaches validated using simulation (Mininet), whereas the works in [72,73,83,84] considered different ML approaches. Moreover, all of the works targeting RPL have used classical approaches [26–28,30,32,42,44] except the work in [61] which combined an ML approach with JSL algorithm to design a trust-based solution. These works have all considered a simulation-based validation mostly on a network simulator (Cooja in most cases). Regarding blockchain technology, it has been considered exclusively in the detection of PF attacks in simulation papers. A signature-based approach is used in [39], whereas the works in [33,40,41] adopted ML-based approaches by taking advantage of Bot-IoT and CICDDoS2019 datasets and running Hyperledger fabric. Finally, all the surveyed works that targeted LoRAWAN [89,92,93,101] have explored the impact of perception layer attacks (JA and DA) on the communication service availability, by running either analytical or empiric validation methods for their mitigation approaches evaluated by computing NI metrics.

An interesting observation that can be made on detection approaches state-of-art is the lack of usage of optimization algorithms like meta-heuristics-based solutions that may reduce computational requirements for the proposed detection solution and also improve the feature selection process for ML-based algorithms. Another interesting research perspective would be the combination of classical algorithms and ML approaches for the creation of a hybrid generic DoS and DDoS attacks detection framework that can detect multiple types of attacks; something that was not explored, as only a few papers [33,37,50,61] combined both algorithms, according to our survey. Furthermore, one can notice the lack of DoS and DDoS detection approaches proposition in both the perception and application layer. This makes the exploration of such research areas interesting as perspectives for future works targeting DoS and DDoS attacks; specifically for attacks that can be executed in real-world scenarios with relative ease such as JA and GR attacks. This statement can be generalized to attacks that have not been sufficiently addressed in the literature. As a suggestion, the use of DL and FL as well as reinforcement learning should be intensively explored to detect those attacks. Moreover, we believe that utilizing confidence value calculations can be beneficial in detecting GR or similar attacks. Specifically, we propose examining the monopolization of IoT network's bandwidth as a potential indicator of dishonest behavior during confidence value computation. This approach could help identify and prevent such attacks, ultimately enhancing the security and reliability of the IoT network. In addition, future research should investigate the efficiency of applying ML-based algorithms to detect perception layer attacks (GR, JA, BX), as far as our knowledge extends, no method has been proposed in the literature for that purpose.

To enhance the credibility of a proposed approach from a theoretical standpoint, it would be worthwhile for future studies to emphasize analytical and empirical validation methods and tools. Analytical methods offer more theoretical evidence of effectiveness, while empirical methods enable more accurate testing of approach performance in real-world situations, as opposed to simulation-based validation methods. However, this requires the implementation of suitable empirical testbeds specifically designed with IoT DoS and DDoS attack in mind, as there is a visible lack of the latter which impacts greatly the generation and hence the availability of datasets dealing with various DoS and DDoS attack types other than PF. As a result, the scarcity of diversified datasets has led the majority of the current research in this domain to always address the same attacks while mimicking in-vogue approaches and adopting the easiest and the least expensive way to validate their solutions.

7. Conclusion

Through the years, the IoT has grown increasingly ubiquitous across all industries, and smart objects have become integral to the daily lives of most individuals. As with every emerging technology, new security challenges are knocking at the door, which need to be addressed to safely take fully advantage of the services provided by this promising technology. In this survey, we have dealt with DoS and DDoS attacks in the IoT domain, which stand to be one of the most dangerous attacks as it can nullify the benefits provided by IoT smart connected objects. As a result, numerous approaches for detecting and mitigating these types of attacks have been proposed in different works throughout the literature. Moreover, these approaches have been developed, implemented and validated using a variety of validation methods and a multitude of tools. They have also been assessed by computing relevant metrics and parameters.

