Indonesian Journal of Electrical Engineering and Computer Science
Vol. 34, No. 3, June 2024, pp. 1840~1857
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v34.i3.pp1840-1857

Uncovering botnets in IoT sensor networks: a hybrid self-organizing maps approach

Mwaffaq Abu AlHija, Hamza Jehad Alqudah, Hiba Dar-Othman
Department Networks and Cyber Security, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan

Article Info

Article history:

Received Jan 21, 2024
Revised Feb 15, 2024
Accepted Feb 27, 2024

Keywords:

Deep learning architectures
AIDS
Detecting botnets
Hybrid self-organizing maps
IoT

ABSTRACT

The integration of the internet of things (IoT) has revolutionized diverse industries, introducing interconnected devices and IoT sensor networks for improved data acquisition. However, this connectivity exposes IoT ecosystems to emerging threats, with botnets posing significant risks to security. This research aims to develop an innovative solution for detecting botnets in IoT sensor networks. Leveraging insights from existing research, the study focuses on designing a hybrid self-organization map (SOM) Approach that integrates lightweight deep learning (DL) techniques. The objective is to enhance detection accuracy by exploring various DL architectures. Proposed methodology aims to balance computational efficiency for resource-constrained IoT devices while improving the discriminatory power of the detection system. The study advancing IoT cybersecurity and addresses critical challenges in botnet detection within IoT sensor networks. The testing of the artificial neural networks (ANN) classifier involves three models, each represented based on parameters related to the construction of the training models. The most effective ANN achieves 86%, works on anomaly intrusion detection systems (AIDS).

Corresponding Author:

Mwaffaq Abu AlHija
Department Networks and Cyber Security, Faculty of Information Technology
Al-Ahliyya Amman University
Amman, Jordan
Email: m.abualhija@ammanu.edu.jo

1. INTRODUCTION

The rapid integration of the internet of things (IoT) has ushered in an era of unparalleled connectivity, enabling diverse devices to interconnect seamlessly and contribute to the fabric of our interconnected world. Within this expansive network, IoT sensor networks play a pivotal role across various industries, facilitating data acquisition and informed decision-making. However, this pervasive integration has concurrently exposed these networks to a growing threat landscape, with botnets emerging as a significant security concern [1]. Botnets, networks of compromised devices manipulated by malicious actors, pose a profound risk to the integrity and functionality of IoT sensor networks. As these networks evolve and expand, the imperative to address the challenges associated with botnet detection becomes increasingly critical. Traditional detection methods, especially those rooted in deep learning (DL), often face constraints when applied to the resource-limited environment of IoT sensor devices.

The research delves into the complexities of botnet detection within IoT sensor networks, aiming to bridge existing research gaps and address pertinent challenges. The detection mechanisms must not only be accurate but also resource-efficient, adaptable to the dynamic nature of evolving botnets, and capable of real-time responsiveness crucial for preventing or mitigating potential damages. Moreover, scalability is imperative to accommodate the large-scale deployments characteristic of IoT networks. Integrating these detection mechanisms seamlessly into the diverse and evolving network architectures of IoT poses an additional layer of complexity [2].

In response to these challenges, this study proposes a hybrid self-organizing maps (SOM) approach integrated with a lightweight DL framework. This innovative methodology strives to offer a comprehensive solution for uncovering botnets in IoT sensor networks, effectively balancing the intricacies of accuracy, resource efficiency, adaptability, real-time responsiveness, scalability, and seamless integration within the dynamic landscape of IoT cybersecurity. Through this research, we seek to contribute to the enhancement of the security and resilience of critical infrastructures in the ever-evolving IoT ecosystem. Botnets, orchestrated networks of compromised devices manipulated by malicious entities, pose substantial risks to the security and operational integrity of IoT sensor networks. The evolution of the internet of Things has been transformative, fueled by advancements in network theory, architectural innovations, and notable progress in sensor technology and microprocessors.

Noteworthy statistics from Gartner indicate a substantial 21% increase in deployed IoT endpoints, totaling an impressive 5.8 billion devices by [1]. This surge underscores the transformative impact of IoT across industries, shaping everyday life in various ways. Moreover, the IoT market, valued at $190 billion in 2018, is projected to experience a remarkable surge, reaching an astonishing $1102.6 billion by 2026, reflecting an impressive compound annual growth rate (CAGR) of 24.7% [2]. Key sectors such as banking and financial services, information technology, telecommunications, healthcare, and government applications contribute significantly to this unprecedented growth. Despite the explosive expansion of IoT promising interconnectedness across billions of devices, it brings forth a unique set of challenges, particularly in the realms of security and privacy. These challenges necessitate careful consideration to ensure the sustainable and secure proliferation of IoT adoption. A previous study introduced a service-oriented architecture (SOA) tailored for the broader IoT [3], laying the groundwork for the comprehensive exploration undertaken in this paper. Building upon prior research, the paper conducts a detailed examination of the constituent layers within the SOA architecture: the sensing layer, network layer, service layer, and interface layer. The sensing layer establishes intricate connections with hardware components, facilitating data collection on the status and conditions of diverse IoT entities. The network layer, as the foundational infrastructure, enables seamless wireless or wired connectivity among these entities. At the core of the architecture, the service layer encompasses essential elements such as service discovery, composition, management, and interfaces, empowering developers to efficiently fulfill end-user requests. Simultaneously, the interface layer governs methods and mechanisms for user or application interactions, ensuring effective and user-friendly communication. IoT devices often face vulnerabilities, primarily stemming from their constrained resources, making them attractive targets for cyberattacks. The vulnerability of interconnected billions of these devices became evident during a coordinated attack on a domain name provider, leading to a widespread denial of service (DoS) attack that affected numerous well-known sites, including GitHub, Twitter, and others (Dyn [4]). It’s worth noting that many of the devices involved in the attack were compromised by the Mirai botnet, primarily because of their persistent use of default usernames and passwords.

The paper commences the threat modelling process by presenting a generic smart home use case [5]. In this scenario, an array of IoT devices can effortlessly connect to an IoT gateway and be supervised through a central server, illustrated in Figure 1. The entire use case is methodically segmented into five distinct zones, with each playing a crucial role in the smart home ecosystem [6]. These zones comprise:

- IoT device zone: this zone encompasses all IoT sensing and actuating devices strategically deployed within the smart home environment. These devices function as the sensory and operational components of the system, establishing connectivity with the cloud zone through the IoT field gateway, as depicted in Figure 1.
- IoT field gateway zone: functioning as an intermediary, the IoT field gateway facilitates communication between the IoT device zone and the broader ecosystem. It operates as a bridge for the transfer of data and commands between the devices and the central control unit.
- Server zone: the server zone forms the core of the smart home use case, consisting of multiple components crafted to monitor and control all IoT devices within the IoT device zone. These components bear the responsibility of overseeing device operations and responding to user requests.
- Cloud gateway zone: within the cloud zone, it is essential to highlight the cloud gateway zone, which orchestrates communication between the IoT device zone and the consumer zone. Serving as a conduit for data exchange, it ensures seamless connectivity.
- Consumer zone: the consumer zone serves as the interface through which end-users interact with the smart home system, incorporating various user interface devices like tablets, cell phones, and similar gadgets. Consumers can employ these devices to access real-time status information for each IoT device

and issue commands to the Azure components, thus controlling the on/off functionality of the IoT devices [7].

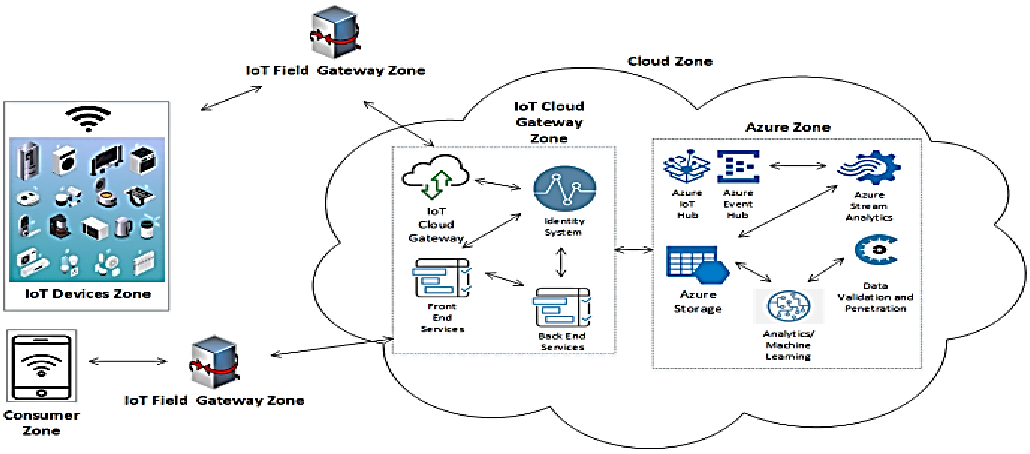

Figure 1. Smart home use case IoT overview [6]

Amaral [7] presented the architecture of an IoT system and emphasized the set of features that a system must possess to be classified as an IoT system [7]. The primary features encompass:

- Interconnection of objects: in this context, "thing" refers to a smart object capable of collecting, generating, processing, and storing data from the user or application perspective.
- Connectivity: the IoT ensures internet connectivity for elements within the system, encompassing devices, applications, and crucial IoT infrastructures.
- Unique identification of objects: IoT devices must possess unique identifiability.
- Sensing/actuation capability: the pivotal component in sensing the environment involves an intelligent sensor that can gather and transmit data from the surroundings to the IoT systems. Simultaneously, an actuator can perform specific operations based on commands received from the IoT system.
- Embedded intelligence: the incorporation of artificial intelligence advancements is essential to be embedded into edge IoT systems.
- Interoperable communication capability: an IoT system should possess the ability to communicate using standardized and interoperable communication protocols.
- Self-configurability: given the heterogeneous nature of connected devices in an IoT system, it is inherent that IoT devices may require self-management and configuration. This spans from software and hardware management to resource allocation. Programmability or Software defined: physical devices within IoT systems can be readily customized with a user's command or software-defined functions, all without the need for physical changes.

