Contents lists available at ScienceDirect

Internet of Things and Cyber-Physical Systems

journal homepage: www.keaipublishing.com/en/journals/internet-of-things-and-cyber-physical-systems

Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models

Fatima Alwahedi, Alyazia Aldhaheri, Mohamed Amine Ferrag *, Ammar Battah, Norbert Tihanyi

Technology Innovation Institute, 9639, Masdar City, Abu Dhabi, United Arab Emirates

ARTICLE INFO

Index Terms:
Cyber threat detection
Intrusion detection
IoT
Machine learning
Security

ABSTRACT

Despite providing unparalleled connectivity and convenience, the exponential growth of the Internet of Things (IoT) ecosystem has triggered significant cybersecurity concerns. These concerns stem from various factors, including the heterogeneity of IoT devices, widespread deployment, and inherent computational limitations. Integrating emerging technologies to address these concerns becomes imperative as the dynamic IoT landscape evolves. Machine Learning (ML), a rapidly advancing technology, has shown considerable promise in addressing IoT security issues. It has significantly influenced and advanced research in cyber threat detection. This survey provides a comprehensive overview of current trends, methodologies, and challenges in applying machine learning for cyber threat detection in IoT environments. Specifically, we further perform a comparative analysis of state-of-the-art ML-based Intrusion Detection Systems (IDSs) in the landscape of IoT security. In addition, we shed light on the pressing unresolved issues and challenges within this dynamic field. We provide a future vision with Generative AI and large language models to enhance IoT security. The discussions present an in-depth understanding of different cyber threat detection methods, enhancing the knowledge base of researchers and practitioners alike. This paper is a valuable resource for those keen to delve into the evolving world of cyber threat detection leveraging ML and IoT security.

1. Introduction

Brendan O'Brien astutely observed, ”If you think the Internet has changed your life, think again. The Internet of Things is about to change it all over again!” [1]. This is indeed the case, as the Internet of Things (IoT) has heralded unprecedented connectivity. The advancements in sensor technology, wireless communication, and data analytics have spurred an exponential increase in connected devices. This influx of connectivity, brought about by integrating IoT into various industries, cities, and households, promotes unmatched efficiency and convenience. As the backbone of IoT, sensors and actuators acquire and convert data from the physical world into digital signals. These compact devices amass a diverse range of data, thereby enabling real-time monitoring and control of numerous systems and processes.

However, the rapid proliferation and extensive integration of IoT devices into everyday life have ushered in various security challenges. These issues must be robustly addressed to ensure the safety and reliability of this expanding ecosystem. The sheer volume and variety of IoT devices and their often inconsistent security features and protocols engender a fragmented environment teeming with potential attack vectors. IoT devices frequently prioritize low cost and user simplicity over security, making them susceptible to breaches and exploitation. As a result, these devices are at risk of various cyber threats, including data breaches, Distributed Denial-of-Service (DDoS) attacks, and malware infections. Any security breach in these devices could significantly compromise privacy and crucial infrastructure systems, given the sensitive nature of the data they handle. Moreover, IoT devices, potentially serving as entry points, might allow attackers to infiltrate broader networks, amplifying the potential impact of security breaches. Another primary concern is the security of communication routes between IoT devices and networks, as many IoT devices utilize wireless communication protocols susceptible to interception or manipulation. These vulnerabilities can be exacerbated by the resource constraints of specific IoT devices, which prevent them from adopting contemporary encryption and authentication techniques. Furthermore, the long lifespan and widespread deployment of IoT devices compound the difficulty of managing security upgrades and patches, as many devices may not receive regular updates or may be difficult to access for maintenance. This could lead to an increased number of outdated or vulnerable devices, further

* Corresponding author.
E-mail addresses: fatima.alwahedi@tii.ae (F. Alwahedi), alyazia.aldhaheri@tii.ae (A. Aldhaheri), mohamed.ferrag@tii.ae (M.A. Ferrag), ammar.battah@tii.ae (A. Battah), norbert.tihanyi@tii.ae (N. Tihanyi).

https://doi.org/10.1016/j.iotcps.2023.12.003
Received 24 August 2023; Received in revised form 27 December 2023; Accepted 27 December 2023
Available online 3 January 2024

List of abbreviations

ADC	Anticipated Distance-based Clustering
ANN	Artificial Neural Network
BN	Bayesian Network
CFS	Correlation-Based Feature Subset Selection
CPS	Cyber-Physical Security
DBScan	Density-Based Spatial Clustering
DDoS	Distributed Denial-of-Service attack
DL	Deep Learning
DT	Decision Tree
DTb	Decision Table
FBPCM	Full Bayesian Possibilistic C-mean Clustering
FGSM	Fast Gradient Sign Method
FNN	Feed-Forward Neural Network
GB	Gradient Boosting
GWO	Grey Wolf Optimization
GWO-PSO	Grey Wolf Optimization and PSO
IDS	Intrusion Detection Systems
InfoGain	Information Gain
IoT	Internet of Things
JRip	Java Ripper
KNN	K-Nearest Neighbor
KOAD	Kernel Online Anomaly Detection
LightGBM	Light Gradient Boosting Machine
LLM	Large Language Model
LNB	Likelihood Naïve Bayes
LR	Logistic Regression
MI	Mutual Information
ML	Machine Learning
MLP	Multilayer Perceptron
MKKM-IC	Multi Kernel K-means
MOEFS	Multi-Objective Evolutionary Feature Selection
MQTT	Message Queue Telemetry Transport
NLP	Natural Language Processing
NB	Naive Bayes
OCSVM	One-Class SVM
PPGO	Perpetual Pigeon Galvanized Optimization
PSO	Particle Swarm Optimization
REP Tree	Reduced Error Pruning Tree
RF	Random Forest
RFFI	Random Forest Feature Importance
RT	Random Table
SAE	Stacked Auto Encoder
S-DPN	Stacked Deep Polynomial Network
SL	Simple Logistic
SMO	Spider Monkey Optimization
SVM	Support Vector Machine
UDP	User Datagram Protocol
WVE	Weighted Voting Ensemble
ZeroR	Zero Rule
OneR	One Rule

exacerbating security concerns [2].

Given the aforementioned security challenges, machine learning (ML) has surfaced as a potent instrument for fortifying and advancing IoT security. The escalating complexity in IoT ecosystems necessitates more sophisticated security systems. ML can supply the requisite intelligence by employing intricate algorithms and insights from gathered data. It achieves this by discerning patterns, identifying anomalies, and forecasting potential threats in real-time. This capability enables a pre-emptive response to vulnerabilities and intrusions [3]. An instance of such a system is depicted in Fig. 1.

A salient application of machine learning in IoT security is anomaly detection. By scrutinizing their behavior, ML algorithms study the typical operational patterns of IoT devices, networks, and communication channels. Establishing a benchmark for normal behavior enables these

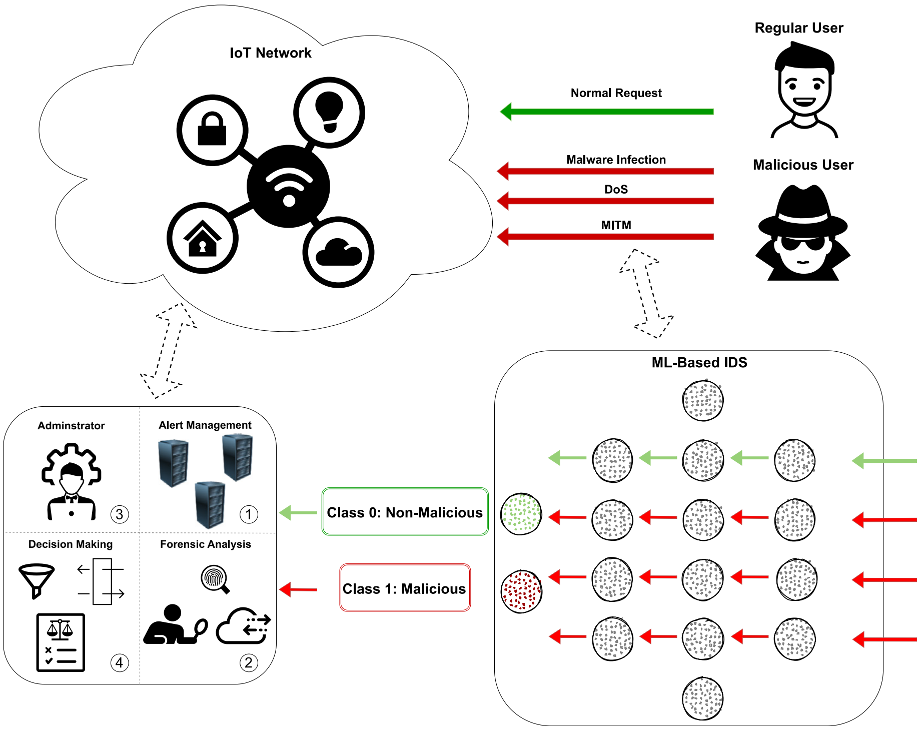

Fig. 1. A sample depiction of cyber threat detection environment based on Machine Learning.

algorithms to swiftly pinpoint deviations or abnormal activities, thereby flagging them as potential security risks. This methodology allows for early detection and response to cyberattacks such as DDoS or malware infections, preventing substantial damage.

Signature-based detection is another prevalent method that can be ameliorated by integrating ML algorithms. This approach relies on identifying known patterns or signatures of malevolent activities or malware, forming a crucial first line of defense against recognized cyber threats. By automating the signature generation and updates process, machine learning can significantly enhance the effectiveness of signature-based detection [4]. As new threats surface and evolve, ML algorithms can scrutinize vast malware sample repositories, extracting unique patterns and characteristics to generate and dynamically update signatures. Consequently, security systems can stay abreast of the evolving threat landscape, fostering a more robust defense against known and emerging threats [5].

Furthermore, ML can assist in analyzing vast volumes of data generated by IoT devices. This allows security professionals to discern hidden correlations, identify trends, and anticipate future threats [6]. Consequently, organizations can make informed decisions based on data and allocate resources more judiciously, bolstering their security. Integrating machine learning into IoT security operations is a robust ally, aiding in tackling the aforementioned challenges and vulnerabilities. This partnership has further contributed to the evolution of a more secure and resilient IoT landscape, thereby safeguarding our progressively interconnected world [7].

In this survey paper, we make the following contributions.

- We provide an exhaustive survey and critical review of recent trends in cyber threat detection methodologies.
- We present a range of ML techniques, emphasizing their respective approaches, applications, and pros and cons.
- We conduct a comparative analysis of cutting-edge ML-based Intrusion Detection Systems (IDSs).
- We discuss the current unresolved issues and challenges within IoT Security.
- We provide the future vision with Generative AI.

The paper's organization is as follows: Section I encompasses the introduction, discussing the IoT, IoT security, and the integration of ML into IoT security. This section also clearly outlines the contributions of our paper. Section II offers a summary of recent works that are relevant to ML-based cyber threat detection. Section III overviews current trends applied to IoT security, showcasing specific examples and case studies. In Section IV, we delve into numerous ML methods and techniques, evaluating them based on their recent implementations as illustrated in various research papers. Section V provides an in-depth examination and comparison of different cyber threat detection methods. Section VI discusses the open challenges that warrant attention. Lastly, Section presents the future vision of ML for Cyber Security in IoT environments, and Section VIII offers a summary of the survey and concludes the paper with final remarks.

1.1. Research strategy

The literature review for this paper was executed through a structured, multi-stage process with a specific focus on Machine Learning (ML) applications in IoT Security. Initially, a broad collection or screening of literature was conducted, drawing from an extensive pool of 22,688 potential sources related to IoT Security across scientific databases like IEEE, Springer, Science Direct, Scopus, and Web of Science. This initial collection included 16,036 conference papers, 5047 journal articles, 759 magazine articles, 494 books, 320 early access articles, 22 standards, and 10 courses.

During the systematic selection phase, we focused on sources that integrated ML in IoT Security, reducing the selection to about 10 % of the initial pool (approximately 2269 sources). The systematic selection phase then followed, during which around 50 sources were selected based on specific criteria: relevance to the ML for IoT Security topic, author, and journal reputation (preferring those with an impact factor above 3), originality of the content, publication date (prioritizing those published within the last five years), and impact (considering papers with at least 50 citations).

The selected references were then classified into two main categories: technical papers (about 90 % of the selected works) and survey papers (10 %). In the final analysis phase, critical information was extracted from approximately 80 % of the technical papers and 90 % of the survey papers. This information was thoroughly analyzed and synthesized into the comprehensive survey presented in this paper, offering a detailed insight into the intersection of ML and IoT Security.

2. Related surveys

Thakkar and Lohiya [16] surveyed various Intrusion Detection Systems (IDS) strategies for IoT networks. The study underscored the necessity for robust security mechanisms to tackle the complex architecture of IoT devices and their inherent communication capabilities, which often lead to security vulnerabilities. The article delved into diverse strategies for IDS placement analysis within the IoT architecture, the broad spectrum of intrusions specific to IoT, and the application of ML and Deep Learning (DL) methods in detecting attacks within IoT networks. The paper highlighted security challenges in IoT networks, suggesting that additional research is warranted in areas such as expanding the scope of attack types, IDS management, enhancing IDS communication security amongst devices, utilizing standardized datasets, and developing techniques to correlate alerts.

Da Costa et al. [8] underscored the persistent challenge of intrusion detection in the IoT context. As the Internet evolves into the IoT, attention has transitioned from mere connectivity to a more focused concern on data security. Their paper scrutinized recent advancements in intrusion detection and the utilization of intelligent techniques for IoT data security. The surveyed literature primarily discussed the apprehensions and efforts of the scientific community and industry to devise optimized security protocols that balance protection and energy consumption. The study also exhibited intelligent techniques employed in computer network security, specifically within intrusion detection, aimed at improving recognition rates. Nevertheless, the reduction of false-positive rates still poses a challenge. Specific techniques may lower these rates but at the cost of extended training and classification time. In contrast, others maintain a stable false positive rate but impose a significant computational burden during the training and testing phases. This dilemma is particularly pertinent in intrusion detection, where real-time detection is crucial.