To provide a comprehensive insight into the research directions of the state of the art regarding this problematic, we conducted a deep study on recent works proposed in the literature by focusing on all the aforementioned aspects. Initially, we gathered a significant number of recent and noteworthy papers that address DoS and DDoS attacks in the context of IoT. Then, after inventorying all DoS and DDoS attack detection algorithms, validation methods, tools, and metrics used in the surveyed papers, we proposed different taxonomies for their classification. We then proceeded to compare the proposed solutions according to technical aspects. Subsequently, we define and answer different research questions that highlight the research directions in this domain as well as identify the most appropriate detection algorithm, validation method, tools, and evaluation metrics to consider in the solution to design and implement in dealing with each specific DoS and DDoS attack. For future work, we suggest the following:

- It would be interesting to extend this study with a classification of DoS and DDoS attack identification and mitigation methods.
- An analysis of the efficiency of the classified detection approaches would be interesting to highlight what are the optimal approaches to use in the different IoT contexts (Smart Home, Smart City, IIoT, SDN, RPL, etc.)
- Similarly, an assessment of the effectiveness of the classified validation methods, tools, and metrics would show what of the latter is more suitable to use to validate which detection, identification, and mitigation approaches.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research is supported by PRFU algerian national project under grant C00L07UN160420220010, and by ASPIRE, the technology program management pillar of Abu Dhabi's Advanced Technology Research Council (ATRC), via the ASPIRE Visiting International Professorship program.

References

[1] M. Bouakouk, A. Abdelli, L. Mokdad, Survey on the cloud-IoT paradigms: Taxonomy and architectures, in: IEEE ISCC, 2020, pp. 1–6.
[2] M. Achir, A. Abdelli, L. Mokdad, J. Benothman, Service discovery and selection in IoT: A survey and a taxonomy, JNCA (2022) 103331.
[3] Y. Sasaki, A survey on IoT big data analytic systems: Current and future, IEEE Internet Things J. 9 (2022) 1024–1036.
[4] A. Mosenia, N.K. Jha, A comprehensive study of security of internet-of-things, IEEE Trans. Emerg. Top. Comput. 5 (4) (2016) 586–602.
[5] F.A. Alaba, et al., Internet of things security: A survey, J. Netw. Comput. Appl. 88 (2017) 10–28.
[6] D.E. Kouicem, A. Bouabdallah, H. Lakhlef, Internet of things security: A top-down survey, Comput. Netw. 141 (2018) 199–221.
[7] Y. Lu, L. Da Xu, Internet of things (IoT) cybersecurity research: A review of current research topics, IEEE Internet Things J. 6 (2) (2018) 2103–2115.
[8] N. Chaabouni, et al., Network intrusion detection for IoT security based on learning techniques, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2671–2701.
[9] J. Sengupta, S. Ruj, S.D. Bit, A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT, J. Netw. Comput. Appl. 149 (2020) 102481.
[10] A. Arshad, Z.M. Hanapi, S. Subramaniam, R. Latip, A survey of sybil attack countermeasures in IoT-based wireless sensor networks, PeerJ Comput. Sci. 7 (2021) e673.
[11] M. binti Mohamad Noor, W.H. Hassan, Current research on internet of things (IoT) security: A survey, Comput. Netw. 148 (2019) 283–294.
[12] K. Lounis, M. Zulkernine, Attacks and defenses in short-range wireless technologies for IoT, IEEE Access 8 (2020) 88892–88932.
[13] A. Bahaa, et al., Monitoring real time security attacks for IoT systems using DevSecOps: a systematic literature review, Information 12 (4) (2021) 154.
[14] R.R. Krishna, et al., State-of-the-art review on IoT threats and attacks: Taxonomy, challenges and solutions, Sustainability 13 (16) (2021) 9463.
[15] A. Lohachab, B. Karambir, Critical analysis of DDoS—an emerging security threat over IoT networks, JCIN 3 (3) (2018) 57–78.