Botnets have evolved significantly to exploit vulnerabilities within IoT sensor networks, rendering these networks a prime target for cybercriminals. Their susceptibility stems from heterogeneity, low-power constraints, and distributed deployment, making them challenging to secure effectively. Consequently, traditional signature-based intrusion detection systems (IDS) must be improved for detecting emerging botnet threats. In response to the challenge, researchers have turned to advanced techniques rooted in machine learning (ML) and DL to enhance security measures within IoT sensor networks [6].

Several studies have made significant contributions to the field, shedding light on various aspects of botnet detection in IoT environments. Williams *et al.* [8] proposed a "Lightweight DL framework" that employs a hybrid SOM approach for botnet detection in IoT sensor networks. The framework introduces a novel method for effectively identifying malicious activities within these networks, emphasizing lightweight computational requirements to align with IoT device constraints.

Yulianto *et al.* [9] provided a comprehensive overview of deep cybersecurity, emphasizing the role of neural networks and DL in enhancing IoT security. The work highlights the potential of DL techniques in addressing botnet threats within the IoT context. Incorporating multi-access edge computing and ML methods. Their research emphasizes leveraging ML techniques to protect IoT sensor networks from botnet attacks. Vinayakumar *et al.* [10] thoroughly reviewed intrusion detection approaches in IoT systems.

Khan *et al.* [11] examined botnet detection techniques in a survey, offering valuable insights into classification, methods, and evaluation strategies. The study provides a thorough understanding of the current state-of-the-art approaches in botnet detection, assisting researchers and practitioners in making well-informed decisions. Tuan *et al.* [12] conducted a review on botnet attack detection within software-defined networking (SDN)-enabled IoT, with a specific focus on ML techniques. Their research delves into the integration of SDN and ML to bolster security in IoT sensor networks. Proposed an intrusion detection framework tailored to energy-constrained IoT devices, addressing the unique challenges such devices face in protecting against botnet attacks. These studies underscore the critical importance of developing innovative botnet detection approaches for IoT sensor networks. They emphasize the role of lightweight DL frameworks, ML methods, and tailored solutions to mitigate the evolving threat landscape in the domain. Furthermore, they highlight the need for adaptable and resource-efficient techniques to safeguard the integrity and functionality of IoT sensor networks in the face of persistent botnet threats.

In light of the background, the research paper presents a novel hybrid approach utilizing SOM and a lightweight deep-learning framework to uncover botnets in IoT sensor networks, building upon and extending the existing body of knowledge in the field. Botnets have evolved to exploit vulnerabilities within IoT sensor networks, making them a prime target for cyber adversaries. These networks are particularly susceptible due to their heterogeneous nature, low-power constraints, and distributed deployment. Traditional signature-based IDSs fall short in identifying emerging botnet threats. Researchers have turned to advanced techniques rooted in ML and DL to bolster security measures within IoT sensor networks. Several studies have made noteworthy contributions to the field, shedding light on various aspects of botnet detection in IoT environments. For instance, Khan *et al.* [13] introduced a feature engineering approach based on fuzzy logic for botnet detection using artificial neural networks (ANN), highlighting the potential of hybrid methods in enhancing detection accuracy.

Joshi *et al.* [14] addressed the challenge of botnet traffic identification using neural networks, demonstrating the effectiveness of ANN-based techniques in distinguishing botnet traffic from legitimate IoT network activity. Biswas and Roy [15] presented research on IoT security, focusing on botnet detection in IoT through ML approaches. Their work underscores the significance of utilizing ML for safeguarding IoT sensor networks against botnet threats. Investigated a ML-centric strategy for identifying IoT-botnet attacks utilizing a sequential architecture. The emphasis was placed on the significance of adaptive detection mechanisms to counter the evolving strategies employed by botnets within IoT environments [15].

Together, these investigations emphasize the vital importance of formulating innovative botnet detection approaches specifically designed for the distinctive characteristics of IoT sensor networks. They highlight the potential of hybrid methodologies, neural networks, and ML techniques in mitigating the evolving threat landscape. Furthermore, they emphasize the need for adaptable and resource-efficient solutions to safeguard the integrity and functionality of IoT sensor networks against persistent botnet threats [16].

Integration with IoT architectures: the proposed solution seeks seamless integration with diverse IoT network architectures. It aligns with the evolving landscape of IoT deployments and ensures that the detection mechanism can coexist harmoniously with existing IoT ecosystems. Experimental validation: the paper contributes to the field by conducting rigorous experimentation and evaluation to validate the effectiveness of the proposed hybrid approach. Benchmark datasets and commonly accepted metrics are used to assess the performance of the botnet detection system [17].

Knowledge synthesis: by synthesizing insights from previous works in IoT security and botnet detection, the paper contributes to the collective knowledge in the field. It builds upon and extends the existing body of research to offer a comprehensive solution tailored to the unique challenges of IoT sensor networks [18]. The research paper offers substantial contributions to IoT cybersecurity, particularly within botnet detection in IoT sensor networks. The paper’s key contributions can be succinctly summarized as follows: innovative hybrid approach: the paper introduces an inventive hybrid methodology that amalgamates SOM with a lightweight DL framework for botnet detection. The novel approach harnesses the inherent strengths of SOMs in clustering and data visualization while simultaneously ensuring computational efficiency, a crucial aspect for IoT devices with limited resources.

While misuse detection depends on predefined characteristics of known attacks to identify threats, anomaly detection is the only approach designed to unveil new and previously unknown threats. The fundamental principle that underlies anomaly detection entails the development of a model that characterizes normal system behavior. Any deviations from these established norms are flagged as potential attacks, encompassing various anomaly detection techniques. SOM [19], [20] have become popular for effectively identifying abnormal patterns in network data. SOM belongs to unsupervised ANNs specializing in visual clustering. Its primary purpose is to map data from a high-dimensional space into a lower-dimensional space, constructing a topological representation that mirrors the data distribution from the original high-dimensional domain SOM is achieved by grouping akin data vectors and presenting them in a graphical format [21], [22].

The methodology employed in this research, titled is designed to systematically address the complex challenge of botnet detection within the resource-constrained environment of IoT sensor networks. The research adopts a multifaceted approach, commencing with a comprehensive review of existing literature to establish a foundational understanding of IoT sensor networks, botnets, and prevalent detection techniques. This literature review informs the subsequent development of the proposed methodology. The core of the methodology lies in the design and implementation of a Hybrid SOM approach integrated with a lightweight DL framework. The SOM model, renowned for its clustering and visualization capabilities, is adapted to the specific context of IoT sensor networks. The integration with a lightweight DL framework ensures computational efficiency, addressing the resource constraints inherent in IoT devices.

Data collection involves sourcing diverse datasets representative of IoT sensor network behaviors and botnet activities. The collected data undergoes meticulous preprocessing to ensure relevance and accuracy. The methodology incorporates supervised and unsupervised learning techniques for the training of the detection model, leveraging DL algorithms to enhance the discriminatory power of the system. Evaluation metrics are carefully chosen to assess the effectiveness of the proposed approach, considering factors such as accuracy, adaptability to evolving botnets, and real-time responsiveness. Rigorous experimental design and testing procedures are employed to validate the performance of the hybrid SOM approach in comparison to existing detection methods [23]. The methodology aims to provide a comprehensive and innovative solution for uncovering botnets in IoT sensor networks, contributing to the advancement of cybersecurity in the dynamic landscape of IoT.

2. LITERATURE REVIEW

2.1. IoT security and IoT sensor networks challenges

The era of digitization has brought numerous advantages in the field of computing. Consequently, nations have steadily heightened their dependence on digitization, software-defined networks (SDN), and the IoT [24], [25]. However, digital transformation has also presented a significant challenge: cybersecurity. Cyber-attacks can disrupt critical systems and cause irreversible damage to economies. These attacks entail a broad spectrum of consequences, including but not limited to the following:

- Electrical blackouts: cyber-attacks can compromise the integrity of power grids, leading to widespread electrical blackouts [26].
- Financial institution theft: financial institutions are susceptible to theft and fraud, jeopardizing the financial sector’s stability [27].
- Theft of sensitive data: valuable and confidential data, such as medical records, can be stolen, leading to privacy breaches and identity theft.
- Network disruption: cyber-attacks can potentially disrupt phone and computer networks, paralyzing essential communication systems [28].