Ashraf et al. [14] provided a comprehensive overview of the application of ML and DL techniques in Intrusion Detection Systems (IDS) specifically designed for IoT networks and systems. The paper elucidated the IoT architecture, associated protocols, vulnerabilities, and potential protocol-level attacks. It also surveyed many research efforts centered on IDS methodology and attack detection techniques specific to IoT. Moreover, the paper sheds light on the available ML and DL techniques for IoT IDS and offers an overview of datasets suitable for IoT security-related research. In conclusion, the authors identified several ongoing challenges, suggesting that the available IDS for IoT are still imperfect and require further refinement.

Ahmad and Alsmadi [17] conducted an examination of the prevailing trends in the utilization of ML techniques for IoT security. They structured their approach to evaluating the most recent research and developing trends in IoT security through an in-depth analysis of the most relevant and scholarly literature from 2019 to 2020. Their study was focused on amalgamating three burgeoning domains: IoT, machine learning, and information security, which consequently led to the formulation of six research questions. This literature review allowed them to identify more recent studies focusing on machine learning techniques to thwart large-scale attacks on IoT devices. The primary objective of their research was to secure IoT devices from widespread attacks such as Distributed Denial of Service (DDoS) and botnets. Both machine and deep learning delivered promising results compared to traditional intrusion detection methods. However, the rapid evolution of IoT devices and cyber-attacks poses a significant challenge for detecting zero-day threats. The paper offered a detailed background and analyzed selected papers to extract methodology and performance results. The review results addressed the six research questions to provide a concentrated yet comprehensive understanding of the latest trends, limitations, and challenges, all to aid in developing efficient intrusion detection systems in the future. The research questions were as follows: What are the primary security issues that render IoT devices vulnerable to hostile incursions? What strategies are employed to secure IoT systems? What types of large-scale attacks have affected IoT devices? What different machine learning and deep learning techniques have been utilized by researchers? What preventative measures are taken against broad-scale attacks? How often is deep learning proposed as a solution for large-scale attacks on IoT?

Hossain et al. [15] delved into the application of ML and DL in the IoT sphere from a security and privacy perspective. The paper emphasized the security and privacy challenges, attack methods, and security needs in IoT. The authors discussed different ML and DL techniques and their application in IoT security, including the limitations of conventional ML approaches. The paper also examined existing security solutions, pointing out current gaps and emerging research areas. To overcome the limitations of ML in IoT security, the authors suggested strengthening the core components of DL and Deep Reinforcement Learning (DRL). These should be evaluated based on learning efficiency and computational complexity. Furthermore, they proposed that innovative combinations of learning strategies and data visualization techniques are vital for practical data interpretation.

Liang et al. [9] debated the pros and cons of deploying ML for cyber-physical security (CPS) and IoT. They presented the benefits of using ML to bolster security in Intrusion Detection Systems (IDS) and CPS while highlighting specific issues and challenges. The paper detailed the vulnerabilities of ML across different stages, including data collection, pre-processing, training, validation, and implementation. Finally, they raised serious concerns about the potential misuse of ML in launching malicious attacks and proposed potential solutions to these issues.

Chaabouni et al. [10] thoroughly reviewed multiple topics pertinent to Network Intrusion Detection Systems (NIDSs) for IoT, utilizing various learning techniques. This review encompassed existing NIDS datasets, implementation tools, and open-source sniffing tools, which were meticulously evaluated. The discussion delved into NID's architecture, deployment, and methodology in IoT systems, encompassing traditional and ML based defense mechanisms. The authors scrutinized the state-of-the-art learning NID for IoT ecosystems, introduced learning terminologies, and compared different strategies. This state-of-the-art review demonstrated high detection accuracy and low false favorable rates. The paper compared leading IoT NID proposals and emphasized potential future research directions, focusing mainly on machine learning algorithms.

To provide a comprehensive review, Al-Garadi et al. [11], presented ML as separate sections, despite the latter being a subset of the former. They meticulously examined ML and DL algorithms for IoT security, discussing their applications, strengths, and weaknesses in detail. Furthermore, the authors highlighted the challenges of employing ML and DL for IoT system security. The paper also offered an overview of general IoT systems (i.e., methodologies, characteristics, and security threats) and deliberated on potential vulnerabilities and different attack surfaces within IoT systems. Moreover, a comparison between ML and DL methods used for securing different layers of IoT was provided. Additionally, the authors discussed concerns regarding IoT data, learning strategies, interconnected environment operations, and possible exploitation of ML and DL by attackers and underscored privacy and security challenges in IoT.

Tahsien et al. [12] concentrated on Machine Learning-based security solutions for IoT systems, incorporating the most recent publications up to 2019. The authors initiated the discussion by introducing the layers of the IoT system and the various security challenges these layers confront, including different forms of cyber-attacks. The review discussed various machine-learning techniques and how they could be employed to address various attacks on IoT systems. The authors offered a state-of-the-art review of security solutions for IoT devices, specifically focusing on applying ML algorithms across the three layers of the IoT system. Conclusively, the authors expounded on the challenges and limitations of ML-based security solutions for IoT systems and proposed potential directions for future research.

Wu et al. [13] conducted a comprehensive examination of the unique characteristics and intricacies of IoT security protection while also exploring how AI methods, including ML and DL, can be leveraged to devise IoT security solutions. Furthermore, the study provided an extensive overview of AI solutions, contrasting a variety of related algorithms and technologies against four critical security threats: device authentication, intrusion detection, malware detection, and defense against both Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. For future work, the researchers could explore the potential challenges of employing AI in IoT security. In another insightful work, Zeadally and Tsikerdekis (2020) [26] studied the various features of IoT devices and the common security threats they frequently encounter. In addition, they classified network-based and host-based machine learning approaches, emphasizing the strengths and weaknesses of each to illuminate ways to improve IoT security. The study also addressed challenges like adversarial ML and algorithm portability.

Compared to all the above-related works, our survey presents a novel contribution to the field, uniquely encompassing all three dimensions of IoT research: ML methods, adapted measures and functions, and IoT challenges. Previous works such as those by Thakkar & Lohiya (2021) [16], Ashraf et al. (2020) [14], Hussain et al. [15], Chaabouni et al. [10], Al Garadi et al. [11], Tahsien et al. [12], Wu et al. [13], and Zeadally & Tsikerdekis [26] included an examination of ML methods and IoT challenges, they did not incorporate a discussion on adapted measures and functions. Similarly, studies by da Costa et al. [8], Ahmad & Alsmadi [17], and Liang et al. [9] focused exclusively on IoT challenges. In contrast, our survey uniquely combines all these components, providing a comprehensive understanding of the complex IoT landscape, making it a pioneering effort in the field. By integrating these aspects, our survey adds a new depth to the existing literature, facilitating the exploration of new research directions. In addition, this survey uniquely includes a review of the most recent papers in the field, covering works published from 2019 to 2023. This focus ensures that our analysis and conclusions are drawn from the latest trends and developments in IoT. As such, our work not only offers a broader perspective on the topic but also presents an up-to-date overview of the cutting-edge research, making it a valuable resource for anyone seeking to understand the current state of IoT research, the application of ML methods in this domain, and the recent trends in adapted measures and functions.

3. ML in IoT security: case studies

Machine learning (ML) models are revolutionizing the way we secure IoT networks by providing real-time monitoring of network traffic to detect anomalies indicative of security breaches. These models continuously analyze data patterns from connected devices, identifying deviations from normal behavior that might signal a cyber threat. This section overviews current trends applied to IoT security, showcasing specific examples and case studies. These illustrate how ML methodologies are implemented in real-world scenarios.

3.1. *Anomaly detection systems*

ML models are used for anomaly detection in IoT networks. They monitor network traffic in real-time, identifying unusual patterns indicative of security breaches and adapting to new threats using historical data [5]. What sets ML apart is its ability to adapt and evolve; it uses historical data to learn and recognize new and emerging threats, ensuring that security measures evolve alongside the changing nature of cyber attacks. This approach not only enhances the ability to identify potential threats preemptively but also helps tailor the security protocols to the unique characteristics of each IoT network, thereby offering a more robust and responsive defense mechanism against a wide range of cyber threats.

3.2. *Predictive maintenance in industrial IoT*

In modern industrial environments, applying ML algorithms for the predictive maintenance of IoT devices has become increasingly significant. This approach involves a proactive maintenance strategy that leverages the extensive sensor data collected by IoT devices to anticipate and address potential equipment failures before they occur [27]. IoT devices in industrial settings are equipped with various sensors that continuously monitor and collect data regarding the performance and condition of machinery. This data may include temperature, vibration, pressure, and other operational parameters. ML algorithms analyze this vast amount of data to identify patterns and anomalies that may indicate potential failures or malfunctions.

3.3. *Smart home security systems*

In the rapidly evolving consumer space, ML has become a pivotal technology in bolstering the security of smart home devices. This integration of ML into home security systems is transforming how security is managed in residential spaces. One of the key applications of ML in this domain is through advanced facial recognition technologies. Unlike traditional security systems that rely on static passcodes or keys, ML-enabled systems can dynamically recognize the faces of residents and regular visitors, providing a more personalized and secure experience. This technology continually adapts and learns, improving its accuracy over time by analyzing the various faces it encounters [28]. Furthermore, these smart security systems can learn and understand regular household patterns and routines. By doing so, they can detect anomalies or unusual activities. For example, if there's movement in the house when it's usually empty or if a door is opened in an unusual manner or at an odd hour, the system can alert the homeowner. This feature is particularly beneficial for monitoring elderly family members or the house while away.

3.4. *Automotive security*

Connected vehicles represent a significant advancement in the automotive industry, integrating communication technologies into vehicles. These technologies enable cars to communicate with each other (V2V - vehicle-to-vehicle), with infrastructure (V2I - vehicle-to-infrastructure), and with other devices (V2X - vehicle-to-everything), enhancing overall transportation efficiency, safety, and convenience. Beyond security, ML algorithms can predict potential vehicle faults before they occur [29]. By analyzing historical data, ML can identify patterns that typically precede equipment failures, allowing for preemptive maintenance and reducing the risk of malfunctions that cyber threats could exploit. While ML significantly enhances automotive security, it also presents challenges. These include ensuring the privacy of collected data, guarding against ML model manipulation, and the need for regular updates and maintenance of the ML systems to keep up with evolving cyber threats.

3.5. *Healthcare IoT security*

IoT devices, such as wearable health monitors, connected medical equipment, and patient tracking systems, have become increasingly integral in the healthcare sector. These devices collect, transmit, and process vast amounts of sensitive patient data, necessitating robust security measures. ML algorithms can enhance the security of data transmission between IoT devices and the central servers. This includes ensuring encryption standards and identifying potential intercepts or data leakages in real-time. In addition, ML can automate the process of responding to security threats. For instance, an ML system could temporarily restrict access or alert security personnel upon detecting suspicious activity, reducing reliance on manual monitoring.

3.6. *Supply chain monitoring*

ML models in supply chain management monitor the integrity of goods, especially in sensitive industries. They detect tampering or deviations in environmental conditions, enhancing supply chain security and reliability [30]. Specifically, ML models can analyze data from various sources, such as sensors and IoT devices attached to products or packaging, to ensure that the goods remain in their intended state throughout the supply chain. This could involve monitoring for signs of tampering, damage, or unauthorized access to the products [31].

These case studies demonstrate the adaptability and effectiveness of ML in enhancing IoT security across various sectors, providing innovative solutions to complex security challenges (see Table 1).

4. Cyber threats' detection methods

In this section, we delve into various strategies adopted by studies within cyber detection. While some research employs mainstream methodologies, many have leveraged specialized ML techniques. These prevalent methodologies and distinct ML approaches are categorized separately for comprehensive understanding. Fig. 2 visually represents the methods discussed. In addition, Table II summarizes these methods, the types of attacks they address, and the corresponding evaluations.

4.1. *Deep learning*

Otoum et al. [32] proposed a deep learning-oriented intrusion detection framework designed to enhance IoT attack detection accuracy. The traditional intrusion detection systems (IDS) presented in the literature tend to suffer from suboptimal feature selection and inadequate dataset management. The framework proposed employs the Spider Monkey Optimization algorithm (SMO) for optimal feature selection and a Stacked-Deep Polynomial Network (SDPN) to detect data anomalies via classification. Their DL-IDS model was assessed using the NSL KDD dataset and demonstrated an impressive accuracy rate of 99.02 %.

Similarly, Ge et al. [33] suggested a DL oriented intrusion detection system. This system utilized the Bot-IoT dataset for training and testing and a Feed-Forward Neural Network (FNN) model for binary and multi-class classification of various types of attacks, including reconnaissance, information theft, DoS, and DDoS. The system's effectiveness was evaluated on four parameters: recall, precision, accuracy, and the F1 score. The model achieved impressive results, exceeding 98 % across all parameters for different attack detection.