[16] F.S. Dantas Silva, et al., A taxonomy of DDoS attack mitigation approaches featured by SDN technologies in IoT scenarios, Sensors 20 (11) (2020) 3078.
[17] R. Vishwakarma, A.K. Jain, A survey of DDoS attacking techniques and defence mechanisms in the IoT network, Telecommun. Syst. 73 (1) (2020) 3–25.
[18] Y. Al-Hadhrami, F.K. Hussain, DDoS attacks in IoT networks: a comprehensive systematic literature review, World Wide Web (2021) 1–31.
[19] Z. Shah, et al., Blockchain based solutions to mitigate distributed denial of service (DDoS) attacks in the internet of things (IoT): A survey, Sensors 22 (3) (2022) http://dx.doi.org/10.3390/s22031094, URL https://www.mdpi.com/1424-8220/22/3/1094.
[20] A. Singh, B.B. Gupta, Distributed denial-of-service (DDoS) attacks and defense mechanisms in various web-enabled computing platforms: Issues, challenges, and future research directions, IJSWIS IGI Global 18 (2022) 1–43.
[21] M.R. Kadri, A. Abdelli, L. Mokdad, Evaluation metrics in DoS attacks detection approaches in IoT: A survey and a taxonomy, in: MISC, Springer, 2022, pp. 46–61.
[22] A. Tewari, B.B. Gupta, Security, privacy and trust of different layers in internet-of-things (IoTs) framework, Future Gener. Comput. Syst. 108 (2020) 909–920.
[23] V. Kumar, R.K. Jha, S. Jain, NB-IoT security: A survey, Wirel. Pers. Commun. 113 (4) (2020) 2661–2708.
[24] J. de Carvalho Silva, J. Rodrigues, J. Al-Muhtadi, R. Rabelo, V. Furtado, Management platforms and protocols for internet of things: A survey, Sensors 19 (2019) 676, http://dx.doi.org/10.3390/s19030676.
[25] Y. Hammal, J. Ben-Othman, L. Mokdad, A. Abdelli, Formal modeling of greedy nodes in 802.15. 4 wsn, ICT Express, Elsevier 1 (1) (2014) 10–13.
[26] C. Cervantes, D. Poplade, M. Nogueira, A. Santos, Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for internet of things, in: 2015 IFIP/IEEE ISIM, 2015, pp. 606–611.
[27] M. Surendar, A. Umamakeswari, InDReS: An intrusion detection and response system for internet of things with 6LoWPAN, in: 2016 WiSPNET, 2016, pp. 1903–1908.
[28] Z.A. Khan, P. Herrmann, A trust based distributed intrusion detection mechanism for internet of things, in: 2017 IEEE 31st AINA, 2017, pp. 1169–1176.
[29] D. Yuvaraj, et al., Novel DoS attack detection based on trust mode authentication for IoT, Intell. Autom. Soft Comput. 34 (3) (2022) 1505–1522.
[30] F. Medjek, D. Tandjaoui, I. Romdhani, N. Djedjig, A trust-based intrusion detection system for mobile RPL based networks, in: 2017 IEEE IThings, 2017, pp. 735–742.
[31] S. Hristozov, M. Huber, G. Sigl, Protecting restful IoT devices from battery exhaustion DoS attacks, in: 2020 IEEE HOST, 2020, pp. 316–327.
[32] C. Pu, K.-K.R. Choo, Lightweight sybil attack detection in IoT based on bloom filter and physical unclonable function, Comput. Secur. (2021) 102541.
[33] E.S. Babu, B. SrinivasaRao, S.R. Nayak, A. Verma, F. Alqahtani, A. Tolba, A. Mukherjee, Blockchain-based intrusion detection system of IoT urban data with device authentication against DDoS attacks, Comput. Electr. Eng. 103 (2022) 108287.
[34] M. Ghahramani, et al., RSS: An energy-efficient approach for securing IoT service protocols against the DoS attack, IEEE Internet Things J. 8 (5) (2020) 3619–3635.