An IoT network is composed of physical objects or “things” equipped with limited computational, storage, and communication capabilities. These objects are embedded with various electronic components, including sensors and actuators, and feature software functionalities along with network connectivity. This enables them to collect, occasionally process, and exchange data [29], [30]. IoT has found extensive applications and has significantly benefited various sectors and services, including critical infrastructure management, agriculture, military, smart grids and cities, and personal healthcare.

2.2. IoT-security

The proliferation of devices and the increasing sophistication of hacking tools have given rise to an unprecedented surge in security breaches. Emerging technologies, such as the public cloud, the IoT, and artificial intelligence, have introduced formidable challenges to conventional security practices [31]. IoT, in particular, has heralded a transformative era of connectivity, enabling a spectrum of secure, intelligent services. IoT applications encompass various domains, ranging from intelligent health monitoring and traffic management to creating smart cities, efficient waste disposal, streamlined logistics, responsive emergency services, and precise industrial and retail control systems. IoT has engendered a comprehensive and ubiquitous network of smart devices.

An intrusion prevention system (IPS) takes a proactive stance in countering security attacks by closely examining data patterns, often in network traffic, and promptly identifying deviations from established data records or signatures. Upon detecting an attack, the IPS promptly blocks the malicious data. Typically, it functions as the second line of defense within a security infrastructure, working with an IDS. However, IDS grapples with specific challenges regarding ensuring secure access control [32].

On the contrary, when integrated with a firewall and an IDS, an IPS establishes a resilient defense mechanism capable of detecting and actively preventing attacks within a secure network environment. To further boost its effectiveness, artificial intelligence (AI) technologies, including ML, natural language processing (NLP), and neural networks (NN), can be utilized. These AI technologies enable the rapid identification and mitigation of the impact of attacks, delivering daily alerts and facilitating the operation of an intelligent intrusion detection and prevention system (IDPS) [33].

Nonetheless, intricate IoT devices are susceptible to potential security threats, primarily due to factors like hostile interfaces, development on unregulated platforms, and vulnerabilities inherent in individual components integrated into the network [33]. The landscape's inherent diversity, along with interoperability and accessibility challenges, often results in suboptimal security monitoring within IoT networks. An IDS serves as a valuable solution to address security concerns and mitigate the impact of cyberattacks. IDS plays a crucial role in network and host system security management. It identifies intrusions or instances of misuse within the network or system and promptly reports them to administrators while maintaining a record for subsequent investigations. IDS is skilled at managing suspicious events without disrupting regular activities, even in the face of a malicious outbreak [32].

Numerous tools and techniques are available to combat the threat posed by these attacks. Strong firewall protection is imperative, particularly as conventional firewalls may struggle to discern abnormal or anomalous attack behavior. Traditional antivirus software has limitations in identifying new virus patterns effectively. Intrusion detection, while adept at alerting upon the entry of an attack into the network, often needs to catch up in actively preventing the attack. Present-day IDS systems have limitations, notably flexibility and scalability [33]. The core detection system operates on two primary principles: behavior analysis or pattern recognition, followed by the prevention system, which utilizes a signature mechanism to monitor suspicious network traffic. It proactively blocks both inbound and outbound packets before they can access other network resources. The IPS serves as an integrated component that combines robust firewall protection with multi-layer support and detection functionality [34], [35].

2.3. IoT sensor networks: a brief overview

The ubiquity of IoT sensor networks has revolutionized how we interact with the environment, connecting devices and collecting data on an unprecedented scale. The section delves into the latest developments in IoT sensor networks, examining their characteristics, applications, and the emerging challenges they pose from a security perspective. Recent references provide insights into the evolving landscape of IoT sensor networks and the need for robust security measures to protect them.

Characteristics of IoT sensor networks IoT sensor networks are characterized by their diverse and dynamic nature. Recent research highlights vital characteristics and heterogeneity; IoT networks comprise many devices with varying capabilities, communication protocols, and energy constraints. Heterogeneity complicates standardization and interoperability efforts [36].

Scalability: IoT deployments often involve massive numbers of devices, necessitating scalable solutions to manage and secure these extensive networks [35]. Resource constraints: many IoT devices operate with limited computational resources, requiring efficient security mechanisms that operate within these constraints [36]. Dynamic behavior: IoT networks exhibit dynamic behaviors, with devices joining, leaving, or reconfiguring themselves continuously. Dynamic nature demands adaptive security measures [37]. Data sensitivity: IoT devices collect and transmit sensitive data, making data privacy and confidentiality paramount concerns [38] applications and deployment of IoT sensor networks.

IoT sensor networks find applications across diverse domains. Recent references shed light on their applications and deployment:

- Smart cities: IoT sensors are pivotal in creating smart cities, enabling efficient resource management, traffic control, and environmental monitoring [39].
- Healthcare: in healthcare, IoT devices monitor patients, transmit vital data, and enable remote diagnostics and treatment [40].
- Agriculture: IoT sensors assist in precision agriculture, optimizing irrigation, and crop management [41].
- Environmental monitoring: IoT sensors are crucial in monitoring air quality, climate, and wildlife tracking, aiding conservation efforts [42]. Real-time data for predictive maintenance and quality control [43].
  a. Industrial automation: IoT networks enhance industrial processes by providing.
  b. Security challenges in IoT sensor networks: as IoT sensor networks continue to expand, so do the associated security challenges. Recent references provide insights into these challenges:
  c. Botnet threats in IoT: botnets formed by compromised IoT devices are a significant threat. Recent studies emphasize their role in DDoS attacks, data exfiltration, and malware propagation.

The evolving landscape of IoT sensor networks presents both opportunities and challenges. While these networks offer unprecedented capabilities in various domains, their security challenges continue to expand. Recent references illuminate the intricate interplay between IoT sensor networks' characteristics, applications, and the need for robust security measures. Addressing these challenges requires ongoing research and innovation to ensure IoT sensor networks' continued growth and security in a connected world [34]. The proliferation of IoT sensor networks has ushered in an era of unprecedented data collection and connectivity across various domains, from smart cities to healthcare and agriculture. These networks are characterized by many resource-constrained devices [35], each equipped with sensors and communication modules, working together to collect and transmit data from the physical world to centralized or distributed processing units. Recent advancements have expanded the applications of IoT sensor networks, making them integral to modern society. The critical attributes of IoT sensor networks include:

- Large-scale deployment: IoT sensor networks are often deployed extensively, involving thousands to millions of devices. Managing and securing such vast deployments pose logistical and security challenges.
- Resource constraints: many IoT devices operate with limited processing power, memory, and energy resources. These resource constraints dictate the complexity of security mechanisms that can be implemented on these devices.
- Dynamic behavior: IoT networks exhibit dynamic behaviors, with devices constantly joining, leaving, or reconfiguring themselves. Dynamic nature necessitates adaptive security measures to accommodate changes in network topology.
- Data sensitivity: IoT devices collect and transmit sensitive data, ranging from personal health information to industrial process data. Protecting data privacy and ensuring confidentiality are paramount concerns.
- Security challenges in IoT sensor networks: the characteristics of IoT sensor networks give rise to many security challenges, some of which have evolved. Recent references shed light on these challenges and provide insights into emerging trends [36].

2.3.1. IoT sensor networks wireless sensor nodes (WSN)

IoT sensor networks comprise WSNs, which are small devices equipped to sense the environment and perform small computations [8]. These devices form a WSN that collaboratively senses and responds to the environment [8]. They communicate with IoT-enabled devices like routers, providing access to the broader infrastructure for data retrieval and processing. WSNs consist of sensor nodes and sink nodes, with sink nodes serving as data collection hubs and WSN gateways [9]. Users can directly observe the IoT-based WSN via a local IP-based network or remotely over the Internet, and they can send commands via sink nodes [9]. Sensor nodes are composed of four fundamental components: the power unit, the sensor, the processor, and the radio. The sensor is responsible for measuring environmental variables, while the radio handles communication. The processor organizes tasks and converts data into signals for transmission, and the power unit consists of the node's battery pack [44]. Additionally, the processor manages sleep cycles, which nodes utilize to conserve power. Typically, the most energy-consuming function is the exchange of data, with the energy requirement increasing exponentially as the data needs to travel further [45].

Therefore, careful consideration must be given to node density and deployment patterns. The IoT stack is structured in a manner like the TCP/IP stack, featuring five horizontal layers that define end-to-end communication from the physical medium (layer 1) up to the application (layer 5). It also incorporates additional vertical 'planes' that represent processes requiring management at each layer. These include (a) power (i.e., the sharing of power between node functions), (b) mobility (i.e., the tracking of nodes), and (c) tasks (i.e., communication, message detection, and sensing activities). Protocols at each layer need to address these three vertical processes.

2.3.2. Botnets in IoT environments

The proliferation of IoT devices has brought numerous benefits to various industries but has also introduced new security challenges. One of the significant threats in IoT environments is the emergence of botnets, which can compromise the security and privacy of IoT systems. Section reviews the existing literature on botnets in IoT environments, highlighting recent research, trends, and key findings. Please provide an overview of what botnets are in IoT environments, discussing their characteristics and specific challenges compared to traditional botnets.