4.2. *Adversarial attacks*

Papadopoulos et al. [34] conducted experiments on adversarial attacks aimed at both traditional machine learning and deep learning IDS models to assess their robustness. Two types of attacks were launched against these models: label poisoning and the Fast Gradient Sign Method (FGSM). The former attack induces incorrect classification, whereas the latter evades detection measures. The aim was to determine if such

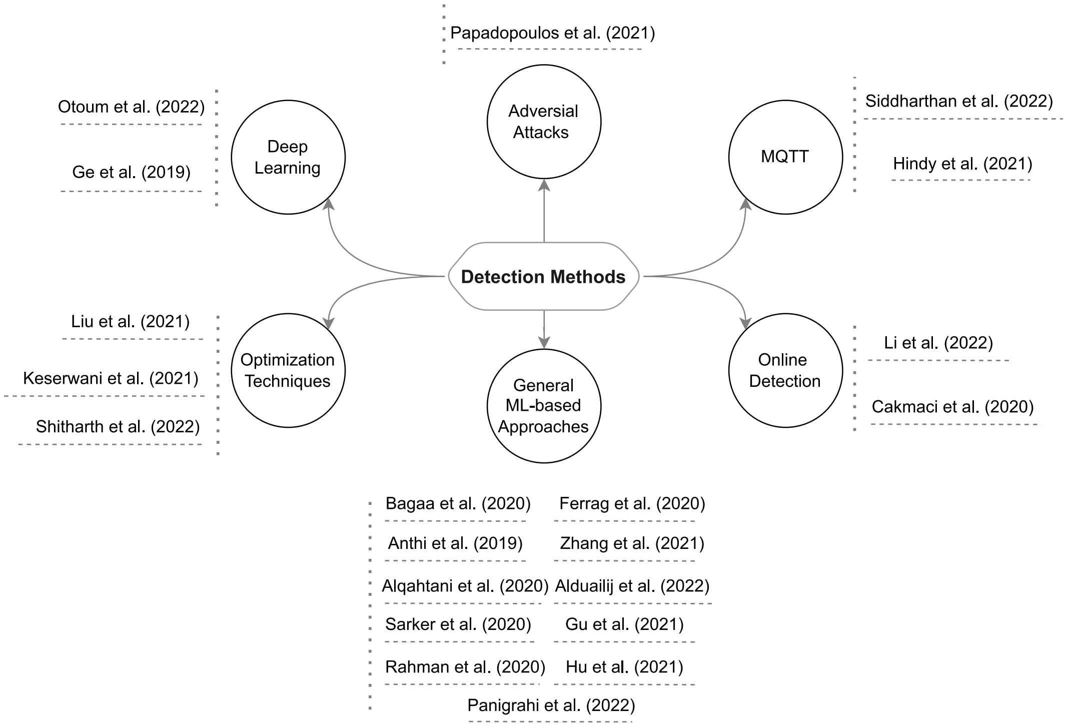

Fig. 2. Overview of State-of-the-art work in the cyber threat detection domain.

Table 1
Comparison with related surveys.

Year	Authors	ML methods	IoT Challenges	Generative AI
2019	da Costa et al. [8]	x	Ć	x
2019	Liang et al. [9]	x	Ć	x
2019	Chaabouni et al. [10]	Ć	Ć	x
2020	Al-Garadi et al. [11]	Ć	Ć	x
2020	Tahsien et al. [12]	Ć	Ć	x
2020	Wu et al. [13]	Ć	Ć	x
2020	Ashraf et al. [14]	Ć	Ć	x
2020	Hussain et al. [15]	Ć	Ć	x
2021	Thakkar & Lohiya [16]	Ć	Ć	x
2021	Ahmad & Alsmadi [17]	x	Ć	x
2023	Ferrag et al. [3]	Ć	Ć	x
2023	Aldhaheri et al. [18]	x	Ć	x
2023	Alex et al. [19]	Ć	Ć	x
2023	Siwakoti et al. [20]	Ć	Ć	x
2023	Mathur et al. [21]	x	Ć	x
2023	Issa et al. [22]	x	Ć	x
2023	Turner et al. [23]	x	Ć	x
2023	Ahmadvand et al. [24]	x	Ć	x
2023	Ahanger et al. [25]	x	Ć	x
/	This survey	Ć	Ć	Ć

x: Not Supported, Ć: Fully supported.

attacks would significantly impact model accuracy or precision. The experiments confirmed that these attacks could considerably compromise intrusion detection's effectiveness.

4.3. MQTT

Siddharthan et al. [35] proposed an IDS that utilizes Elite Machine Learning (EML) algorithms and implemented the Message Queue Telemetry Transport (MQTT) to uphold time constraints between devices. They employed the SEN-MQTTSET dataset for their system, which the authors themselves generated. Data were collected via a sensor under three different DoS scenarios: normal case, attack on a subscriber, and attack on a broker. The system evaluation involved multiple machine learning algorithms such as Logistic Regression (LR), Random Forest (RF), Naive Bayes (NB), K-Nearest Neighbor (k-NN), Decision Tree (DT), Support Vector Machine (SVM), and Gradient Boosting (GB). Various parameters, including accuracy and the F1 score, were considered during the evaluation. The proposed model yielded promising results, particularly with algorithms such as DT, RF, and GB, demonstrating an accuracy exceeding 99 %.

Hindy et al. [36] proposed a machine learning-based IDS for detecting Message Queue Telemetry Transport (MQTT) attacks. They generated a unique IoT MQTT dataset for training and evaluation purposes, encompassing multiple attack types such as Sparta SSH brute-force, aggressive scan, MQTT brute-force attack, and User Datagram Protocol (UDP) scan. Several classification algorithms were employed, including NB, LR, RF, k-NN, DT, and SVM. The research compared flow-based feature detection with packet-based feature detection using multiple performance parameters (recall, precision, and the F1 score). The results suggested that flow-based features were more suitable for MQTT-based networks than packet-based ones due to the similarity between benign MQTT communication and attacks. This distinction was especially notable in benign traffic and MQTT brute-force attacks.

4.4. Optimization techniques

Liu et al. [37] proposed an IoT intrusion detection model based on Particle Swarm Optimization (PSO). They utilized the UNSW-NB15 dataset for training and testing and employed the One-Class SVM (OCSVM) algorithm to recognize normal and abnormal data. The results demonstrated that the PSO-LightGBM model exhibited a strong capacity for effective detection of high-frequency data types such as Normal and Generic, and it significantly improved the detection rate for Worms, Shellcode, and Backdoors. Concerning accuracy and the rate of false positives, the model also performed admirably. Its data detection, identification efficiency, robustness, and generalization demonstrated in simulation experiments make it suitable for practical IoT applications.

Keserwani et al. [38] put forward an IDS based on machine learning, named GWO-PSO-RF-NIDS, which uses a hybrid of Grey Wolf Optimization and Particle Swarm Optimization (GWO-PSO) for feature selection. The model was trained and tested using various datasets, including

Table 2
Deep learning approaches for cyber threat intelligence detection.

Reference	Year	Methods	Attacks	Accuracy	Recall (or detection rate)	F1	Precision
Otoum et al. [32]	2022	SMO S-DPN	DoS, U2R, R2L, Probe	99.02 %	99.38 %	N/A	99.38 %
Ge et al. [33]	2019	FNN	DoS, DDoS, Reconnaissance, Information theft	Binary classification: (DoS/reconnaissance attacks) Above 99 %	Above 99 %	Above 99 %	Above 99 %
Papadopoulos et al. [34]	2021	SVM ANN	DoS, DDoS, Theft, Reconnaissance	After Random: 0.441 targeted: 0.610	After Random: 0.613 targeted: 0.913	After Random: 0.612 targeted: 0.737	After Random: 0.610 targeted: 0.621
Siddharthan et al. [35]	2022	LR, KNN, RF, NB SVM, GB, DT	DoS	DT = Above 99 %	DT = 100 %	DT = 100 %	DT = 100 %
Hindy et al. [36]	2021	LR, Gaussian NB k-NN, SVM, DT RF	Brute-force attacks and Scanning attacks	N/A	Uni: 93.77 % Bi: 98.85 %	Uni: 82.42 % Bi: 98.46 %	Uni: 97.19 % Bi: 99.04 %
Liu et al. [37]	2021	PSO-LightGBM OCSVM	Backdoor, Shellcode, Generic, DoS, Reconnaissance, Analysis, Fuzzers, Worms, Exploits	86.68 %	Backdoor = 51.28 % Shellcode = 64.47 % Worms = 77.78 %	N/A	N/A
Keserwani et al. [38]	2021	GWO PSO RF	DDoS, DoS:slowloris, DoS:Slowhttptest DoS Hulk, DoS:GoldenEye, Heartbleed, PortScan, Bot, FTP-Patator, SSH-Patator, Web attack BruteForce, Web attack-XSS, Web attack-SQL injection, Infiltration	CICIDS-2017 = 99.88 % NSL–KDD = 99.24 KDDCup99 = 99.66 Average = 99.66 % (for multiclass)	Binary classification CICIDS-2017 = 100 %	Binary classification CICIDS-2017 = 100 %	Binary classification CICIDS-2017 = 100 %
Shitharth et al. [39]	2022	ADC, DBSCAN PPGO, LNB	Different attacks for each dataset	89.56 % In general.	93.89 % In general.	NSL-KDD = 97.584 % CICIDS2017 = 99.998 % Bot-IoT = 99.993 %	N/A
Li et al. [40]	2022	FBPCM AdaBoost	MITM, Reconnaissance, DDoS, Exploit attack Other datasets have different attacks	82 %	80.3 %	79.7 %	79.2 %
Çakmakçı et al. [41]	2020	Shannon entropy KOAD Mahalanobis distance Chi-square test	DDoS	99.55 %	95.24 %	N/A	95.24 %
Bagaa et al. [42]	2020	J48, BN, RF Hoeffding Tree AdaBoost	DoS, U2R, R2L, Probe	Between 98.7 % and 99.9 %	Between 98.7 % and 98.9 %	N/A	N/A
Anthi et al. [43]	2019	NB, BN, J48 ZeroR, OneR, SL SVM, MLP, RF	Various reconnaissance, IoT-scanner, various DoS, various MITM, replay attack, ARP/DNS spoofing, 4 multi-stage scripts	N/A	J48 = 89.9 %	J48 = 88.8 %	Binary detection J48 = 90 %
Alqahtani et al. [44]	2020	BN, NB, RF, DT RT, DTb, ANN	DoS, U2R, R2L, Probe	RF = 94 %	RF = 93 %	RF = 97 %	RF = 99 %
Sarker et al. [45]	2020	IntruDTree (DT)	Binary classification “anomaly”	98 %	98 %	98 %	98 %
Rahman et al. [46]	2020	SAE, SVM, CFS InfoGain, OneR MLP, J48	Impersonation attack	Semi = 99.97 % Distributed = 97.80 %	Semi = 99.96 % Distributed = 98.26 %	Semi = 99.97 % Distributed = 97.81 %	N/A
Ferrag et al. [47]	2020	REP Tree, JRip Forest PA	DoS, Brute-Force, Web Attack (Several subcategories)	CICIDS2017: 96.665 % Bot-IoT: 96.995 %	CICIDS2017: 94.475 % Bot-IoT: 95.175 %	N/A	N/A
Zhang et al. [48]	2021	MFFSEM, DT RF	Dataset dependent	CIC-IDS2017 = 99.95 %	CIC-IDS2017 = 99.95 %	N/A	N/A
Alduailij et al. [49]	2022	MI, RFFI, RF, GB WVE, KNN, LR	DDoS	99.997 %	LR = 94 % KNN ≥99 % GB ≥ 98 % RF = a WVE = 100 %	LR = 94 % KNN ≥ 0.99 % GB ≥ 98 % RF ≥ 99 % WVE = 100 %	LR = 95 % KNN = 0.99 % GB ≥ 98 % RF ≥ 99 % WVE = 100 %
Gu et al. [50]	2021	NB SVM	Different attacks	NSL-KDD = 97.58 % NB-SVM = 99.35 %	NB-SVM = 99.24 %	N/A	N/A

(*continued on next page*)

Table 2 (*continued*)

Reference	Year	Methods	Attacks	Accuracy	Recall (or detection rate)	F1	Precision
				NB-SVM2 = 99.36 % (best result)	NB-SVM2 = 99.25 %		
Hu et al. [51]	2021	MKKM-IC	Dos, Probing, R2L, and U2R. Fuzzers, Analysis, Backdoors, Dos, Exploits, Generic, Reconnaissance, Shellcode, Worms, Flooding, Impersonation, and Injection	AWID (best result) 95.60 %	N/A	89.11 %	88.24 %
Panigrahi et al. [52]	2022	MOEFS DTb NB	Botnets, Port scan, DoS/DDoS Brute force attacks Web attacks (multiple)	96.8 %	96.70 %	N/A	97.40 %

NSL-KDD, KDDCup99, and CICIDS-2017. Multiple classifiers, including Decision Tree, Logistic Regression, Random Forest, and Naïve Bayesian, were employed for data classification in conjunction with the hybrid GWO-PSO. A comparison of the results from these classifiers indicated that the RF provided the highest average accuracy (99.66 %) across the three datasets.

Shitharth et al. [39] introduced a novel clustering-based classification method for network intrusion detection (NID) designed to overcome the limitations of existing NIDs. They used three datasets for implementation: Bot-IoT, NSL-KDD, and CICIDS. The key idea behind the proposed method is to include a clustering phase after data normalization to enhance classification accuracy. The clustering methods used were Anticipated Distance-based Clustering (ADC) and Density-Based Spatial Clustering (DBScan), and the feature selection technique employed was a novel approach known as Perpetual Pigeon Galvanized Optimization (PPGO). The Likelihood Naïve Bayes (LNB) classifier was used, which identified and integrated optimal parameters with the classifier to enhance accuracy and efficiency. The results were particularly noteworthy for the CICIDS 2017 dataset, where the proposed method achieved an accuracy of 99.99 %, surpassing other methods.