[35] A.S.S. Thuluva, et al., Secure and efficient transmission of data based on caesar cipher algorithm for sybil attack in IoT, EURASIP J. Adv. Signal Process. 2021 (1) (2021) 1–23.
[36] K. Hameed, et al., A context-aware information-based clone node attack detection scheme in internet of things, J. Netw. Comput. Appl. 197 (2022) 103271.
[37] T.A.S. Srinivas, S. Manivannan, Prevention of hello flood attack in IoT using combination of deep learning with improved rider optimization algorithm, Comput. Commun. 163 (2020) 162–175.
[38] F. De Rango, M. Tropea, P. Fazio, Mitigating DoS attacks in IoT EDGE layer to preserve QoS topics and nodes' energy, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, IEEE, 2020, pp. 842–847.
[39] R.F. Hayat, et al., ML-DDoS: A blockchain-based multilevel DDoS mitigation mechanism for IoT environments, IEEE Trans. Eng. Manage. (2022).
[40] P. Kumar, R. Kumar, G.P. Gupta, R. Tripathi, A distributed framework for detecting DDoS attacks in smart contract-based blockchain-IoT systems by leveraging fog computing, Trans. Emerg. Telecommun. Technol. 32 (6) (2021) e4112.
[41] R. Kumar, et al., A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network, J. Parallel Distrib. Comput. 164 (2022) 55–68.
[42] A. Le, J. Loo, K.K. Chai, M. Aiash, A specification-based IDS for detecting attacks on RPL-based network topology, Information 7 (2) (2016).

[43] D. Midi, A. Rullo, A. Mudgerikar, E. Bertino, Kalis — A system for knowledge-driven adaptable intrusion detection for the internet of things, in: 2017 IEEE ICDCS, 2017, pp. 656–666.
[44] A. Agiollo, et al., DETONAR: Detection of routing attacks in RPL-based IoT, IEEE Trans. Netw. Serv. Manag. (2021).
[45] G. Liu, W. Quan, N. Cheng, H. Zhang, S. Yu, Efficient DDoS attacks mitigation for stateful forwarding in internet of things, J. Netw. Comput. Appl. 130 (2019) 1–13.
[46] U. Kumar, S. Navaneet, N. Kumar, S.C. Pandey, Isolation of DDoS attack in IoT: A new perspective, Wirel. Pers. Commun. 114 (2020) 2493–2510.
[47] A. Abdelli, L. Mokdad, J. Ben Othman, Y. Hammal, Dealing with a non green behaviour in WSN, Simul. Model. Pract. Theory 84 (2018) 124–142.
[48] H. Moudoud, L. Khoukhi, S. Cherkaoui, Prediction and detection of fdia and DDoS attacks in 5g enabled iot, IEEE Netw. 35 (2) (2020) 194–201.
[49] R. Paudel, T. Muncy, W. Eberle, Detecting DoS attack in smart home IoT devices using a graph-based approach, in: Big Data, IEEE, 2019, pp. 5249–5258.
[50] C.-L. Chen, J. Hengchang, W. Jian, Detection of DDoS attack within industrial IoT devices based on clustering and graph structure features, Secur. Commun. Netw. 2022 (1) (Jan 2022) http://dx.doi.org/10.1155/2022/1401683.
[51] R. Yaegashi, D. Hisano, Y. Nakayama, Light-weight DDoS mitigation at network edge with limited resources, in: 2021 IEEE 18th Annual Consumer Communications & Networking Conference, CCNC, IEEE, 2021, pp. 1–6.
[52] J. Bhayo, S. Hameed, S.A. Shah, An efficient counter-based DDoS attack detection framework leveraging software defined IoT (SD-IoT), IEEE Access 8 (2020) 221612–221631.
[53] J. Bhayo, et al., A time-efficient approach toward DDoS attack detection in IoT network using SDN, IEEE Internet Things J. 9 (5) (2022) 3612–3630.