- Traditional IDS: discuss traditional IDS approaches and their limitations in detecting IoT-based botnets. Mention recent studies that have explored the use of traditional IDS in IoT environments [46].
- ML-based detection: explore the application of ML techniques for botnet detection in IoT. Highlight supervised, unsupervised, and reinforcement learning methods [47].
- Behavioral analysis and anomaly detection: discuss research focusing on behavioral analysis and anomaly detection to identify botnet activity in IoT networks [48].

- Block chain for botnet detection: discuss recent developments in using block chain technology for enhancing botnet detection and security in IoT environments [49].
- Real-world botnet attacks in IoT: present real-world examples of botnet attacks in IoT environments, including the Mirai botnet and its variants. Discuss the impact of these attacks and the lessons learned.
- IoT security frameworks and solutions: explore recent research on proposed security frameworks and solutions designed to protect IoT environments from botnets [50].

Summarize the key points discussed in the related work section, emphasizing the current state of research on botnets in IoT environments and the significance of addressing threats.

2.3.3. IoT botnets

The term “botnet” is a combination of “robot” and “network.” A bot refers to a program capable of autonomously executing user-centric tasks without requiring direct user interaction [51]. Botnets are broadly categorized into two primary architectures based on how the bots are controlled: centralized client-server and decentralized peer-to-peer (P2P) approaches. In a centralized client-server setup, the bot master controls and supervises all bots from a single central point through command and control (C&C) commands. Conversely, in a decentralized mode, each bot acts as a server and a client, facilitating the distribution and receipt of commands [52]. In the realm of networks, IoT botnets become integral components, manipulating computing devices maliciously through three critical operations. The first operation involves identifying vulnerable devices through scanning. Subsequently, a suitable bot, aligned with the architecture of the vulnerable device, is installed in a process known as propagation. Finally, an attack is initiated through command-and-control operations, enabling the botnet to carry out its malicious activities [53]. For instance, the Mirai botnet comprises various components, including attack vectors, a scanner process actively seeking other devices to compromise, and the command-and-control infrastructure that controls the compromised devices (bots), facilitating further propagation and initiating attacks [54].

An IoT botnet constitutes a network of devices, including cameras, routers, DVRs, wearables, and other embedded devices that have been infected with malware [54]. Notable examples of botnet malware attacks impacting IoT include Mirai and BASHLITE. These botnets are designed to launch various cyberattacks. One of the most common attacks is identity theft, where the bot-infected code on a device collects sensitive user information and transmits it to the bot master. Additionally, infected devices are employed in spam attacks to generate and send fraudulent emails. In key-logging attacks, user inputs are recorded and transmitted to the bot master.

2.4. ML and DL in IoT security

Within the domain of AI, numerous powerful ML techniques have emerged and are widely employed in data mining. These techniques empower the system to glean valuable structural patterns and models from the training dataset [55]. The ML process typically comprises two pivotal phases: (1) the training phase and (2) the decision-making phase (as illustrated in subsequently, in the decision-making phase, the system can generate estimated outputs based on the trained model when presented with new input data [56]. There exist four primary types of ML algorithms: supervised learning, reinforcement learning, unsupervised learning, and semi-supervised learning. The subsequent subsections delve into the commonly employed ML techniques [56], [57].

2.4.1. Supervised learning

Supervised learning involves algorithms acquiring the capability to predict outcomes for unknown cases based on learned representations from labelled input data. Examples of such learning methods include the support vector machine (SVM) and random forest techniques. SVM is particularly well-suited for classification-related problems, while the random forest approach is utilized for problems involving both classification and regression [58]. Within the network intrusion detection system (NIDS) research, SVM stands out as one of the most frequently employed algorithms, owing to its practical computational efficiency and robust classification capabilities. SVM algorithms are highly suitable for handling high-dimensional data, although selecting an appropriate kernel function can be challenging. It’s worth noting that SVMs demand substantial memory and processing resources, making them resource-intensive [59]. A practical supervised ML approach uses the random forest algorithm in scenarios involving imbalanced data. However, random forest algorithms may encounter the issue of overfitting, where the model becomes excessively tailored to the training data, potentially leading to reduced generalization performance [60].

2.4.2. Unsupervised learning

Unsupervised learning algorithms specialize in acquiring representations from unlabeled input data. The primary objective of unsupervised learning is to anticipate unknown cases by uncovering the inherent structures or patterns within the data [61]. Two key examples of unsupervised learning methods are principal component analysis (PCA) and SOM. PCA is frequently employed in feature reduction techniques, while SOM is instrumental in clustering approaches. PCA expedites feature learning, and many researchers utilize it for feature selection before classification [61].

In the context of anomaly detection, clustering algorithms like k-means and other distance-based algorithms play a pivotal role. In NIDS, SOM, an ANN, is utilized to reduce payload size. During anomaly detection, clustering algorithms are often subjected to initial conditions, such as generating a high false positive rate and centroids, which can pose challenges for these algorithms. Among unsupervised learning algorithms, k-means hold significant importance. provides an overview of commonly used ML approaches.

The term “self-organizing feature map” is an alternate name for SOM, which is regarded as one of the most significant neural network models for unsupervised learning. SOM serves two primary objectives: data clustering and dimensionality reduction. Typically, a SOM comprises the input layer and the map layer. During data clustering with SOM, the total number of neurons in the map layer corresponds to the desired number of clusters. Each neuron is assigned a weight vector, and the SOM algorithm effectively addresses the data clustering task. For readers interested in a more detailed exploration of the SOM algorithm, references [61] provide comprehensive discussions.

2.4.3. SOM in intrusion detection

Intrusion detection within IoT sensor networks is an evolving challenge, with traditional methods often needing help keeping up with botnet attacks’ dynamic nature. To address this issue, researchers have turned to advanced techniques like SOM as a powerful tool for intrusion detection in these complex environments. Self-organizing maps, also known as Kohonen maps, represent a type of ANN primarily employed for unsupervised learning. SOMs are especially well-suited for intrusion detection in IoT sensor networks due to their unique capabilities [62].

At the core, SOMs are designed to map high-dimensional data into a lower-dimensional space while preserving the topological relationships present in the original data. They do so by grouping similar data vectors, creating a topological representation that mirrors the data distribution. SOMs have garnered attention in the field of intrusion detection for several reasons:

- Unsupervised learning: SOMs are particularly advantageous in intrusion detection because they operate without needing labeled training data. This is a significant advantage in the context of IoT sensor networks, where data can be diverse and rapidly changing.
- Clustering and visualization: SOMs are adept at clustering data vectors based on similarity. Capability is invaluable for identifying anomalies and potential intrusion attempts within network traffic data.
- Topological representation: the topological maps created by SOMs offer a visual representation of network traffic data, making it easier to identify patterns, outliers, and potential threats.
- Anomaly detection: SOMs excel at anomaly detection, crucial for identifying previously unknown threats. By establishing a model of normal behavior, any data significantly deviating from the model can be flagged as suspicious.
- Handling high-dimensional data: in the context of network traffic data, which can be high-dimensional and complex, SOMs offer an effective solution for dimensionality reduction and pattern recognition [63], [64].
- Recent research has focused on enhancing the effectiveness of SOMs in intrusion detection within IoT sensor networks. These advances include:
  a. Hybrid approaches: researchers have explored hybrid models that combine SOMs with other ML or DL techniques to improve detection accuracy and reduce false positives.
  b. Real-time detection: efforts have been made to adapt SOMs for real-time intrusion detection, allowing quicker responses to emerging threats.
  c. Scalability: as IoT networks grow, scalability has become a key consideration. Recent work has looked into scalable SOM implementations that can handle large volumes of data.

2.4.4. Challenges and future directions

While SOMs show promise in IoT intrusion detection, challenges remain, such as:

- Configuration and tuning: configuring SOMs for specific IoT environments and fine-tuning their parameters can be complex.
- Resource requirements: training and running SOMs can be computationally intensive, demanding significant processing power and memory [65].
- Interpretability: interpreting SOM results and determining the significance of identified clusters can be challenging.

Self-organizing maps offer a valuable approach to intrusion detection in IoT sensor networks, particularly for identifying botnet threats. Their unsupervised learning capabilities, clustering and visualization strengths, and topological representations make them ill-suited for these networks' dynamic and complex nature. However, further research and development are needed to address challenges and enhance their practicality in real-world IoT security scenarios.

2.5. Previous approaches to botnet detection

2.5.1. NIDS

IDS play a crucial role in identifying and mitigating threats within computer networks. They monitor network packets to detect internal and external malicious and abnormal activities [66]. However, the effectiveness of IDS can be challenged by the sheer volume of network traffic and the diverse distribution of data. IDS monitors various information sources, including networks and individual computers, with the primary objective of reporting any unauthorized access or suspicious activities. It collects data from multiple network sources and systems, subjecting data to analysis to identify potential threats and attacks. summarizes the deployment environments and detection techniques employed by IDS. IDS implementation encompasses various techniques and methods, which can be broadly categorized into the following groups:

- ML-based methods: ML techniques train IDS to recognize patterns associated with different types of attacks. ML models can learn from historical data to detect deviations from normal network behavior, signaling potential intrusions.
- Data mining methods: data mining techniques are applied to IDS to uncover hidden patterns or anomalies within large datasets. These methods can help identify unusual network behavior indicative of attacks.
- Statistical techniques: statistical approaches establish baseline network behavior and detect deviations from the norm. Statistical analysis helps in flagging activities that fall outside expected patterns.