4.5. Online detection

Li et al. [40] proposed an online fuzzy Intrusion Detection System (IDS) with enhanced adaptability, explicitly aiming to handle concept drift, reduce noise impact, and prevent overfitting. In pursuit of these objectives, the system employed Full Bayesian Possibilistic Clustering (FBPCM) and multiple fuzzy decision trees consolidated under a sample reweighting scheme. The system was trained and evaluated using various datasets, including UNSW-NB15, KDD'99, CIC-IDS, and a novel dataset developed by the authors. Model performance was assessed using precision, accuracy, F1 score, and recall. The researchers also compared the performance of their model with that of different classification methods on the mentioned datasets. The proposed system delivered commendable results, with an accuracy of 82 %, highlighting the benefits of implementing fuzzy systems within the IoT domain.

Çakmakçı et al. [41] introduced a novel adaptive IDS technique to tackle online Distributed Denial of Service (DDoS) attacks. Their approach integrates multiple methods, such as the Mahalanobis distance metric, kernel-based anomaly detection, and the chi-square test, eliminating data labeling. The proposed system was trained and evaluated using the CICIDS2017 dataset, with model performance assessed through various metrics, including precision, accuracy, and recall. The system demonstrated promising results, boasting an accuracy of 99.55 %, a recall of 95.24 %, and a precision of 95.24 %.

4.6. General machine learning-based approaches

Bagaa et al. [42] proposed an innovative ML based security framework for IoT systems. The core concept of their proposal involved integrating Network Functions Virtualization (NFV) and Software Defined Networking (SDN) with ML to achieve optimal security through a closed-loop automation process. This process comprises a monitoring agent and an AI-reacting agent. Utilizing network patterns or IoT misbehaviors, the monitoring agent scrutinizes traffic and reports suspicious activities to the AI-reacting agent. The AI agent then reacts by classifying threats using the NSL KDD dataset and three distinct techniques: distributed data mining system, supervised learning, and neural networks. The framework demonstrated impressive results, particularly with the data mining technique, due to its high performance and low costs. The framework was further incorporated into an anomaly-based Intrusion Detection System (IDS), which utilized a One-Class Support Vector Machine (SVM) and achieved a detection accuracy exceeding 98.

Anthi et al. [43] proposed a supervised 3-layer IDS employing ML. The system possesses three functionalities: profiling the IoT device, binary classification (i.e., normal vs. malicious), and multi-classification (i.e., specifying the type of attack). Empirical validation and a dataset were compiled for training and testing. Multiple classifiers were employed to evaluate the model, with J48 delivering the best performance results across all three functionalities, scoring 96.2 %, 90.0 %, and 98.0 % respectively. The paper also furnished valuable resources to facilitate IDS automation.

Alqahtani et al. [44] proposed an IDS leveraging a variety of prominent machine learning classifiers, including Random Tree, Bayesian Network, Artificial Neural Network, Decision Tree, Decision Table, Random Decision Forest, and Naive Bayes classifier. The KDD'99 dataset was used for training and testing. Various performance indicators, such as recall, precision, F1-score, and overall accuracy, were assessed to evaluate the model's performance. The random forest algorithm achieved the most impressive results, which achieved 94 % accuracy, 99 % precision, 93 % recall, and a 97 % F1 score.

Sarker et al. [45] proposed a machine learning-based IDS, "IntruDTree" which prioritizes security features based on their importance and subsequently constructs a generalized tree-based model using these selected vital features. A unique dataset was employed for training and testing. The model provided anomaly-based binary classification. Regarding performance evaluation metrics, accuracy, recall, F1, and precision were all evaluated, with the model scoring 98 % across each parameter.

Rahman et al. [46] introduced parallel ML-based IDS models using two techniques - distributed and semi-distributed - to mitigate latency issues found in centralized IDS models. The semi-distributed approach conducts feature selection at the edge side, where the parallel models operate, while the multi-layer perceptron classification occurs in the fog. Conversely, the distributed method enables both the feature selection and the multi-layer perceptron classification to be carried out by parallel models on the edge side, reserving the final decision-making process for the fog. Training and testing were performed on the AWID dataset, utilizing various algorithms like SVM, SAE, OneR, CFS, Information Gain, J48, and MLP for feature extraction, selection, and classification. Performance evaluation parameters encompassed recall, accuracy, F1 score, and others. The study compared the scores achieved by various approaches, finding the two proposed models to yield promising results: the semi-distributed model delivered 99.97 % accuracy, and the distributed model required 73.52 % CPU time to build the model (TTBM).

Ferrag et al. [47] proposed a hierarchical IDS model (RDTIDS) that amalgamates three classification algorithms - JRip, REP Tree, and Forest PA - to minimize classification errors and enhance IDS performance. In the proposed system, two classifiers process raw data in parallel, one providing binary classification and the other offering multi-classification. Furthermore, the third classifier processes the raw data alongside the output from the first two classifiers to provide multi-classification. The model was trained and tested on two datasets, namely CICIDS2017 and Bot-IoT, and could classify various attacks grouped into three main categories: DoS, Brute-Force, and Web Attack. The results demonstrated an accuracy exceeding 96 % and a detection rate above 94 % for both datasets.

Zhang et al. [48] suggested a multi-dimensional feature fusion stacking ensemble mechanism (MFFSEM) for network intrusion detection (NID), observing that existing ML methods do not sufficiently consider the influence of different data types or sources and how they might interact or complement each other. The proposed method seeks to advance NID by effectively detecting anomalies. Several datasets were used for training and testing, including CIC-IDS2017, NSL-KDD, KDD Cup 99, and UNSW-NB15. The MFFSEM method combines two classification algorithms: Decision Tree, employed as a primary learning algorithm, and Random Forest, used as a meta-learning algorithm. When compared to the results of the primary and meta classifiers (DT and RF), it was found that MFFSEM offered superior detection. Notably, the proposed scheme outperformed other contemporary schemes regarding accuracy and recall, particularly with the CIC-IDS2017 dataset (achieving 99.95 % accuracy and recall).

Alduailij et al. [49] introduced a machine learning technique for detecting DDoS attacks in cloud computing, intending to reduce misclassification errors in DDoS detection. They used two versions of the CICIDS dataset (CICIDS 2017 and 2019) for training and testing. Feature selection was accomplished using the Random Forest Feature Importance (RFFI) and Mutual Information (MI) methods. Classification was conducted using various classifiers, including the Weighted Voting Ensemble (WVE), Gradient Boosting (GB), K Nearest Neighbor (K-NN), Random Forest (RF), and Logistic Regression (LR). The proposed method achieved high accuracy (99.997 %) compared to other techniques.

Gu et al. [50] proposed an IDS incorporating data quality considerations by utilizing an SVM coupled with naïve Bayes feature embedding to enhance the system's performance. The researchers developed two versions of the proposed system: The first version, NB-SVM, uses only the transformed feature (i.e., the output data of naïve Bayes feature embedding) with SVM classification, while the second version, NB-SVM2, employs both the transformed and original features. Several datasets were used to train and test the system, including NSL-KDD, CICIDS2017, UNSW-NB15, and Kyoto 2006+. The proposed system exhibited excellent performance, achieving over 99 % accuracy and recall in both versions.

Hu et al. [51] introduced a novel technique for IDSs using Multiple Kernel Clustering (MKC). The technique addresses two common challenges in the field: data diversity and incompleteness. To manage these issues, the proposed technique suggests estimating missing attribute values by assessing the similarity of sampled data and generating a kernel matrix from incomplete data, thereby enhancing detection accuracy. Several datasets, including UNSW-NB15, NSL-KDD, and AWID, were used in the study. Compared to other clustering techniques such as Density Peaks, K-means, and Gaussian Mixture Models (GMMs), the proposed model achieved high scores, with a 95.60 % accuracy rate when using the AWID dataset.

Panigrahi et al. [52] proposed a signature-based IDS that utilizes Multi-Objective Evolutionary Feature Selection (MOEFS) for feature selection and hybrid classification techniques, specifically a decision table and naïve Bayes. The system was trained and tested using the CICIDS2017 dataset, enabling it to detect various attacks, such as botnets, port scanning, DoS/DDoS attacks, multiple brute force attacks, and several web attacks. The system's performance was evaluated using various metrics, including recall, accuracy, and precision, and it yielded impressive results: 96.80 %, 96.70 %, and 97.40 %, respectively.

5. Classification of machine learning methods

Machine learning can be bifurcated into several categories delineated by unique characteristics. These categories include supervised learning, semi-supervised learning, unsupervised learning, self-supervised learning (a subset of unsupervised learning), reinforcement learning, and ensemble learning. Each category harnesses distinct methods and techniques, many of which can significantly bolster cybersecurity measures. This paper will explore these machine learning types and their respective techniques, as referenced in various technical publications. A summary of the methods is presented in Table III.

5.1. *Supervised learning*

Supervised learning involves training a model using labeled data to make accurate predictions or decisions when presented with new, unlabelled data. Fig. 3 illustrates an overview of the Supervised Learning methods. The objective of supervised learning models is to classify new data into their correct labels, which can be achieved using a variety of algorithms and techniques. Here are some of the methods employed in supervised learning.

5.1.1. *K-NN*

The K-Nearest Neighbor (K-NN) is a non-parametric supervised learning algorithm predominantly used for classification but can also be applied for regression. It avoids making assumptions about the underlying distribution. Instead, it identifies the 'k' closest training examples in the feature space to the new observation and predicts the target variable based on the majority label or mean, depending on the task—-classification or regression. Several technical papers have utilized the K-NN algorithm, including [38,39], and [46]. In Refs. [38,39], K-NN was used for classification to train and test the proposed IDS models (SENMQTT-SET, MQTT-IoT-IDS2020) on their specific datasets, achieving high performance (99.89 %, 99.9 % accuracy, respectively). In Ref. [46], K-NN was employed in the proposed IDS model for classification using CICIDS 2017–2019 datasets and performed commendably, with an accuracy exceeding 99 %.

5.1.2. *ANN*

The Artificial Neural Network (ANN) is a machine learning algorithm modeled after the structure and functionality of the human brain. The ANN, a deep learning model, consists of interconnected nodes (or artificial neurons) that process input data to produce output signals. ANN can be employed in supervised and unsupervised learning tasks, including classification, clustering, dimensionality reduction, and anomaly detection. In Ref. [36], adversarial methods were deployed using FGSM against binary and multi-class ANNs trained on the Bot-IoT dataset. The results significantly impacted both ANN models regarding detection accuracy and precision. ANN was also utilized for attack classification in Ref. [37] using the KDD'99 dataset, achieving commendable accuracy (91 %), albeit surpassed by other algorithms (such as RF with 94 %).

- *FNN* A Feed-Forward Neural Network (FNN) is a type of ANN where the information flows in one direction, from the input to the output layer, without loops or feedback. FNNs can be used for supervised learning tasks such as classification and regression but also unsupervised learning tasks. The IDS in Ref. [34] used an FNN for binary

Table 3
Machine learning methods summary.

Method	ML type	ML tasks “security”	Advantages	Disadvantages
Decision Tree	Supervised learning	Cyber attacks' classification and regression tasks	• Intuitive knowledge expression •High classification accuracy •Simple implementation	•Information gain bias towards high-level categorical features
RF	Supervised learning	Classification and regression	•Resistance to overfitting •Feature selection isn't required •Low number of control and model parameters •Variance reduction	•Low model interpretability •Performance loss
RFFI	Supervised learning	Feature selection	•Identifying biases •Interpretability	•Model-specific
KNN	Supervised learning	Classification primarily (can be used for regression)	•Simple •Non-parametric algorithm	•Sensitive to neighbourhood order •Feature reduction is often required •Bias towards dominant classes in skewed class distributions
NB	Supervised learning	Classification (can be used for regression)	•It is an online algorithm •Linear time training •Optimal for conditionally independent features •Simple to implement •Arbitrary number of independent features •Fast and efficient	•Doesn't perform well on attribute-related data
SVM	Supervised learning	Classification and Regression	•Good accuracy •Effective in high-dimensional spaces •Global optimum (looks for results that produce higher margin between classes)	•Computationally expensive if the data dimensionality is large
OCSVM	Unsupervised learning (can be semi-supervised)	Anomaly and outliers' detection	•Doesn't require labeled data. •Can handle non-linear data.	•Assumes a single, compact “normal” class
GB	Supervised learning	Classification and regression (Classification for anomaly-based and signature-based)	•Non-parametric •Can handle missing values. •Measures features importance	•Limited Interpretability •Sensitive to outliers
LightGBM	Supervised learning	Classification and regression	•Parallel and GPU support •Categorical feature support	•No support for online learning
J48	Supervised learning	Classification	•Can handle missing values. •Can handle categorical and continuous data	•Sensitive to noise. •Can cause overfitting
CFS	Supervised feature selection technique	Feature selection	•Improve accuracy and interpretability. •Reduce overfitting.	•Scope is limited. •Sensitive to data prepossessing
SAE	Unsupervised learning technique	Feature extraction	•Good for complex and high dimensional data •Dimensionality reduction	•Prone to overfitting •Computationally expensive •Black box model
JRip	Supervised learning	Classification algorithm to minimize the classification error.	•Scalability •Accuracy	•Sensitive to noise
REP Tree	Supervised learning	Classification and regression	•Reduced error pruning “good accuracy” •Interpretability •Efficient for large datasets	•Sensitive to data distribution •Prone to bias and overfitting
MLP	Supervised learning	Attacks and prediction classification, regression, and pattern recognition	•Ability to model non-linear relationships. •Robust and scalable	•Requires hyperparameters tuning
InfoGain	Supervised learning	Feature selection	•Unbiased •Efficient for high-dimensional datasets	•Can't handle datasets with missing values.
OneR	Supervised learning	Classification	•Simple •Interpretability. •Baseline model.	•Prone to overfitting •Limited scalability and feature selection
LR	Supervised learning	Classification	•Easy to construct. •Efficient in training	•Can't deal with non-linear data
Gaussian Function	Mathematical function to model continuous data.	Classification	•Can handle missing data •Performs well with high-dimensional data	•Vulnerable to irrelevant features •Limited expressiveness
Bayesian Network	Supervised and Unsupervised learning	Classification and clustering	•Can handle noisy and missing data. •Can measure the uncertainty in the prediction	•Computationally expensive
SMO	Optimization algorithm (aids in enhancing supervised learning algorithms performance)	Feature selection Clustering Hyperparameter optimization	•Requires few numbers of control parameters (feasible for solving complex optimization problems). •Global search capability	•Sensitivity to initial conditions •Lack of theoretical guarantees
S-DPN	Supervised learning	Classification	•Robust to Adversarial Attacks •Regularization (improve generalization performance and reduce overfitting) •Accuracy	•Complexity •Lack of Interpretability.
Hoeffding Tree	Supervised learning	Classification and regression (Anomaly Detection + attacks attribution)	•Incremental learning (model can be updated instead of retraining from scratch.”	•Sensitive to hyperparameters •Limited expressiveness.