[54] C. Pu, S. Lim, A light-weight countermeasure to forwarding misbehavior in wireless sensor networks: design, analysis, and evaluation, IEEE Syst. J. 12 (1) (2016) 834–842.
[55] F.S. Sadek, K. Belkadi, A. Abouaissa, P. Lorenz, Identifying misbehaving greedy nodes in IoT networks, Sensors 21 (15) (2021) 5127.
[56] F. Shaikh, et al., IoT threat detection testbed using generative adversarial networks, in: 2022 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom, IEEE, 2022, pp. 77–84.
[57] J. Galeano-Brajones, et al., Detection and mitigation of dos and DDoS attacks in IoT-based stateful sdn: An experimental approach, Sensors 20 (3) (2020) 816.
[58] M. Aridoss, Defensive mechanism against DDoS attack to preserve resource availability for iot applications, Int. J. Handheld Comput. Res. (IJHCR) 8 (4) (2017) 40–51.
[59] A. Mishra, N. Gupta, B. Gupta, Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller, Telecommun. Syst. 77 (2021) 47–62.
[60] D.K. Sharma, et al., Anomaly detection framework to prevent DDoS attack in fog empowered IoT networks, Ad Hoc Netw. 121 (2021) 102603.
[61] K. Prathapchandran, T. Janani, A trust aware security mechanism to detect sinkhole attack in RPL-based IoT environment using random forest–RFTRUST, Comput. Netw. 198 (2021) 108413.
[62] P. Bhale, S. Biswas, S. Nandi, LORD: LOw rate DDoS attack detection and mitigation using lightweight distributed packet inspection agent in IoT ecosystem, in: 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems, ANTS, IEEE, 2019, pp. 1–6.
[63] D. Yin, L. Zhang, K. Yang, A DDoS attack detection and mitigation with software-defined internet of things framework, IEEE Access 6 (2018) 24694–24705.
[64] E. Anthi, L. Williams, A. Javed, P. Burnap, Hardening machine learning denial of service (DoS) defences against adversarial attacks in IoT smart home networks, Comput. Secur. 108 (2021) 102352.
[65] H. Tyagi, R. Kumar, Attack and anomaly detection in IoT networks using supervised machine learning approaches, Rev. d'Intelligence Artif. 35 (1) (2021) 11–21.
[66] R. Yadav, I. Sreedevi, D. Gupta, Augmentation in performance and security of WSNs for IoT applications using feature selection and classification techniques, Alex. Eng. J. 65 (2023) 461–473.
[67] J.G. Almaraz-Rivera, J.A. Perez-Diaz, J.A. Cantoral-Ceballos, Transport and application layer DDoS attacks detection to IoT devices by using machine learning and deep learning models, Sensors 22 (9) (2022).
[68] M. Shafiq, et al., CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine-learning techniques, IEEE Internet Things J. 8 (5) (2020) 3242–3254.
[69] G. Shirvani, S. Ghasemshirazi, B. Beigzadeh, IoT-shield: A novel DDoS detection approach for IoT-based devices, in: 2021 11th SGC, IEEE, 2021, pp. 1–7.
[70] M.F. Ashfaq, et al., Classification of IoT based DDoS attack using machine learning techniques, in: 2022 16th IMCOM, 2022, pp. 1–6.
[71] P. Kumar, et al., Sad-IoT: Security analysis of DDoS attacks in iot networks, Wirel. Pers. Commun. 122 (1) (2022) 87–108.
[72] M. Zang, E.O. Zaballa, L. Dittmann, SDN-based in-band DDoS detection using ensemble learning algorithm on IoT edge, in: 25th ICIN, IEEE, 2022, pp. 111–115.
[73] Y. Yang, J. Wang, B. Zhai, J. Liu, IoT-based DDoS attack detection and mitigation using the edge of SDN, in: Cyberspace Safety and Security: 11th International Symposium, CSS 2019, Guangzhou, China, December 1–3, 2019, Proceedings, Part II 11, Springer, 2019, pp. 3–17.