IDS implementations can exhibit significant variations, ranging from tiered monitoring systems to dedicated antivirus software designed to track network traffic across an entire network. These implementations can be further categorized into specific classes to address various aspects of intrusion detection.

- It is crucial to emphasize that selecting the appropriate IDS implementation and technique hinges upon the unique security demands and prerequisites of the particular network or system under consideration. The examination of incoming network traffic is undertaken by the system, commonly referred to as a NIDS [67].
- The system monitors critical operating system files and is designated a "Host-based intrusion detection system (HIDS)."

These previously mentioned categorizations of IDS are subject to further classification, with signature and anomaly detection as fundamental bases for commonly utilized variations. Signature-based IDS are designed to identify potential threats by scrutinizing specific patterns, such as ll-established intrusion sequences used by Trojans or distinct byte sequences present in network traffic. Nomenclature was borrowed from antivirus software, which initially referred to these detected patterns as "signatures." Signature-based IDS excels at detecting known attacks but encounters challenges when recognizing new attacks that do not exhibit discernible patterns [68].

2.5.2. Anomaly-based detection

Anomaly-based detection is an innovative technique developed to address the challenge of identifying unknown attacks, often stemming from the proliferation of malware. This approach leverages ML to build a reliable model of typical system behavior. Subsequently, the behavior of the newly generated model is compared to the genuine model. While the method can effectively detect unknown attacks, it does carry the risk of producing false positives, where legitimate activities may be erroneously classified as malicious [69].

In a specific study [70], an algorithm tailored for an anomaly-based system was introduced, known as the AdaBoost algorithm. This algorithm incorporates two distinct feature selection approaches, namely PCA and ensemble feature selection (EFS), to select relevant features from a novel dataset called CICIDS 2018. Experimental results demonstrated that the integration of EFS with AdaBoost outperformed PCA with AdaBoost. The strategically positioned IDS within the network analyses passing traffic, continuously monitors traffic from network devices, and cross-references it with a comprehensive library of all known threats. Upon identifying an attack, it detects any unusual behavior and promptly alerts the administrator.

3. METHODS AND MATERIAL

ML or DL algorithms can undergo training through various procedures, including supervised and unsupervised learning. Supervised learning involves classification based on data instances marked in the training phase. Algorithms employed in supervised learning encompass ANN, DT (both types c4.5, ID3), KNN, NB, RF, SVM, and CNN. On the other hand, unsupervised learning detects unlabeled data instances, with clustering being the dominant learning method. Algorithms in unsupervised learning include k-means clustering, EM clustering, and SOM.

3.1. Datasets

The most challenging aspect of evaluating IDS is selecting an appropriate dataset. This section discusses commonly [37], [38]. However, these datasets are inherently complex, necessitating advanced solutions. The most popular datasets used in the literature are described as follows:

3.1.1. UNSW-NB15

The UNSW-NB15 dataset was generated using the IXIA perfect storm tool and a simulation program. In contrast to NSL-KDD, this dataset encompasses original instances of various intrusion detection cases that commonly occur in contemporary scenarios. The dataset comprises 175,341 normal classes, 82,332 anomaly classes, and 49 extracted features [52]. The attacks in the dataset include fuzzier, analysis, backdoor, DoS, exploit, generic, and reconnaissance, shellcode, and worm attacks. illustrates the utilization of popular datasets in the literature.

3.1.2. CICIDS2017

The CICIDS2017 dataset closely emulates real-world network data (PCAPs) and leverages CICFlowmeter-V3.0 to extract 78 features and 79 labels. The dataset encompasses the distinctive behavioral patterns of 25 users across various protocols, including HTTP, HTTPS, FTP, SSH, and email, as detailed. Data are captured over different periods. In alignment with the 2016 McAfee report, the attacks in the dataset are categorized into brute force FTP, brute force SSH, DoS, Heartbleed, infiltration, botnet, and DDoS attacks, distinguishing it from the previously mentioned datasets [53]. CICIDS2017 achieves a profiling of abstract characteristics of human interactions using the B-profile system and applies the alpha profile to simulate diverse multi-stage attack scenarios. The dataset's main features set it apart from others in terms of its realistic and reliable benchmarking. The benchmarking encompasses 11 criteria, including complete traffic and available protocols, ensuring the reliability of the evaluation [54].

3.1.3. NSL-KDD

The NSL-KDD dataset comprises the fundamental records of the KDD Cup 1999 dataset. This dataset categorizes its data features into several groups [10]. The NSL-KDD dataset reduces data size by removing duplicate records, thereby enhancing the performance of ML algorithms.

4. BENCHMARKING OF ML-AIDS IN CICIDS2017

AIDS has garnered substantial research interest since the beginning of the decade. Constructing an AIDS using AI or ML techniques to fortify networks against modern attacks remains a vital concern for researchers. Numerous studies aim to enhance AIDS and assess their results using various metrics. Given that most ML algorithms are parameterized, estimating their behavior from processed data is challenging. Moreover, random parameters can significantly impact the performance of AIDS models [55]. Consequently, parameter behavior must be tuned to achieve a satisfactory evaluation. Proposed benchmarking methodology and associated procedures for testing and evaluating ML-AIDS models.

5. PRE-PROCESSING

The study utilizes the Botnet.CSV data, a component of the CICIDS-2017 dataset. Botnet.CSV comprises eight traffic monitoring sessions, each represented as a comma-separated value (CSV) file. The file encompasses regular traffic labelled as "Botnet" and anomaly traffic categorized as "Attacks" traffic. The various types of attacks in the dataset are detailed in the second column. During the numericalization step, noise values like null or infinity symbols are replaced with zeros or mean values. Subsequently, normalization is applied, addressing the extensive values and non-distributed data of some CICIDS2017 attributes based on their histograms. This ensures that all attribute values are brought to the same scale, applying the normalization procedure interval.

6. TRAINING

Numerous strategies can be employed to generate hyper parameters, with trial and error being the most common [56]. Another strategy involves k-fold cross-validation, where the dataset is divided into training and testing parts. This approach provides a means for evaluating the performance of ML-AIDS models and determining optimal settings.

In the benchmarking methodology, k-fold cross-validation partitions are meticulously established with specific limits for training and testing percentages (e.g., 40%-60%, 50%-50%, or 60%-40%). These partitions are concealed during the training stage to assess the reliability, generalizability, and effectiveness of ML-AIDS models. The training process involves tuning the parameters of each algorithm, as elaborated in the following section, and evaluating the output. This exact process is iterated for the remaining parameters.

7. TESTING

During the testing phase, ten popular supervised and unsupervised ML algorithms are employed to identify effective and efficient ML-AIDS for networks and computers. The supervised algorithms encompass ANN, DT, KNN, NB, RF, SVM, and CNN. On the other hand, the unsupervised algorithms include EM, k-means, and SOM. Multiple models of these algorithms are introduced, and the tuning and training parameters of each algorithm are scrutinized to attain optimal classification results.

8. BENCHMARKING

The performance of ML algorithms cannot be visually interpreted, and quantitative metrics such as precision, recall, F1-score, and confusion matrix need to be employed for evaluation. The choice of the best evaluation metrics depends on the selected ML algorithms, processed data, and the application domain. The study evaluates AIDSs using the CICIDS2017 dataset, considering accuracy, precision, sensitivity (recall), F-score, training time, and prediction time as the evaluation metrics.

8.1. Accuracy

The first evaluation metric evaluated by the study is accuracy. Accuracy is defined as the ratio of correct predictions. Accordingly, this definition encompasses both true positives (TP) and true negatives (TN) of attacks, relative to the total number of tested cases.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

8.2. Precision

Another metric evaluated in the scope of this study is precision. Precision (TP rate) measures the proportion of positives that are correctly identified. The mathematical formula is shown in (2).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

8.3. Sensitivity

Sensitivity is yet another important metric evaluated in this study. Sensitivity (recall) measures the number of correct classifications penalized by the number of missed entries identified. Sensitivity can be calculated using the in (3).

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{3}$$

8.4. F1-score

An important aspect in evaluating the performance of ML algorithms is finding a balance between precision and recall. Hence, F1-score as an evaluation metric finds a balance between precision and recall. F1-score can be calculated using the as in (4).

$$\text{F1}_{\text{Score}} = 2 * \frac{Precision + Recall}{Precision * Recall} \tag{4}$$

9. RESULTS AND DISCUSSION

The section outlines the benchmarking of classification algorithms for a multi-class labeled AIDS CICIDS2017 dataset. These 10 ML algorithms are categorized into seven supervised (KNN, SVM, DT (both types c4.5, ID3), RF, ANN, NB, and CNN) and three unsupervised (K-means clustering, EM clustering, and

SOM) algorithms. Some of these algorithms consist of several models, and their settings involve parameter tuning to determine the best-fit parameters and optimal initial values for training and testing. The ML-AIDS algorithms are implemented using Python3 in Anaconda 3 on a computer with OPTIPLEX 3010 Dell, Intel Core i3, 3.60 GHz processor, 4 GB primary memory, and 2 GB GPU on Ubuntu 16.04. The selected ML algorithms undergo seven supervised and three unsupervised learning tests. The evaluation metrics include accuracy, precision, recall, F1-score, T1, and T2. The AIDS CICIDS2017 dataset used in these tests has four types of classes, and their names, labels, and supports are presented.