(*continued on next page*)

Table 3 (*continued*)

Method	ML type	ML tasks “security”	Advantages	Disadvantages
			•Online learning •Interpretability	
ANN	Supervised and Unsupervised learning	Classification, clustering, dimensionality reduction, and anomaly detection	•It can model non-linear relations such as EX-OR logic. •Can handle noisy data efficiently.	•Long runtimes during learning
FNN	Supervised learning	Classification and regression	•Parallel processing •Nonlinear Mapping •Universal approximation	•Requires careful selection of the architecture. •Black box model •Overfitting •Training Data Dependencies
k-means	Unsupervised learning	Clustering	•Adapts well to linear data. •Strong interpretability and fast convergence speed	•Sensitive to the initialization condition and the parameter K. •Isn't ideal for nonconvex data
Multi-kernel k-means	Unsupervised learning	Clustering	•Flexibility •Can handle non-linear data •Better clustering performance	•Limited interpretability •Higher computational complexity
PSO	Optimization algorithm	Feature selection, Clustering, Regression	•Robustness •Adaptability •Strong distributed ability •Quick convergence •Can be combined with other algorithms	•The convergence and convergence rate are not proven mathematically yet
GWO	Optimization algorithm	Feature selection, Clustering, Regression	•Versatility and simplicity •Fast convergence •Robustness	•Premature convergence •Limited scalability
AdaBoost	Supervised learning	Classification	•Versatility •Robustness to noise •High accuracy	•Sensitivity to hyperparameters and outliers •Risk of overfitting
MI	Supervised learning	Feature selection	•Captures nonlinear relationships •Robust to feature scaling	•Sensitive to estimation methods.
KOAD	Unsupervised learning	Anomalies and outliers' detection	•Can handle high-dimensional and non-linear data. •Online learning	•Sensitivity to kernel choice
Shannon Entropy	Measure of randomness or “surprise”	Feature selection and Anomaly detection	•Impurity Measure •Clustering Evaluation	•Limitations in Non-Discrete Data •Sensitivity to Data Distribution
Mahalanobis distance	Measure of distance between a point and a distribution	Classification, Clustering, dimensionality reduction, feature selection	•Scale Invariance •Robustness to Multicollinearity	•Sensitivity to Covariance Estimation •Limited Applicability for Non-Linear Relationships
Chi-square test	Statistical test	Feature selection, Dimensionality reduction, Test of independence	•Non-parametric •Identifying associations •Simplicity	•Inability to identify causation •Sensitive to data distribution
ADC	Unsupervised learning	Clustering	•Distance metric flexibility •Robustness •Can handle clusters of arbitrary shapes	•Computational complexity •Sensitivity to parameters
DBSCAN	Unsupervised learning	Clustering	•Handles noise and outliers •Density-based clustering •Minimal parameter tuning	•Sensitivity to parameters •Handling varying densities

and multiclass classifications on the BoT-IoT dataset, achieving high accuracy (99 % for multiclass classification).

– **MLP** A Multilayer Perceptron (MLP) is a type of FNN that contains multiple layers of nodes. These nodes process input signals to generate output signals. Typically implemented for supervised learning tasks, an MLP can also be adapted for unsupervised learning tasks. More generally, MLPs can be used for classification, regression, and pattern recognition tasks, including cyber security applications such as attack classification and prediction. In their IDS [42], used an MLP as a binary classifier for the final stage after feature selection from the AWID dataset using other classifiers in semi-distributed and distributed approaches. Both approaches achieved impressive results (99.97 % and 97.80 %, respectively).
– **SAE** Despite being a type of FNN, a Stacked Autoencoder (SAE) is an unsupervised learning technique composed of numerous layers of autoencoders. These autoencoders are forms of neural networks trained to recreate their input. To create a hierarchical representation of the input data, the output of the encoder in one layer is utilized as input to the following layer. This hierarchical form can be used for classification, regression, or clustering applications. Furthermore, SAE can also be used for feature extraction and dimensionality reduction or as a pre-training step for supervised learning algorithms to improve their performance. In Ref. [42], an SAE was implemented for the feature extraction task for both distributed and semi-distributed models proposed by the authors. The models were designed for impersonation attack detection and were trained on the AWIS dataset. In the proposed models, the feature extraction phase followed data pre-processing and preceded feature selection and classification, with the models achieving promising results.

5.1.3. SVM

A Support Vector Machine (SVM) is a supervised learning algorithm for classification and regression tasks. An SVM classifier separates data points into two or more classes by constructing a hyperplane in high-dimensional space. While primarily performing binary classification, SVM can also handle multiclass classification by training each class against all other classes (i.e., one-vs-all) or constructing a binary classifier for each pair of classes (i.e., one-vs-one).

In [36], adversarial attacks (specifically, label noise attacks) were launched against an SVM IDS model trained on the Bot-IoT dataset. The experimental results suggested that manipulating labels with a high margin significantly impacts SVM's classification performance. Conversely [38], utilized SVM to train an IDS model for attack classification using their proprietary dataset, where the SVM model demonstrated high accuracy (99.73 %).

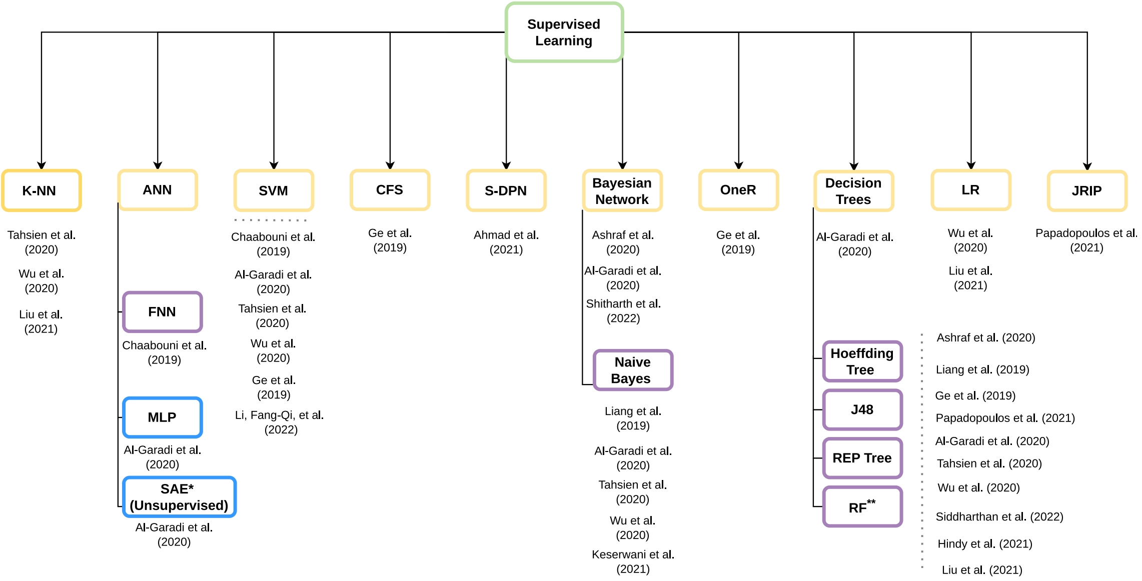

Fig. 3. Supervised Learning methods used by surveyed work to detect Cyber intrusion. *Also considered an unsupervised method. **Also considered an ensemble method.

The SVM classifier was also employed in Ref. [39], utilizing RBF and linear kernels, and it achieved strong results (96.61 %, 98.5 % accuracy) for bidirectional features. Similarly [42], used the SVM method for feature selection and employed the AWID dataset to detect impersonation attacks. The results indicated that SVM delivered strong performance in distributed and semi-distributed IDSs.

Finally, in Ref. [49], SVM was implemented to classify transformed data (i.e., the output of naive Bayes feature embedding) in what can be considered an ensemble learning model. The “NB-SVM” model achieved high accuracy (exceeding 93 %).

5.1.4. Correlation-based feature subset selection (CFS)

Correlation-based feature subset selection (CFS) is a feature selection method used as a preprocessing step to enhance the performance of supervised learning algorithms. CFS capitalizes on the correlation between features and the target variable to pinpoint relevant features. Bagaa et al. (2020) [42] applied CFS in their distributed and semi-distributed models for feature selection, resulting in satisfactory outcomes.

5.1.5. Stacked deep polynomial network (S-DPN)

The Stacked Deep Polynomial Network (S-DPN) is an advanced deep neural network architecture that utilizes polynomial activation functions rather than conventional ones. It is engineered to address common issues associated with deep neural networks, such as vanishing gradients and the complexity of high-dimensional features. S-DPN operates using supervised learning and can be implemented for classification tasks. Ge et al. (2019) [33] utilized S-DPN for binary attack classification in the Intrusion Detection System (IDS), following the extraction of the most pertinent features using Spider Monkey Optimization (SMO) from the NSL-KDD dataset. The S-DPN model achieved an impressive accuracy of 99.02 %.

5.1.6. Bayesian Network (BN)

A Bayesian Network (BN) is a probabilistic graphical model that illustrates a system's potential relationships between variables. Here, the nodes signify variables, and the directed edges between them depict the probabilistic dependencies between these variables. BN can be employed for supervised and unsupervised learning tasks such as classification and clustering. Otoum et al. (2022) [32] used BN for attack classification in their proposed model embedded within an AI-based reaction agent. In contrast, Liu et al. (2021) [37] utilized BN in their proposed IDS for attack classification using the KDD’99 Cup dataset, achieving an accuracy of 90 %. Meanwhile, Zhang et al. (2021) [48] employed Full Bayesian Possibilistic Clustering in their IDS for pattern recognition in IoT traffic. Subsequently, they combined fuzzy decision trees with a sample reweighting scheme to boost accuracy. When tested on their dataset and compared with other models, the results indicated that their model outperformed others, achieving the highest scores.

- *Naïve Bayes* Naïve Bayes (NB) is a type of Bayesian Network that depends on Bayes' theorem, which allows it to calculate the probability of a hypothesis given some observed evidence. Although a Bayesian Network can be implemented for supervised and unsupervised learning, Naïve Bayes (NB) is a supervised learning algorithm because it requires labeled training data. NB is mainly used for classification, but it can also be used for regression. In Ref. [35], NB was used for a classification task to classify data from three aspects: device profiling, attack detection, and attack type. The paper used their dataset, and the NB model achieved good results in the attack detection and attack type (above 90 recall score for both) but not very well in device profiling (above 50 recall) [37]. used NB to train their IDS on the KDD’99 cup dataset for attack classification and the NB model achieved 91 % accuracy [38]. also trained their IDS using their dataset on multiple machine learning methods, including the NB classifier. The results were good but not the best compared to other classifiers (97.8 % accuracy). In Ref. [39], the authors implemented NB in their model for attack classification, and they used their dataset. NB classifier achieved good accuracy (97.55 %) for bidirectional feature classification. For attacks classification [47], didn't use the standard NB in their proposed IDS; instead, they used Likelihood Naïve Bayes (LNB), which is a variant of naïve Bayes that can deal with continuous features by estimating probabilities using Gaussian distribution. In their model, they processed the data in three different stages: clustering, feature selection, and attack detection. LNB was used in the last stage for binary classification. They compared the results of the proposed system that implements LNB with the standard NB classifier, and LNB achieved better accuracy (76.74 % vs. above 80 %) [49]. used NB feature embedding. This technique transforms data from the original feature space to a new one using Naive Bayes probabilities to enhance the features' quality before training the model using the SVM classifier. The model achieved good accuracy (above 93 %). The proposed IDS in Ref. [52] implemented NB with a decision table (DT) as a hybrid technique for attack classification. The CIDS2017 dataset was used for training, and the model achieved 96.80 % accuracy.

5.1.7. LR

Logistic Regression algorithm (LR) is a supervised learning algorithm for binary classification tasks. We can consider LR as a generalized linear model that uses a logistic function to model the probability of the binary outcome [38]. implemented LR for attack classification in their proposed IDS, and the classifier achieved 90.36 % accuracy [39]. also used the LR classifier in their model, and it achieved 99.44 % accuracy for bidirectional feature classification [46]. deployed the LR classifier on selected features for DDoS detection, achieving good accuracy (approximately 94 %).

5.1.8. Decision trees

Decision trees (DT) are supervised machine learning algorithms in classification and regression. DT recursively splits the data into smaller subsets based on the features or attributes of the data until a prediction or decision can be made [37]. implemented the DT algorithm in their IDS for attack classification. The IDS used the KDD'99 cup dataset and the DT model achieved 92 % recall which is higher than most other classifiers used in the experiment. Several types of DT are used for cyber security tasks in the research papers, some of which will be mentioned here.