[74] L. Huang, Design of an IoT DDoS attack prediction system based on data mining technology, J. Supercomput. 78 (4) (2022) 4601–4623.
[75] Z.A. Baig, et al., Averaged dependence estimators for DoS attack detection in IoT networks, Future Gener. Comput. Syst. 102 (2020) 198–209.
[76] S. Rachmadi, S. Mandala, D. Oktaria, Detection of DoS attack using AdaBoost algorithm on IoT system, in: ICoDSA, 2021, pp. 28–33.
[77] I. Cvitić, D. Perakovic, B.B. Gupta, K.-K.R. Choo, Boosting-based DDoS detection in internet of things systems, IEEE Internet Things J. 9 (3) (2021) 2109–2123.
[78] Y.-E. Kim, Y.-S. Kim, H. Kim, Effective feature selection methods to detect IoT DDoS attack in 5G core network, Sensors 22 (10) (2022) 3819.
[79] H. Qiu, et al., Adversarial attacks against network intrusion detection in IoT systems, IEEE Internet Things J. (2020).
[80] G.D.L.T. Parra, P. Rad, K.-K.R. Choo, N. Beebe, Detecting internet of things attacks using distributed deep learning, J. Netw. Comput. Appl. 163 (2020) 102662.
[81] Y. Jia, et al., Flowguard: an intelligent edge defense mechanism against IoT DDoS attacks, IEEE Internet Things J. 7 (10) (2020) 9552–9562.
[82] A. Mihoub, O.B. Fredj, O. Cheikhrouhou, A. Derhab, M. Krichen, Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques, Comput. Electr. Eng. 98 (2022) 107716.
[83] V. Ravi, R. Chaganti, M. Alazab, Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system, Comput. Electr. Eng. 102 (2022) 108156.
[84] Y.-H. Chen, Y.-C. Lai, P.-T. Jan, T.-Y. Tsai, A spatiotemporal-oriented deep ensemble learning model to defend link flooding attacks in IoT network, Sensors 21 (4) (2021) 1027.
[85] J. Li, L. Lyu, X. Liu, X. Zhang, X. Lyu, FLEAM: A federated learning empowered architecture to mitigate DDoS in industrial IoT, IEEE Trans. Ind. Inform. 18 (6) (2021) 4059–4068.
[86] D. Stiawan, M.E. Suryani, M.Y. Idris, M.N. Aldalaien, N. Alsharif, R. Budiarto, et al., Ping flood attack pattern recognition using a K-means algorithm in an internet of things (IoT) network, IEEE Access 9 (2021) 116475–116484.
[87] I. Ko, D. Chambers, E. Barrett, Adaptable feature-selecting and threshold-moving complete autoencoder for DDoS flood attack mitigation, J. Inf. Secur. Appl. 55 (2020) 102647.
[88] N.-N. Dao, et al., Securing heterogeneous IoT with intelligent DDoS attack behavior learning, IEEE Syst. J. (2021).
[89] M. Ingham, J. Marchang, D. Bhowmik, IoT security vulnerabilities and predictive signal jamming attack analysis in LoRaWAN, IET Inf. Secur. 14 (4) (2020) 368–379.

[90] V. La, W. Montes de Oca, A. Cavalli, A framework for security monitoring of real IoT testbeds, in: Proceedings of the 16th International Conference on Software Technologies - ICSOFT, SciTePress, 2021, pp. 645–652.
[91] S. Siboni, et al., Security testbed for internet-of-things devices, IEEE Trans. Reliab. 68 (1) (March 2019) 23–44, http://dx.doi.org/10.1109/TR.2018.2864536.
[92] G. Bernardinetti, F. Mancini, G. Bianchi, Disconnection attacks against LoRaWAN 1.0. X ABP devices, in: 2020 Mediterranean Communication and Computer Networking Conference, MedComNet, IEEE, 2020, pp. 1–8.