9.1. Results of supervised learning algorithms

9.1.1. ANN

The testing of the ANN classifier involves three models, each represented based on parameters related to the construction of the training models. Default values are assigned to these parameters (activation = 'ReLU,' alpha = 0.0001, batch size = 'auto,' number of hidden layers = '4,' optimizer ="). Table 1 provides the results for the three ANN models. Some variations are observed in the training parameters. Specifically, one ANN model successfully detects C1, C2, C3, and C4 attacks, while the other two struggle to detect C4 attacks. The most effective ANN model has the settings Solver = 'Adam,' loss = 'categorical_crossentropy,' and epoch = 15. This model achieves 86% accuracy, 88% precision, 94% recall, 91% F1-score, 53.78s training time, and 48.03s testing time. In addition, the loss is the value of the cost function for training data. Figure 2 below show a comparison between model loss and epoch.

Table 1. Classification report for ANN

	Precision	Recall	F1-score	Support
0	0.88	0.94	0.91	1,214
1	0.77	0.59	0.67	386
Accuracy			0.86	160
Macro avg	0.82	0.77	0.79	160
Weighted avg	0.85	0.86	0.85	1,600

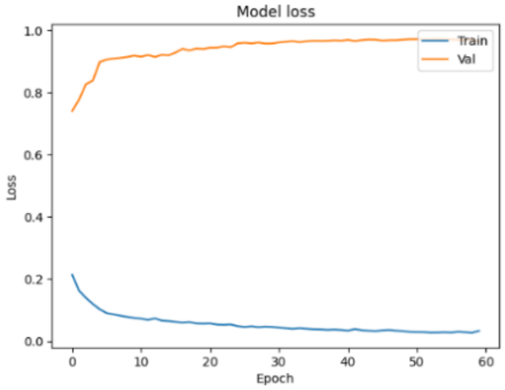

Figure 2. Model loss Vs. Epcoh

9.1.2. RF

The RF classifier employs numerous subtrees constructed to detect various types of attacks. The number of subtrees and the maximum tree level in RF impact the detection rate and time complexity. Consequently, the subtree is treated as a parameter in the evaluation. The testing of the RF classifier involves five models, each represented based on parameters related to the construction of the training models. The adjustable parameters include the maximum depth (max depth) and the number of estimators (n estimators). Table 2 presents the results of the five RF models. Variations are observed in the training parameters, and Figure 3 displays the heat map of the classification report. Specifically, two RF models can detect C1, C2, C3, and C4 attacks, two fail to detect C4 attacks, and one fails to detect C2, C3, and C4 attacks. In summary, the dataset is highly imbalanced, and the class might be too "balanced" for the training model to pay more attention to the skewed attacks than the other majority classes. The best RF model has the settings n estimators = 100, max depth = none, and class weight = balanced. This model achieves 97% accuracy, 96% precision, 1% recall, 98% F1-score, 9.38s training time, and 6.76s testing time. However, RF is more time-consuming compared to the other models.

Table 2. Classification report for RF, (by author)

	precision	recall	F1-score	support
0	0.96	1.00	0.98	122
1	1.00	0.88	0.93	37
accuracy			0.97	160
macro avg	0.98	0.94	0.96	160
weighted avg	0.97	0.97	0.97	160

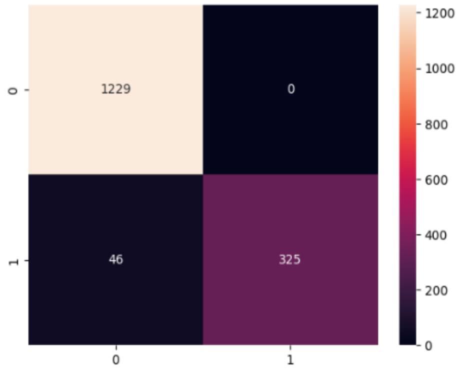

Figure 3. Classification report for RF

9.1.3. SVM

The SVM classifier testing involves four models represented based on the parameters related to the construction of the training models. The SVM model utilizes a kernel function in these datasets. Table 3 presents the results of four SVM models, and Figure 4 shows the heat map of the classification report. Variations are detected in the kernel function and training parameters. Specifically, one SVM model can detect C1, C2, C3, and C4 attacks, whereas the other three fail to detect the rare C4 attacks. In summary, the dataset could be more balanced. Accordingly, the class might be set to “balanced” (to pay more attention to the skewed attacks than the other majority classes), and the number of iterations is set to –1. The best SVM model has the settings kernel = RBF, max iter = –1, and class weight = balanced and achieves 77% accuracy, 77% precision, 100% recall, 87% F1-score, 343.56s training time, and 33.17s testing time. However, the model has a longer detection time than the other models.

Table 3. Classification report for SVM, (by author)

	precision	recall	F1-score	support
0	0.77	1.00	0.87	1,229
1	0	0	0	371
accuracy			0.77	1,600
macro avg	0.38	0.5	0.43	160
weighted avg	0.59	0.77	0.67	160

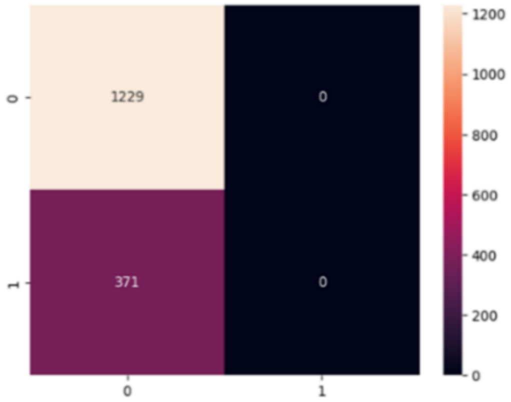

Figure 4. Heat map of classification for SVM

9.1.4. KNN

The KNN classifier testing involves five models denoted by 1, 2, 3, 4, and 5 k neighbors. Table 4 presents the results of these models, and Figure 5 shows the heat map of the classification report. Variations are observed in their k values. Specifically, 3-NN models can detect all BENIGN (C1), brute force (C2), XSS (C3), and SQL injection (C4) attacks, whereas 4-NN and 5-NN models fail to detect C4 attacks. The 1-NN model obtains the best results for all classes with 93% accuracy, 96% precision, 94% recall, 95% F1-score, 11.13s training time, and 7.92s testing time.

Table 4. Classification report for KNN

	precision	recall	F1-score	support
0	0.96	0.94	0.95	122
1	0.83	0.87	0.85	371
accuracy			0.93	1,600
macro avg	0.89	0.91	0.90	1,600
weighted avg	0.93	0.93	0.93	1,600

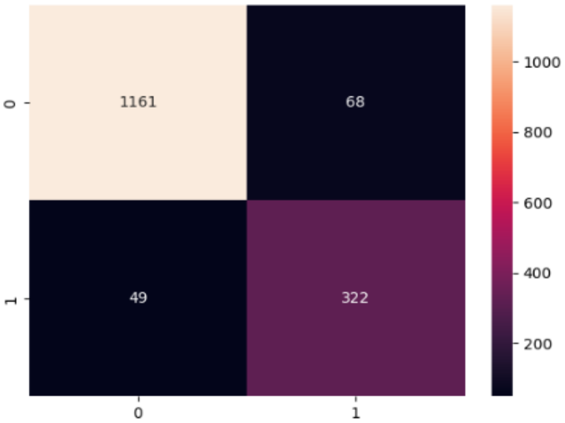

Figure 5. Heat map of classification report for KNN

10. CONCLUSION

The study evaluates the performance of related ML-based AIDS models in detecting attacks, specifically on a binary dataset. Limitations are identified in these models, particularly in detecting novel attack types with multi-classification challenges. Moreover, a majority of the reviewed studies rely on accuracy as the primary evaluation metric, limiting the ability to perform a fair and comprehensive comparison of various ML-AIDS approaches. The utilization of supervised learning methodologies in the proposed approach has proven to be a promising avenue for enhancing botnet detection accuracy. The integration of a diverse range of ML algorithms, particularly KNN-AIDS, DT-AIDS, and NB-AIDS, showcased commendable performance in effectively identifying various types of botnet attacks. This underscores the significance of leveraging labeled data for training models, enabling them to learn and generalize from known instances, thereby improving detection efficacy. The experimental results demonstrate that no single supervised learning algorithm is universally effective in detecting all forms of botnet attacks. Different algorithms exhibit distinct strengths and weaknesses, emphasizing the importance of a diverse and adaptive approach in botnet detection. The observed excellence of KNN-AIDS, DT-AIDS, and NB-AIDS models signifies their potential applicability in real-world scenarios.