● *Hoeffding Tree*

Named after the mathematician Wassily Hoeffding, the Hoeffding Tree is one of the DT types that is built incrementally and can handle large datasets and adapt to new arriving data by using a statistical test to determine when a split should be made in the tree instead of examining all the data at once. Hoeffding Tree can be implemented in cyber security for classification and regression tasks such as attack attribution and anomaly detection [32]. deployed the Hoeffding Tree algorithm in their IDS for attack classification, and the model showed stable performance but achieved very low precision for Root (U2R) attacks (11.5 %).

● *J48*

J48 is a widely used decision tree algorithm that implements the C4.5 tree algorithm. Each node in the tree represents a decision based on one or more features. The best features to split are selected using a heuristic approach based on criteria like information gain. The J48 algorithm is used mainly for classification as a supervised learning algorithm. In Ref. [32] J48 and other classifiers were deployed for intrusion detection, and the model was trained on the NSLKDD dataset. The results of J48 were great (above 90 % precision) except for the user-to-root attacks (U2R) as the classifier didn't score good precision (70 %) in this attack [35]. also used J48 for device profiling, binary, and multiclass attacks classification, and the algorithm was the best among the others with very high results (above 97 % P,R,F) [42]. implemented J48 in their semi-distributed IDS for feature selection on the AWID dataset, and the classifier scored 99.87 % recall for impersonation attacks.

● *REP Tree*

Reduced Error Pruning Tree (REP Tree) is a supervised learning algorithm for classification and regression. This algorithm is similar to the C4.5 algorithm but with some modifications to the pruning step as it simplifies the process and improves the generalization ability [43]. deployed REP Tree in their hybrid IDS for binary attack classification and combined it with other classifiers. They trained their model on the CICIDS2017 dataset and BoT-IoT dataset and the hybrid model achieved better results than the results achieved using the regular REP Tree algorithm (both over 90 % recall).

● *RF*

Random Forest (RF) is a supervised learning algorithm for classification and regression tasks. RF is an extension of decision tree algorithms, and it also can be considered ensemble learning because it combines multiple decision trees and aggregates their predictions, resulting in a final prediction [37]. implemented RF in their IDS for attack classification and the classifier achieved the best results among all other used classifiers on the KDD'99 cup dataset (94 % accuracy) [38]. deployed RF in their proposed IDS for attack classification. The model was trained on their dataset, and the RF classifier achieved 100 % accuracy [39]. also implemented RF in their IDS, and the classifier achieved 99.97 % accuracy for bidirectional feature classification [44]. implemented RF as a meta-learning algorithm in their IDS, proposing a novel hybrid technique that combines RF and decision tree algorithms. The authors trained the model and tested it using different datasets like UNSW-NB15, NSL-KDD, and others, and the model achieved good results [45]. deployed RF in their IDS for attack classification after doing feature optimization using two algorithms (i.e., GWO, PSO) to different datasets such as CICIDS-2017 and NSL–KDD. Other classifiers were used for comparison, but RF provided the best accuracy (average of 99.66 %) for all used datasets. Moreover [46], used RF in their IDS model for classification to detect DDoS attacks and the RF classifier achieved high accuracy (above 98 %).

- *RFFI* Random Forest Feature Importance (RFFI) is a method used for determining feature importance in a dataset when building an RF model. The method depends on measuring the decrease in the model's performance when a specific feature is removed or its values are randomized such that the significant drop in the model's performance when changing certain features indicates its high importance and vice versa [46]. used RFFI in their IDS model for feature selection (i.e., Random Forest Feature Importance (RFFI)) with an RF classifier, and the model achieved good accuracy (above 98 %).

5.1.9. OneR

One Rule (OneR) is a simple supervised learning method for classification tasks. OneR provides the most straightforward and precise approach for producing a single rule matching to a single predictor of the data and then selects the most reliable rule as "one rule." In Ref. [42] OneR was implemented in the proposed distributed and semi-distributed IDS for feature selection from the AWID dataset. The classifier scored good accuracy (above 99 %).

5.1.10. JRip

JRip is a classification algorithm used for supervised learning tasks. It is based on the Repeated Incremental Pruning to Produce Error Reduction (RIPPER) algorithm. And it minimizes the classification error by repeatedly pruning the decision rules learned by RIPPER. In Ref. [43], JRip was deployed in their hybrid IDS as a second classifier for multiclass attack classification, and the model achieved good recall (above 90 %).

5.2. Unsupervised learning

In unsupervised learning, the model is provided with unlabelled data and depends on finding patterns or relationships between data. Fig. 4 illustrates an overview of the Unsupervised Learning methods. Two main categories fall under unsupervised learning, namely clustering and dimensionality reduction. In clustering similar instances are grouped based on their features, while the dimensionality reduction goal is to reduce the number of features in the data while retraining as much data as possible.

5.2.1. Clustering

Clustering is the most common unsupervised learning technique in which mathematical, probabilistic, or statistical methods are used to

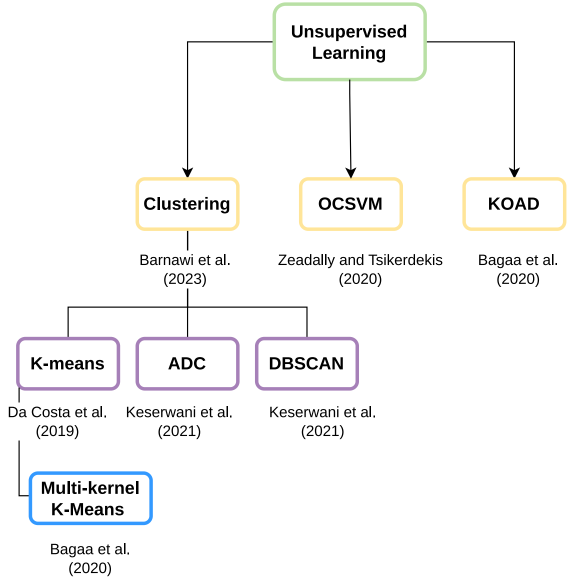

Fig. 4. Unsupervised Learning methods used by surveyed work to detect Cyber intrusion.

group data. It includes different types such as K-means clustering, Multi-kernel k-means, and Density-based clustering. Such techniques are usually implemented for anomaly detection tasks in IoT environments. Sometimes, they are also used as a preliminary step before classification to enhance the overall performance [9].

- *K-means*

The k-means clustering algorithm is an unsupervised learning method used to partition a set of data points into K clusters, where K is a predefined number of clusters. Initially, K data points are selected randomly as clusters' centroids. Then, Euclidean distance is calculated between the data points and centroids to assign the nearest centroid to each data point. This leads to frequent change in the centroids and the algorithm stops when there is no more change or if there is a predefined maximum number of iterations. Although unsupervised, this method still needs human involvement for output interpretation. Also, such methods could detect zero-day attacks by putting unknown patterns in separate clusters [26].

- *Multi-kernel K-means*

Multi-kernel k-means can be considered an extension of the k-means clustering algorithm with some differences such as the ability to handle non-linearly separable data and complex data distribution by facilitating multiple kernel functions [51]. implemented multi-kernel K-means clustering in their IDS to solve the data diversity and incompleteness issues. The authors tested their model on different datasets such as AWID and UNSW_NB15 and the model achieved better accuracy than the normal K-mean model (93.80 % vs. 84.20 %).

- *ADC*

Anticipated Distance-based Clustering (ADC) is a clustering algorithm that groups similar data points based on their similarity (measured by anticipated distance). ADC falls under unsupervised learning techniques such as other clustering algorithms [47]. deployed ADC and another clustering algorithm (i.e., DBSCAN) in their IDS.

- *DBSCAN*

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is an unsupervised clustering algorithm that groups data points based on their density. DBSCAN is beneficial for finding clusters of arbitrary shapes and identifying noise in datasets. In Ref. [47], DBSCAN and ADC were implemented for clustering as a predecessor phase to feature selection and classification to enhance the classification's accuracy. The proposed model achieved good accuracy (99.99 %).

- *OCSVM*

One-class SVM (OCSVM) is an extension of the SVM algorithm. However, it is an unsupervised (or semi-supervised) learning method that is used for detecting anomalies or outliers in datasets. The primary objective of OCSVM is to identify the smallest hyperplane or hypersphere that encloses most of the data points, effectively separating normal data from possible anomalies. This is accomplished by training the model using only normal or non-outlier instances without needing labeled anomaly examples. Once trained, the OCSVM can identify new instances that fall outside the learned decision boundary as potential anomalies [40]. integrated OCSVM (for anomaly detection) with other machine learning techniques like PSO and LightGBM in their proposed IDS. The proposed model scored good accuracy on the UNSW-NB15 dataset (86.68 %).

- *KOAD*

Kernel-based Online Anomaly Detection (KOAD) is an unsupervised learning algorithm that depends on kernel functions to map the input data and its underlying structure to high-dimensional space, allowing the capture of complex patterns and relationships. The main task of the algorithm is detecting anomalies, and its ability to adapt to changes and update its internal representation continuously makes it suitable for real-time applications [51]. used an enhanced version of the KOAD algorithm (E-KOAD) in their proposed IDS merged with other methods in a novel algorithm to detect DDoS attacks. In the proposed algorithm, E-KOAD was used specifically to identify the suspicious points. The model was trained using the CICIDS2017 dataset, achieving a good accuracy of 99.55 %

5.3. *Ensemble learning*

Ensemble learning is a technique for machine learning that integrates the predictions of different models to increase the final prediction's accuracy and resilience. Fig. 5 illustrates an overview of the Ensemble Learning methods. By utilizing several models' variety and complementary qualities, ensemble learning aims to attain superior performance than any single model. Examples of ensemble learning techniques are RF (mentioned above) and Gradient Boosting.

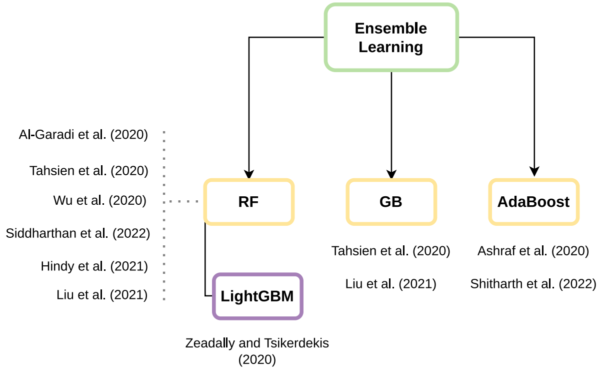

Fig. 5. Ensemble Learning methods used by surveyed work to detect Cyber intrusion.

5.3.1. GB

Gradient Boosting (GB) is an ensemble learning technique that combines multiple weak models to create a robust ensemble mode. The objective of gradient boosting is to reduce the overall error of the ensemble model by training each weak model on the mistakes of the prior model. GB can be considered a supervised learning technique because it requires labeled training data to learn and make predictions. Hence, it can be used for classification and regression tasks [38,46]. implemented the GB algorithm in their IDS for classification and the classifiers achieved good accuracy (100 %, above 98 %).

- *LightGBM*

Light Gradient Boosting Machine (LightGBM) is an open-source high-performance GB framework developed by Microsoft. As an extension of the GB algorithm, LightGBM follows GB principles in addition to several optimizations and features for performance, speed, and memory usage enhancement. Several techniques are used for this enhancement such as leaf-wise tree growth, GB one-side sampling (GOSS), and exclusive feature bundling (EFB) [40]. used LightGBM with PSO for double-dimension reduction and feature extraction, aiming to reduce the size of the dataset.

5.3.2. AdaBoost

Adaptive Boosting (AdaBoost) is an ensemble learning algorithm that falls in the supervised learning category as it requires labeled data for classification tasks. AdaBoost algorithm iteratively trains a series of weak classifiers on different weighted versions of the training data and the weights of the misclassified instances are increased after each iteration. The purpose of this process is to force the next weak classifier to focus more on the instances that are harder to classify. Doing so forms a robust final classifier representing the weighted combination of the weak classifiers. In Ref. [32] AdaBoost was implemented in a distributed classification system (for cyber-attacks on IoT) for merging the models obtained by JRip algorithm and the AdaBoost model achieved 99.8793 % precision [48]. also deployed AdaBoost to combine base fuzzy decision trees with online learning to accommodate the concept drift in their IDS model. The model scored good accuracy when used for their dataset (82 %, real source traffic).

5.4. Optimization algorithms

Optimization algorithms are mathematical techniques used to determine the optimal solution to a problem, often by minimizing or optimizing an objective function. Optimization algorithms are used in machine learning to alter the parameters of a model to enhance its performance on a particular task. Usually, they are used for feature selection. Several optimization algorithms, such as PSO, GWO, and SMO, were inspired by animals' behavior.

5.4.1. PSO

Particle Swarm Optimization (PSO) is a common optimization technique inspired by the social behavior of flocks of birds or schools of fish. It is a stochastic optimization method that mimics the movement of a collection of particles across a high-dimensional search space, where each particle represents a potential solution to an optimization problem. PSO can be used for supervised and unsupervised learning tasks such as feature selection, classification, clustering, and regression [40]. used PSO and a Light Gradient Boosting Machine (LightGBM) in their proposed IDS for feature extraction from the UNSW-NB15 dataset. The reason behind this combination is to avoid the problem of uneven distribution of large-scale datasets. The model's accuracy was better than all other methods (86.68 %) [45]. also implemented PSO in their IDS for feature selection together with another optimization algorithm (GWO) and validated it using different datasets such as NSLKDD and CICIDS-2017. Their model achieved an accuracy of 99.66 % for multiclass classification.