[93] C. Del-Valle-Soto, C. Mex-Perera, J.A. Nolazco-Flores, A. Rodríguez, J.C. Rosas-Caro, A.F. Martínez-Herrera, A low-cost jamming detection approach using performance metrics in cluster-based wireless sensor networks, Sensors 21 (4) (2021) 1179.
[94] Harada, et al., Quick suppression of DDoS attacks by frame priority control in IoT backhaul with construction of mirai-based attacks, IEEE Access 10 (2022) 22392–22399.
[95] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208, http://dx.doi.org/10.1109/TIT.1983.1056650.
[96] Cryptographic protocol verifier in the formal model, ProVerif, URL https://bblanche.gitlabpages.inria.fr/proverif/.
[97] Y. Liu, Q. Zeng, Y. Zhao, K. Wu, Y. Hao, Novel channel-hopping pattern-based wireless IoT networks in smart cities for reducing multi-access interference and jamming attacks, EURASIP J. Wireless Commun. Networking 2021 (1) (2021) 1–19.
[98] X. Tang, P. Ren, Z. Han, Jamming mitigation via hierarchical security game for IoT communications, IEEE Access 6 (2018) 5766–5779.
[99] N. Namvar, W. Saad, N. Bahadori, B. Kelley, Jamming in the internet of things: A game-theoretic perspective, in: 2016 GLOBECOM, IEEE, 2016, pp. 1–6.
[100] X. Chen, L. Xiao, W. Feng, N. Ge, X. Wang, DDoS defense for IoT: A stackelberg game model-enabled collaborative framework, IEEE Internet Things J. 9 (12) (2022) 9659–9674, http://dx.doi.org/10.1109/JIOT.2021.3138094.
[101] E. Van Es, H. Vranken, A. Hommersom, Denial-of-service attacks on LoRaWAN, in: 13th IC ARS, 2018, pp. 1–6.
[102] Time petrinet analyzer, The TINA toolbox Home Page, URL https://projects.laas.fr/tina/index.php.
[103] CPN Tools-A tool for editing, simulating, and analyzing Colored Petri nets, CPN Tools, URL https://cpntools.org/.
[104] M. Zeeshan, et al., Protocol-based deep intrusion detection for DoS and DDoS attacks using UNSW-NB15 and bot-IoT data-sets, IEEE Access 10 (2022) 2269–2283.
[105] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, T. Voigt, Cross-level sensor network simulation with COOJA, in: Proceedings. 2006 31st IEEE Conference on Local Computer Networks, 2006, pp. 641–648, http://dx.doi.org/10.1109/LCN.2006.322172.
[106] G. Glissa, A. Meddeb, 6LowPSec: An end-to-end security protocol for 6LoWPAN, Ad Hoc Netw. 82 (2019) 100–112.
[107] M. Hossain, Y. Karim, R. Hasan, Secupan: A security scheme to mitigate fragmentation-based network attacks in 6LoWPAN, in: 8th ACM DASP, 2018, pp. 307–318.
[108] G. Glissa, A. Meddeb, 6LoWPAN multi-layered security protocol based on IEEE 802.15.4 security features, in: 13th IWCMC, IEEE, 2017, pp. 264–269.
[109] K. Heurtefeux, et al., Enhancing RPL resilience against routing layer insider attacks, in: 29th ICAINA, IEEE, 2015, pp. 802–807.
[110] R.F. Ibrahim, Q. Abu Al-Haija, A. Ahmad, DDoS attack prevention for internet of thing devices using ethereum blockchain technology, Sensors 22 (18) (2022) 6806.
[111] R.E. Navas, et al., Physical resilience to insider attacks in IoT networks: Independent cryptographically secure sequences for DSSS anti-jamming, Comput. Netw. 187 (2021) 107751.