However, the study also reveals limitations, notably the suboptimal performance of SOM-AIDS and EM-AIDS models, primarily due to high FP and FN alarms. This underscores the need for continuous refinement and optimization of supervised learning models, acknowledging the complexities and dynamics inherent in botnet behaviors. In conclusion, the application of supervised learning within the Hybrid SOM Approach offers a nuanced perspective on botnet detection, shedding light on the strengths and challenges of different algorithms. The outcomes provide a foundation for further research and optimization, encouraging ongoing efforts to enhance the accuracy and adaptability of supervised learning models in safeguarding IoT sensor networks against the evolving threat landscape of botnets.

REFERENCES

[1] L. Goasduff, “Gartner says 5.8 billion enterprise and automotive IoT endpoints will be in use in 2020,” *Gartner Report 2019*, 2020. https://www.gartner.com/en/site-index.

[2] Fortune Business Insights, “COVID-19 impact: high dependency on novel technology to bode well for market,” *fortunebusinessinsights*, 2020. https://www.fortunebusinessinsights.com/industry-reports/internetof-things-iot-market-100307.

[3] S. Zhao, S. Li, L. Qi, and L. Da Xu, “Computational intelligence enabled cybersecurity for the internet of things,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, pp. 666–674, 2020, doi: 10.1109/TETCI.2019.2941757.

[4] A. Greenberg, “Hackers remotely kill a jeep on the highway—with me in it | WIRED,” *Wired.com*, 2015, (accesed 21 Jul 2015) http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

[5] M. S. Gadelrab, M. ElSheikh, M. A. Ghoneim, and M. Rashwan, “BotCap: machine learning approach for botnet detection based on statistical features,” *International Journal of Communication Networks and Information Security*, vol. 10, no. 3, pp. 563–579, 2018, doi: 10.17762/ijcnis.v10i3.3624.

[6] S. G. Abbas, S. Zahid, F. Hussain, G. A. Shah, and M. Husnain, “A threat modelling approach to analyze and mitigate Botnet attacks in smart home use Case,” in *Proceedings - 2020 IEEE 14th International Conference on Big Data Science and Engineering, BigDataSE 2020*, Dec. 2020, pp. 122–129, doi: 10.1109/BigDataSE50710.2020.00024.

[7] M. Parreira do Amaral, “Imagining and transforming higher education. knowledge production in the new geopolitics of knowledge,” in *Educational Governance Research*, vol. 17, 2022, pp. 35–51.

[8] R. Williams, E. McMahon, S. Samtani, M. Patton, and H. Chen, “Identifying vulnerabilities of consumer internet of things (IoT) devices: A scalable approach,” *2017 IEEE International Conference on Intelligence and Security Informatics: Security and Big Data, ISI 2017*, pp. 179–181, 2017, doi: 10.1109/ISI.2017.8004904.

[9] A. Yulianto, P. Sukarno, and N. A. Suwastika, “Improving AdaBoost-based intrusion detection system (IDS) Performance on CIC IDS 2017 dataset,” *Journal of Physics: Conference Series*, vol. 1192, no. 1, p. 012018, Mar. 2019, doi: 10.1088/1742-6596/1192/1/012018.

[10] R. Vinayakumar, K. P. Soman, P. Poornachandran, M. Alazab, and A. Jolfaei, “DBD: deep learning DGA-based Botnet Detection,” in *Advanced Sciences and Technologies for Security Applications*, 2019, pp. 127–149.

[11] R. U. Khan, R. Kumar, M. Alazab, and X. Zhang, “A Hybrid Technique To Detect Botnets, Based on P2P Traffic Similarity,” in *2019 Cybersecurity and Cyberforensics Conference (CCC)*, May 2019, pp. 136–142, doi: 10.1109/CCC.2019.00008.

[12] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, “Performance evaluation of Botnet DDoS attack detection using machine learning,” *Evolutionary Intelligence*, vol. 13, no. 2, pp. 283–294, 2020, doi: 10.1007/s12065-019-00310-w.

[13] R. U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz, and M. Alazab, “An adaptive multi-layer botnet detection technique using machine learning classifiers,” *Applied Sciences (Switzerland)*, vol. 9, no. 11, p. 2375, Jun. 2019, doi: 10.3390/app9112375.

[14] C. Joshi, R. K. Ranjan, and V. Bharti, “A fuzzy logic based feature engineering approach for Botnet detection using ANN,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6872–6882, 2021, doi: 10.1016/j.jksuci.2021.06.018.

[15] R. Biswas and S. Roy, “Botnet traffic identification using neural networks,” *Multimedia Tools and Applications*, vol. 80, no. 16, pp. 24147–24171, 2021, doi: 10.1007/s11042-021-10765-8.

[16] S. Pokhrel, R. Abbas, and B. Aryal, “IoT Security: Botnet detection in IoT using Machine learning,” *arXiv preprint*, 2021, [Online]. Available: http://arxiv.org/abs/2104.02231.

[17] A. M. U. D. Khanday, S. T. Rabani, Q. R. Khan, N. Rouf, and M. Mohi Ud Din, “Machine learning based approaches for detecting COVID-19 using clinical text data,” *International Journal of Information Technology (Singapore)*, vol. 12, no. 3, pp. 731–739, 2020, doi: 10.1007/s41870-020-00495-9.

[18] Bashir U and C. M, “Snort intrusion detection and prevention system,” 2014, [Online]. Available: http://www.snort.org.

[19] V. Kotu and B. Deshpande, “Anomaly detection,” in *Data Science*, Elsevier, 2019, pp. 447–465.

[20] G. Kumar, K. Kumar, and M. Sachdeva, “The use of artificial intelligence based techniques for intrusion detection: a review,” *Artificial Intelligence Review*, vol. 34, no. 4, pp. 369–387, 2010, doi: 10.1007/s10462-010-9179-5.

[21] A. Pérez-Suárez, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa, “A review of conceptual clustering algorithms,” *Artificial Intelligence Review*, vol. 52, no. 2, pp. 1267–1296, 2019, doi: 10.1007/s10462-018-9627-1.

[22] “Self-organizing map.” https://commons.wikimedia.org/wiki/Category: Self-organizing-map (accessed Jun. 09, 2018).

[23] M. Liukkonen and Y. Hiltunen, “Recognition of systematic spatial patterns in silicon wafers based on SOM and K-means,” *IFAC-PapersOnLine*, vol. 51, no. 2, pp. 439–444, 2018, doi: 10.1016/j.ifacol.2018.03.075.

[24] X. Qu *et al.*, “A survey on the development of self-organizing maps for unsupervised intrusion detection,” *Mobile Networks and Applications*, vol. 26, no. 2, pp. 808–829, Apr. 2021, doi: 10.1007/s11036-019-01353-0.

[25] S. Li, L. Da Xu, and S. Zhao, “The internet of things: a survey,” *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, Apr. 2015, doi: 10.1007/s10796-014-9492-7.

[26] H. Al-Rushdan, M. Shurman, S. H. Alnabelsi, and Q. Althebyan, “Zero-day attack detection and prevention in software-defined networks,” in *Proceedings - 2019 International Arab Conference on Information Technology, ACIT 2019*, 2019, pp. 278–282, doi: 10.1109/ACIT47987.2019.8991124.

[27] A. Rouhi and H. Nezamabadi-Pour, “Feature selection in high-dimensional data,” in *Optimization, Learning, and Control for Interdependent Complex Networks*, M. H. Amini, Ed. Cham: Springer International Publishing, 2020, pp. 85–128.

[28] F. Almudaires and M. Almaiah, “Data an overview of cybersecurity threats on credit card companies and credit card risk mitigation,” in *2021 International Conference on Information Technology (ICIT)*, Jul. 2021, pp. 732–738, doi: 10.1109/ICIT52682.2021.9491114.

[29] J. P. Yaacoub, H. Noura, O. Salman, and A. Chehab, “Security analysis of drones systems: attacks, limitations, and recommendations,” *Internet of Things (Netherlands)*, vol. 11, 2020, doi: 10.1016/j.iot.2020.100218.

[30] K. Alieyan, A. Almomani, R. Abdullah, B. Almutairi, and M. Alauthman, “Botnet and internet of things (IoTs),” in *Research Anthology on Combating Denial-of-Service Attacks*, 2020, pp. 138–150, doi: 10.4018/978-1-7998-5348-0.ch007.

[31] J. M. Blythe, N. Sombatruang, and S. D. Johnson, “What security features and crime prevention advice is communicated in consumer IoT device manuals and support pages?,” *Journal of Cybersecurity*, vol. 5, no. 1, Jan. 2019, doi: 10.1093/cybsec/tyz005.

[32] P. Malhotra, Y. Singh, P. Anand, D. K. Bangotra, P. K. Singh, and W. C. Hong, “Internet of things: evolution, concerns and security challenges,” *Sensors*, vol. 21, no. 5, pp. 1–35, 2021, doi: 10.3390/s21051809.

[33] R. M. Gomathi, G. H. S. Krishna, E. Brumancia, and Y. M. Dhas, “A survey on iot technologies, evolution and architecture,” 2018, doi: 10.1109/ICCCSP.2018.8452820.

[34] L. Santos, C. Rabadão, and R. Gonçalves, “Flow monitoring system for IoT networks,” *Advances in Intelligent Systems and Computing*, vol. 931, pp. 420–430, 2019, doi: 10.1007/978-3-030-16184-2_40.