5.4.2. GWO

Grey Wolf Optimization (GWO) is a metaheuristic population-based optimization algorithm inspired by grey wolves' social hierarchy and hunting behavior in the wild. 2014 marked its introduction by Mirjalili et al. Each wolf symbolizes a possible solution to the optimization issue. GWO can be used in clustering, feature selection, and in conjunction with classifiers to optimize their performance. In Ref. [45], the combination of GWO and PSO as hybrid GWO–PSO has been employed to enhance feature subset selection.

5.4.3. SMO

Spider Monkey Optimization (SMO) is a metaheuristic optimization method inspired by spider monkeys' social behaviour and foraging technique. Mirjalili et al. introduced SMO in 2015, which has been applied to various optimization issues. Like other metaheuristic optimization methods, SMO does not require previous knowledge of the issue area and may be used in various optimization situations. The method is based on the social behavior of spider monkeys, in which they collaborate and coordinate their food hunt by following the most successful monkey. In Ref. [33] SMO was utilized in the proposed IDS for the feature selection phase before using S-DPN for classification to achieve higher accuracy. The NSL-KDD dataset was used to validate the model, and the results were very promising with an accuracy of 99.02 %.

5.5. Functions and measures

5.5.1. Gaussian function

A Gaussian function, usually called a Gaussian or normal distribution, is a mathematical function defining a continuous probability distribution. It was introduced by the mathematician Carl Friedrich Gauss in the early 19th century and bears his name. In machine learning and data analysis, the Gaussian function can model the probability distribution of data and estimate the likelihood of different outcomes [39]. used the Gaussian function to model the probability distribution in the Gaussian Naïve Bayes classifier in their proposed IDS. The classifier scored 97.55 % accuracy for bidirectional features.

5.5.2. InfoGain

Information gain is a statistic used in decision trees and other machine learning methods to assess the amount of information supplied by a feature or attribute in a dataset regarding the target variable (i.e., the output variable). To maximize the information gain and minimize the data's entropy (i.e., uncertainty), it is used to pick the optimum feature or attribute for dividing the data in a decision tree. In decision tree algorithms such as C4.5 and ID3, the InfoGain function is frequently used to pick the optimal feature or characteristic for dividing the data at each tree node. It is also utilized by other machine learning methods, such as Random Forest and Gradient Boosting, to choose the best features or characteristics for model construction. In Ref. [42], InfoGain was deployed in the proposed IDS to evaluate the relevance of any given feature to the class label and this feature selection method scored above 94 % accuracy.

5.5.3. MI

Mutual Information (MI) is a measure used in supervised learning tasks, often for feature selection. MI helps to identify the most informative features for a particular target variable by quantifying the dependence or statistical relationship between random variables [46]. deployed MI and RFFI in their IDS for feature selection before feeding the selected features to several classifiers, including RF, GB, WVE, KNN, and LR. The proposed model achieved good accuracy (99.997 %).

5.5.4. Shannon entropy

Shannon entropy is a fundamental concept in information theory, named after Claude E. Shannon (an American mathematician and electrical engineer). The entropy value represents a measure of uncertainty or randomness in a single random sample where the correlation between the entropy value and uncertainty level is negative [51]. used Shannon entropy in their proposed IDS algorithm to utilize feature construction. For the proposed system to detect DDoS attacks, the entropy will either increase or drop significantly in DDoS attacks, and the value will remain uniform when the traffic is normal. Other measures are also combined within the proposed algorithm, such as E-KOAD (mentioned earlier), which identifies suspicious points after calculating Shannon entropy.

5.5.5. Mahalanobis distance

Named after the Indian statistician Prasanta Chandra Mahalanobis, Mahalanobis distance measures the distance between a point and a distribution. By taking into account the shape and orientation of the distribution, the Mahalanobis distance can distinguish between points that are genuinely distant from the mean and those that appear distant due to the shape or orientation of the distribution [51]. have utilized Mahalanobis distance in their proposed anomaly detection algorithm to measure the distance between suspicious points and the distribution of dictionary members (which represent normal behavior).

5.5.6. Chi-square test

The chi-square test is a statistical test used to indicate a significant association between two categorical variables in a sample by comparing observed frequencies (actual) and expected frequencies (under the assumption of no association between variables). As mentioned above [51], have deployed multiple measures in their proposed anomaly detection algorithm. A tsome point in the algorithm, the Chi-square test is used to examine the significance of the calculated Mahalanobis distance. The significant difference in the test indications implies that the suspicious vector is likely an anomaly or DDoS attack.

5.6. Other categories

5.6.1. Semi-supervised learning

Semi-supervised learning lies between supervised and unsupervised learning so that it can use labeled and unlabelled data for training. This approach is beneficial in real-world applications because obtaining only labeled data is infeasible, and unlabelled data are abundantly available. The ability to benefit from labeled and unlabelled data improves the learning process and performs better than using only one. To use unlabelled data, multiple techniques are used such as bootstrapping, self-training, co-training, and graph-based methods to find the underlying patterns and relationships between data. Semi-supervised learning can be used for different tasks such as classification, anomaly detection, and natural language processing (NLP).

5.6.2. Self-supervised learning

Self-supervised learning is a subcategory of unsupervised learning because it doesn't rely on labeled data for training. In contrast, it uses unlabelled data to learn useful representations that can be used later for different supervised or unsupervised tasks such as clustering, classification, and regression. In self-supervised learning, pseudo-labels or “targets” are generated from the data by leveraging its inherent structure or properties using several techniques including autoencoders, predictive learning, and contrastive learning. There are many applications for self-supervised learning like anomaly detection and computer vision, but one of the most prominent and impactful applications is in natural language processing (NLP).

5.6.3. Reinforcement learning

Reinforcement learning (RL) is a type of machine learning where the model learns to make decisions by interacting with its environment in a trial-and-error approach. The model's actions either lead to “reward” or “penalty” feedback from the environment. The main target for the model is to learn a “policy” that maximizes the cumulative reward value over time by continuously updating the policy and optimizing its actions based on the environment's feedback. RL is better if the optimal solution is not obvious or difficult to determine using traditional supervised learning techniques. Different algorithms are used for RL such as Q-learning and Deep Q-Networks and others.

6. Challenges

Implementing machine learning for detecting cyber threats in the IoT environment poses several significant challenges.

1) **Data Complexity:** The foremost challenge arises from the nature of data generated by IoT devices. This data is often overwhelming in volume, variety, and velocity, posing a substantial challenge to traditional ML techniques. Therefore, there's a pressing need for efficient and scalable algorithms capable of handling this complexity.
2) **Device Heterogeneity:** The diversity in IoT devices, encompassing various architectures, protocols, and operating systems, forms a second major hurdle. This heterogeneity makes it difficult to develop universally applicable solutions, as each device may require a tailored approach.
3) **Dynamic Nature of Threats:** Cyber threats constantly evolve, necessitating continuous learning and adaptation in threat detection methods. This dynamic nature often requires manual intervention, which can be a limiting factor in the efficiency of these systems.

While unsupervised learning methods offer a partial solution by identifying patterns and anomalies without data labeling, they are unreliable. These techniques may struggle to ascertain the significance of their findings, often necessitating manual analysis and interpretation. Another concern is privacy. Using sensitive user data for training ML models raises privacy issues, necessitating the development of secure techniques that protect user data while ensuring efficient threat detection. Deep learning methods might address some of these issues but are not a panacea. Given that many IoT devices have limited computational resources, the resource-intensive nature of DL methods introduces additional complications.

Alternate approaches like transfer learning, federated learning, and edge computing offer promising avenues to mitigate some challenges. However, they do not present complete solutions to all problems. A critical aspect often overlooked is the security of ML models themselves. Adversaries may exploit vulnerabilities in these models, making it imperative to develop robust methods against adversarial attacks and manipulation.

In summary, while ML presents a viable path for enhancing IoT cyber threat detection, it is fraught with challenges that require innovative solutions and ongoing research to ensure effectiveness, security, and privacy.

7. Future vision with generative AI and LLMs

A language language model (LLM) represents a category of AI algorithms employing deep learning methods and extensive data collections to interpret, summarize, produce, and forecast novel content. The concept of generative AI is intimately linked to LLMs. These models are a specialized form of generative AI, explicitly designed to facilitate the creation of text-based material. The future of IoT security, powered by Generative AI and large language models (LLMs), promises a more secure, intelligent, and adaptive approach to protecting the ever-expanding universe of IoT devices and networks. The key will be to harness these advanced technologies responsibly, balancing innovation with ethical considerations [3].

7.1. Generative AI and LLMs

The current landscape of LLMs is diverse and rapidly evolving, reflecting significant advancements in natural language processing and AI research. Among the notable models, BERT [53], introduced by Google in 2018, stands out for its transformer-based design and ability to convert data sequences, influencing Google search enhancements. Claude, developed by Anthropic, is notable for its focus on constitutional AI, ensuring outputs are helpful, harmless, and accurate.

Other notable models include Ernie by Baidu, notable for its Mandarin proficiency and extensive user base; Falcon 40B/180b [54] from the Technology Innovation Institute, an open-source model available on Amazon SageMaker. OpenAI's GPT series is particularly influential, with GPT-3 introducing a massive scale of 175 billion parameters in 2020 and becoming exclusively available through Microsoft. GPT-3.5, an upgrade with fewer parameters, powers ChatGPT and was integrated into Bing search. The latest GPT-4, released in 2023, is a multimodal model capable of handling language and images.

Llama [55], Meta's model released in 2023, is notable for its open-source availability and variety in model sizes. Orca by Microsoft demonstrates efficiency with fewer parameters. Palm from Google specializes in complex reasoning tasks across various fields. Smaller, specialized models like Phi-1 from Microsoft focus on quality over quantity in data training. Stability AI's StableLM series aims for transparency and accessibility. Finally, Vicuna 33B, trained by fine-tuning LLaMA, is an influential open-source model with a smaller parameter count but effective capabilities.

7.2. Cyber threat detection

Generative AI is set to revolutionize IoT security through advanced threat detection techniques. These models can use deep learning and natural language processing (NLP) to analyze unstructured data from various sources, including IoT device logs and network traffic. For example, an LLM can parse through gigabytes of log data from a smart home system to identify unusual patterns that might indicate a cyber-attack, such as a sudden spike in outbound data suggesting data exfiltration.

7.3. Reinforced encryption and authentication

In the realm of encryption and authentication, LLMs can aid in developing more sophisticated protocols tailored to IoT devices' unique constraints. For example, an LLM can optimize lightweight encryption algorithms for low-power IoT sensors, ensuring secure data transmission without overburdening the device's limited resources.

7.4. Blockchain-based systems

Generative AI can significantly enhance the security of blockchain-based systems for IoT in several ways. LLMs can assist in automatically generating and verifying smart contracts. By analyzing and understanding the nuances of contract language, LLMs can identify potential vulnerabilities or logic flaws in smart contracts before they are deployed on the blockchain. In addition, in threat detection, LLMs can be trained to monitor and analyze blockchain transactions and IoT communication patterns to detect anomalous or potentially malicious activity. By leveraging their vast knowledge base and pattern recognition capabilities, LLMs can alert system administrators to suspicious behavior, enabling rapid response to security threats. Finally, as mentioned before, LLMs can contribute to developing more robust encryption and authentication protocols for IoT devices interacting with blockchain networks, enhancing overall system security.

7.5. Access control

Generative AI can significantly enhance access control in IoT devices through their advanced natural language understanding and contextual analysis capabilities [56]. LLMs allow for intuitive user interactions with IoT systems by processing and interpreting natural language commands. They can analyze usage patterns and context to make informed access decisions, bolstering security through anomaly detection for potential unauthorized access [57]. Furthermore, integrating LLMs with biometric systems like voice recognition can add layer of secure user authentication. These models can also automate and refine access control policies by learning from user behaviors and preferences, ensuring a personalized and efficient user experience [58]. However, implementing LLMs in this domain requires careful attention to data privacy, security, and the potential for model manipulation [59].

7.6. IoT software security

Generative AI can be applied to enhance IoT software security through advanced vulnerability detection. They excel in static code analysis, identifying common and complex vulnerabilities such as buffer overflows and SQL injections in IoT software by analyzing code patterns and anomalies [60,61]. LLMs streamline the detection process, allowing quicker and more thorough software audits, which is crucial in the dynamic IoT environment. Continuously updated with the latest security research, LLMs adapt to new threats, ensuring ongoing protection. This integration of LLMs in vulnerability detection offers a robust solution for maintaining the security of increasingly complex IoT software systems.

7.7. Penetration testing

Generative AI can significantly enhance penetration testing in IoT networks by automating and optimizing various process aspects. These AI tools can generate realistic phishing emails and social engineering content, simulate sophisticated cyber-attacks to test network defenses, and even predict potential vulnerabilities by analyzing code or system configurations [62]. They can also assist in generating custom scripts for testing applications and networks and, in some cases, use natural language processing to interpret and analyze the results of penetration tests, offering insights and recommendations for strengthening security. Moreover, LLMs can be trained on the latest cybersecurity trends and exploits, ensuring that the penetration tests are up-to-date and cover a wide range of potential threats.

8. Conclusion

IoT devices have become fundamental to everyday life, offering increased connectivity and convenience. However, this accelerated development in IoT devices also brought several security challenges that need to be addressed to guarantee the safety and reliability of these interconnected systems. The survey was organized into five sections detailing the role of machine learning in enhancing IoT security. We have provided an overview of the current trends in ML for cyber threat detection in IoT environments. Furthermore, we survey recent cyber detection methods, define them, highlight the approach, and detail the attack surface utilized along with their evaluations. The utilized ML techniques are also discussed, defined, and compared regarding advantages and drawbacks in the relevant use cases. Moreover, open issues were discussed briefly, concluding that further research and development is required as no current solution can address the issues related to IoT cyber threat detection. A holistic strategy combining diverse techniques and strategies is required as a final recommendation. We present this survey as a reference for the current advancements in the field and point out the direction being taken. Our goal is for the survey to create a more secure and resilient IoT environment by identifying the current issues, understanding their root causes and possible implications, and developing innovative solutions.

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.

References

[1] J. Piqueira, B. Mishra, Understanding Cyber Threats and Attacks, 2020, p. 10.
[2] B. Kaur, S. Dadkhah, F. Shoeleh, E.C.P. Neto, P. Xiong, S. Iqbal, P. Lamontagne, S. Ray, A.A. Ghorbani, Internet of Things (Iot) Security Dataset Evolution: Challenges and Future Directions, Internet of Things, 2023 100780.
[3] M.A. Ferrag, O. Friha, B. Kantarci, N. Tihanyi, L. Cordeiro, M. Debbah, D. Hamouda, M. Al-Hawawreh, K.-K.R. Choo, Edge learning for 6g-enabled internet of things: a comprehensive survey of vulnerabilities, datasets, and defenses, IEEE Communications Surveys & Tutorials 25 (4) (2023) 2654–2713.
[4] A. Barnawi, S. Gaba, A. Alphy, A. Jabbari, I. Budhiraja, V. Kumar, N. Kumar, A systematic analysis of deep learning methods and potential attacks in internet-of-things surfaces, Neural Comput. Appl. (2023) 1–16.
[5] A. Abusitta, G.H. de Carvalho, O.A. Wahab, T. Halabi, B.C. Fung, S. Al Mamoori, Deep Learning-Enabled Anomaly Detection for Iot Systems, vol. 21, Internet of Things, 2023 100656.
[6] R. Alghamdi, M. Bellaiche, A cascaded federated deep learning based framework for detecting wormhole attacks in iot networks, Comput. Secur. 125 (2023) 103014.
[7] D. Thakur, J.K. Saini, S. Srinivasan, Deepthink iot: the strength of deep learning in internet of things, Artif. Intell. Rev. (2023) 1–68.
[8] K.A. da Costa, J.P. Papa, C.O. Lisboa, R. Munoz, V.H.C. de Albuquerque, Internet of things: a survey on machine learning-based intrusion detection approaches, Comput. Network. 151 (2019) 147–157.
[9] F. Liang, W.G. Hatcher, W. Liao, W. Gao, W. Yu, Machine learning for security and the internet of things: the good, the bad, and the ugly, IEEE Access 7 (2019) 158 126–158 147.
[10] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, P. Faruki, Network intrusion detection for IoT security based on learning techniques, IEEE Communications Surveys & Tutorials 21 (3) (2019) 2671–2701.
[11] M.A. Al-Garadi, A. Mohamed, A.K. Al-Ali, X. Du, I. Ali, M. Guizani, A survey of machine and deep learning methods for internet of things (IoT) security, IEEE Communications Surveys & Tutorials 22 (3) (2020) 1646–1685.
[12] S.M. Tahsien, H. Karimipour, P. Spachos, Machine learning based solutions for security of internet of things (IoT): a survey, J. Netw. Comput. Appl. 161 (2020) 102630.
[13] H. Wu, H. Han, X. Wang, S. Sun, Research on artificial intelligence enhancing internet of things security: a survey, IEEE Access 8 (2020) 153 826–153 848.
[14] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, A. Wahab, A review of intrusion detection systems using machine and deep learning in internet of things: challenges, solutions and future directions, Electronics 9 (7) (2020).
[15] F. Hussain, R. Hussain, S.A. Hassan, E. Hossain, Machine learning in IoT security: current solutions and future challenges, IEEE Communications Surveys & Tutorials 22 (3) (2020) 1686–1721.
[16] A. Thakkar, R. Lohiya, A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges, Arch. Comput. Methods Eng. 28 (2021) 3211–3243.
[17] R. Ahmad, I. Alsmadi, Machine Learning Approaches to IoT Security: A Systematic Literature Review, vol. 14, Internet of Things, 2021 100365.
[18] A. Aldhaheri, F. Alwahedi, M.A. Ferrag, A. Battah, Deep Learning for Cyber Threat Detection in Iot Networks: A Review, Internet of Things and Cyber-Physical Systems, 2023.
[19] C. Alex, G. Creado, W. Almobaideen, O.A. Alghanam, M. Saadeh, A Comprehensive Survey for Iot Security Datasets Taxonomy, Classification and Machine Learning Mechanisms, Computers & Security, 2023 103283.
[20] Y.R. Siwakoti, M. Bhurtel, D.B. Rawat, A. Oest, R. Johnson, Advances in Iot Security: Vulnerabilities, Enabled Criminal Services, Attacks and Countermeasures, IEEE Internet of Things Journal, 2023.
[21] S. Mathur, A. Kalla, G. Gür, M.K. Bohra, M. Liyanage, A survey on role of blockchain for iot: applications and technical aspects, Comput. Network. 227 (2023) 109726.
[22] W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, Z. Tari, Blockchain-based federated learning for securing internet of things: a comprehensive survey, ACM Comput. Surv. 55 (9) (2023) 1–43.
[23] S.W. Turner, M. Karakus, E. Guler, S. Uludag, A Promising Integration of Sdn and Blockchain for Iot Networks: A Survey, IEEE Access, 2023.
[24] H. Ahmadvand, C. Lal, H. Hemmati, M. Sookhak, M. Conti, Privacy-preserving and Security in Sdn-Based Iot: A Survey, IEEE Access, 2023.
[25] T.A. Ahanger, A. Aljumah, M. Atiquzzaman, State-of-the-art survey of artificial intelligent techniques for iot security, Comput. Network. 206 (2022) 108771.
[26] S. Zeadally, M. Tsikerdekis, Securing internet of things (IoT) with machine learning, Int. J. Commun. Syst. 33 (1) (2020) e4169, e4169 dac.4169.
[27] S.H. Mekala, Z. Baig, A. Anwar, S. Zeadally, Cybersecurity for industrial iot (iiot): threats, countermeasures, challenges and future directions, Comput. Commun. 208 (2023) 294–320.
[28] B. Huang, D. Chaki, A. Bouguettaya, K.-Y. Lam, A survey on conflict detection in iot-based smart homes, ACM Comput. Surv. 56 (5) (2023) 1–40.
[29] X. Wang, H. Zhu, Z. Ning, L. Guo, Y. Zhang, Blockchain Intelligence for Internet of Vehicles: Challenges and Solutions, IEEE Communications Surveys & Tutorials, 2023.
[30] E. Manavalan, K. Jayakrishna, A review of internet of things (iot) embedded sustainable supply chain for industry 4.0 requirements, Comput. Ind. Eng. 127 (2019) 925–953.
[31] R. Toorajipour, V. Sohrabpour, A. Nazarpour, P. Oghazi, M. Fischl, Artificial intelligence in supply chain management: a systematic literature review, J. Bus. Res. 122 (2021) 502–517.
[32] Y. Otoum, D. Liu, A. Nayak, DL-IDS: a deep learning–based intrusion detection framework for securing IoT, Transactions on Emerging Telecommunications Technologies 33 (3) (2022) e3803, e3803 ett.3803.
[33] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, A. Robles-Kelly, Deep learning-based intrusion detection for IoT networks, in: 2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC), 2019, pp. 256–265, 609.
[34] P. Papadopoulos, O. Thornewill von Essen, N. Pitropakis, C. Chrysoulas, A. Mylonas, W.J. Buchanan, Launching adversarial attacks against network intrusion detection systems for IoT, Journal of Cybersecurity and Privacy 1 (2) (2021) 252–273.
[35] H. Siddharthan, T. Deepa, P. Chandhar, SENMQTT-SET: an intelligent intrusion detection in IoT-MQTT networks using ensemble multi cascade features, IEEE Access 10 (2022) 33095–33110.
[36] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, X. Bellekens, Machine learning based iot intrusion detection system: an mqtt case study (mqtt-iot-ids2020 dataset), in: B. Ghita, S. Shiaeles (Eds.), Selected Papers from the 12th International Networking Conference, Springer International Publishing, 2021, pp. 73–84, plus 0.5em minus 0.4emCham.
[37] J. Liu, D. Yang, M. Lian, M. Li, Research on intrusion detection based on particle Swarm optimization in IoT, IEEE Access 9 (2021) 38254–38268.
[38] P.K. Keserwani, M.C. Govil, E.S. Pilli, P. Govil, A smart anomaly-based intrusion detection system for the internet of things (IoT) network using GWO–PSO–RF model, Journal of Reliable Intelligent Environments 7 (2021) 3–21.
[39] S. Shitharth, P.R. Kshirsagar, P.K. Balachandran, K.H. Alyoubi, A.O. Khadidos, An innovative perceptual Pigeon galvanized optimization (PPGO) based likelihood naïve Bayes (LNB) classification approach for network intrusion detection system, IEEE Access 10 (2022) 46 424–446 441.
[40] F.-Q. Li, R.-J. Zhao, S.-L. Wang, L.-B. Chen, A.W.-C. Liew, W. Ding, Online intrusion detection for internet of things systems with Full bayesian possibilistic clustering and ensembled fuzzy classifiers, IEEE Trans. Fuzzy Syst. 30 (11) (2022) 4605–4617.
[41] S. Daneshgadeh Çakmakçı, T. Kemmerich, T. Ahmed, N. Baykal, Online DDoS attack detection using Mahalanobis distance and kernel-based learning algorithm, J. Netw. Comput. Appl. 168 (2020) 102756.
[42] M. Bagaa, T. Taleb, J.B. Bernabe, A. Skarmeta, A machine learning security framework for iot systems, IEEE Access 8 (2020) 114 066–114 077.
[43] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, P. Burnap, A supervised intrusion detection system for smart home IoT devices, IEEE Internet Things J. 6 (5) (2019) 9042–9053.
[44] H. Alqahtani, I.H. Sarker, A. Kalim, S.M. Minhaz Hossain, S. Ikhlaq, S. Hossain, Cyber intrusion detection using machine learning classification techniques, in: Computing Science, Communication and Security: First International Conference, COMS2 2020, Gujarat, India, March 26–27, 2020, Revised Selected Papers 1, plus 0.5em minus 0.4emSpringer, 2020, pp. 121–131.
[45] I.H. Sarker, Y.B. Abushark, F. Alsolami, A.I. Khan, IntruDTree: a machine learning based cyber security intrusion detection model, Symmetry 12 (5) (2020).
[46] M.A. Rahman, A.T. Asyhari, L. Leong, G. Satrya, M. Hai Tao, M. Zolkipli, Scalable machine learning-based intrusion detection system for IoT-enabled smart cities, Sustain. Cities Soc. 61 (2020) 102324.
[47] M.A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, H. Janicke, RDTIDS: rules and decision tree-based intrusion detection system for internet-of-things networks, Future Internet 12 (3) (2020).
[48] H. Zhang, J.-L. Li, X.-M. Liu, C. Dong, Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection, Future Generat. Comput. Syst. 122 (2021) 130–143.
[49] M. Alduailij, Q.W. Khan, M. Tahir, M. Sardaraz, M. Alduailij, F. Malik, Machine-learning-based DDoS attack detection using mutual information and random forest feature importance method, Symmetry 14 (6) (2022).
[50] J. Gu, S. Lu, An effective intrusion detection approach using SVM with naïve Bayes feature embedding, Comput. Secur. 103 (2021) 102158.
[51] N. Hu, Z. Tian, H. Lu, X. Du, M. Guizani, A multiple-kernel clustering based intrusion detection scheme for 5G and IoT networks, International Journal of Machine Learning and Cybernetics (2021) 1–16.
[52] R. Panigrahi, S. Borah, M. Pramanik, A.K. Bhoi, P. Barsocchi, S.R. Nayak, W. Alnumay, Intrusion detection in cyber–physical environment using hybrid naïve bayes—decision table and multi-objective evolutionary feature selection, Comput. Commun. 188 (2022) 133–144.
[53] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018 arXiv preprint arXiv:1810.04805.
[54] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, É. Goffinet, D. Hesslow, J. Launay, Q. Malartic, et al., The Falcon Series of Open Language Models, 2023 arXiv preprint arXiv:2311.16867.
[55] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and Efficient Foundation Language Models, 2023 arXiv preprint arXiv:2302.13971.

[56] A. Maatouk, N. Piovesan, F. Ayed, A. De Domenico, M. Debbah, Large Language Models for Telecom: Forthcoming Impact on the Industry, 2023 arXiv preprint arXiv:2308.06013.

[57] A. Maatouk, F. Ayed, N. Piovesan, A. De Domenico, M. Debbah, Z.-Q. Luo, Teleqna: A Benchmark Dataset to Assess Large Language Models Telecommunications Knowledge, 2023 arXiv preprint arXiv:2310.15051.

[58] L. Bariah, Q. Zhao, H. Zou, Y. Tian, F. Bader, M. Debbah, Large Language Models for Telecom: the Next Big Thing?, 2023 arXiv preprint arXiv:2306.10249.

[59] H. Zou, Q. Zhao, L. Bariah, M. Bennis, M. Debbah, Wireless Multi-Agent Generative Ai: from Connected Intelligence to Collective Intelligence, 2023 arXiv preprint arXiv:2307.02757.

[60] M.A. Ferrag, A. Battah, N. Tihanyi, M. Debbah, T. Lestable, L.C. Cordeiro, Securefalcon: the Next Cyber Reasoning System for Cyber Security, 2023 arXiv preprint arXiv:2307.06616.

[61] N. Tihanyi, T. Bisztray, R. Jain, M.A. Ferrag, L.C. Cordeiro, V. Mavroeidis, The formai dataset: generative ai in software security through the lens of formal verification, PROMISE'23 (2023) 33.

[62] A. Happe, J. Cito, Getting pwn'd by ai: penetration testing with large language models, in: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2023, pp. 2082–2086.