[35] E. Gyamfi and A. Jurcut, “Intrusion detection in internet of things systems: a review on design approaches leveraging multi-access edge computing, machine learning, and datasets,” *Sensors*, vol. 22, no. 10, 2022, doi: 10.3390/s22103744.

[36] Y. Xing, H. Shu, H. Zhao, D. Li, and L. Guo, “Survey on Botnet detection techniques: classification, methods, and evaluation,” *Mathematical Problems in Engineering*, vol. 2021, 2021, doi: 10.1155/2021/6640499.

[37] W. G. Negera, F. Schwenker, T. G. Debelee, H. M. Melaku, and Y. M. Ayano, “Review of Botnet attack detection in SDN-enabled IoT using machine learning,” *Sensors*, vol. 22, no. 24, 2022, doi: 10.3390/s22249837.

[38] I. H. Sarker, “Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective,” *SN Computer Science*, vol. 2, no. 3, p. 154, May 2021, doi: 10.1007/s42979-021-00535-6.

[39] S. Ryu and B. Yang, “A comparative study of machine learning algorithms and their ensembles for botnet detection,” *Journal of Computer and Communications*, vol. 06, no. 05, pp. 119–129, 2018, doi: 10.4236/jcc.2018.65010.

[40] J. Arshad, M. A. Azad, M. M. Abdeltaif, and K. Salah, “An intrusion detection framework for energy constrained IoT devices,” *Mechanical Systems and Signal Processing*, vol. 136, 2020, doi: 10.1016/j.ymssp.2019.106436.

[41] M. Gopinath and S. C. Sethuraman, “A comprehensive survey on deep learning based malware detection techniques,” *Computer Science Review*, vol. 47, 2023, doi: 10.1016/j.cosrev.2022.100529.

[42] N. Ahmed *et al.*, “Network threat detection using machine/deep learning in SDN-based platforms: a comprehensive analysis of state-of-the-art solutions, discussion, challenges, and future research direction,” *Sensors (Basel, Switzerland)*, vol. 22, no. 20, 2022, doi: 10.3390/s22207896.

[43] P. L. S. Jayalaxmi, R. Saha, G. Kumar, M. Conti, and T. H. Kim, “Machine and deep learning solutions for intrusion detection and prevention in IoTs: a survey,” *IEEE Access*, vol. 10, pp. 121173–121192, 2022, doi: 10.1109/ACCESS.2022.3220622.

[44] Y. Yue, S. Li, P. Legg, and F. Li, “Deep Learning-Based Security Behaviour Analysis in IoT Environments: A Survey,” *Security and Communication Networks*, vol. 2021, 2021, doi: 10.1155/2021/8873195.

[45] M. Kocakulak and I. Butun, “An overview of Wireless Sensor Networks towards internet of things,” *2017 IEEE 7th Annual Computing and Communication Workshop and Conference, CCWC 2017*, pp. 1–6, 2017, doi: 10.1109/CCWC.2017.7868374.

[46] M. M. Salim, S. Rathore, and J. H. Park, “Distributed denial of service attacks and its defenses in IoT: a survey,” *Journal of Supercomputing*, vol. 76, no. 7, pp. 5320–5363, Jul. 2020, doi: 10.1007/s11227-019-02945-z.

[47] M. Wazzan, D. Algazzawi, O. Bamasaq, A. Albeshri, and L. Cheng, “Internet of things botnet detection approaches: Analysis and recommendations for future research,” *Applied Sciences (Switzerland)*, vol. 11, no. 12, p. 5713, Jun. 2021, doi: 10.3390/app11125713.

[48] M. H. Bhuyan, V. C. Bhavsar, D. K. Bhattacharyya, J. K. Kalita, and M. Conti, “IoT Botnets: recent advances, key challenges, and countermeasures,” *IEEE Internet of Things Journal*, 2020.

[49] S. Kim, A. K. Dey, and H. Kim, “A survey of IoT security solutions for DDoS attacks and Botnet defenses,” *IEEE Access*, 2020.

[50] A. Conti, M. D. Poletti, and M. Secchi, “Detecting IoT Botnets: a multistage hybrid approach,” *IEEE Transactions on Industrial Informatics*, 2018.

[51] J. Zheng and A. Jamalipour, “Wireless sensor networks,” in *Wireless Sensor Networks: A Networking Perspective*, J. Zheng and A. Jamalipour, Eds. Wiley, 2009, pp. 1–489.

[52] G. Dizon and J. M. Gayed, “Examining the impact of grammarly on the quality of mobile L2 writing,” *JALT CALL Journal*, vol. 17, no. 2, pp. 74–92, 2021, doi: 10.29140/JALTCALL.V17N2.336.

[53] M. Wazzan, D. Algazzawi, O. Bamasaq, A. Albeshri, and L. Cheng, “Internet of things botnet detection approaches: analysis and recommendations for future research,” *Applied Sciences (Switzerland)*, vol. 11, no. 12, 2021, doi: 10.3390/app11125713.

[54] Y. Meidan *et al.*, “N-BaIoT-network-based detection of IoT botnet attacks using deep autoencoders,” *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018, doi: 10.1109/MPRV.2018.03367731.

[55] S. Dange, “IoT Botnet : the largest threat to the IoT network IoT Botnet : the largest threat to the IoT network,” *Springer: Singapore*, no. September, pp. 137–157, 2019.

[56] G. Gales, “Network intrusion detection system and method,” *Google Patents Application US12/411,916*, 2003. https://patents.google.com/patent/US20030084323A1/en (accessed Oct. 07, 2022).

[57] J. Brownlee, “Supervised and unsupervised machine learning algorithms,” *Understand Machine Learning Algorithms*, 2016. https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/%0Ahttp://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ (accessed Aug. 26, 2022).

[58] G. Bonaccorso, “Machine learning algorithms,” in *Packt Publishing Ltd.: Birmingham, UK*, IGI Global, 2017.

[59] M. Zamani and M. Movahedi, “Machine learning techniques for intrusion detection,” *arXiv*, 2013.

[60] S. V. N. Vishwanathan and M. N. Murty, “SSVM: a simple SVM algorithm,” *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 2393–2398, 2002, doi: 10.1109/ijcnn.2002.1007516.

[61] I. El Naqa and M. J. Murphy, “What is machine learning? Springer: Berlin/Heidelberg,” *Springer*, pp. 3–11, 2015.

[62] S. Zwane, P. Tarwireyi, and M. Adigun, “A flow-based IDS for SDN-enabled tactical networks,” 2019, doi: 10.1109/IMITEC45504.2019.9015900.

[63] T. Kohonen, “Self-organized formation of topologically correct feature maps,” *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982, doi: 10.1007/BF00337288.

[64] A. Ultsch, “Self-organizing neural networks for visualization and classification,” in *Proceedings of the International Neural Network Conference (INNC)*, 1993, pp. 307–313.

[65] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM Computing Surveys*, vol. 46, no. 4, 2014, doi: 10.1145/2523813.

[66] I. Žliobaitė, A. Bifet, J. Read, B. Pfahringer, and G. Holmes, “Evaluation method and decision theory for classification streaming data with temporal dependence,” *Machine Learning*, vol. 98, no. 3, pp. 455–482, 2015, doi: 10.1007/s10994-014-5441-4.

[67] F. Hu, Q. Hao, and K. Bao, “A survey on software-defined network and OpenFlow: from concept to implementation,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014, doi: 10.1109/COMST.2014.2326417.

[68] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, “Survey on SDN based network intrusion detection system using machine learning approaches,” *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, 2019, doi: 10.1007/s12083-017-0630-0.

[69] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, “Shallow and deep networks intrusion detection system: a taxonomy and survey,” *arXiv*, Jan. 2017, [Online]. Available: http://arxiv.org/abs/1701.02145.

[70] J. Tripathi, S. Tiwari, A. Saini, and S. Kumari, “Prediction of movie success based on machine learning and twitter sentiment analysis using internet movie database data,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 29, no. 3, pp. 1750–1757, Mar. 2023, doi: 10.11591/ijeecs.v29.i3.pp1750-1757.

BIOGRAPHIES OF AUTHORS

Mwaffaq Abu AlHija Mwaffaq Abu AL-Haija is an Associate Professor at the department of Computer Science at Al-Ahliyya Amman University, Jordan. His main research interests lie in the areas of operating system design, distributed computing systems, multimedia communication and networking, mobile and wireless networks, data and network security, wireless sensor networks, sorting and searching algorithms, and parallel computing. He can be contacted at email: m.abualhija@ammanu.edu.jo.

Hamza Jehad Alqudah Cyber Security Researcher at the Networks and Cyber Security Department, Al-Ahliyya Amman University, Jordan. His research interests include digital forensics, cyberbullying, IoT, network security, and malware detection. He can be contacted at email: hamzaalqudah777@gmail.com.

Hiba Dar-Othman Cyber Security Researcher at the Networks and Cyber Security Department, Al-Ahliyya Amman University, Jordan. His research interests include malware detection, IoT, database management system, and network security. She can be contacted at email: hibaothman85@gmail.com.