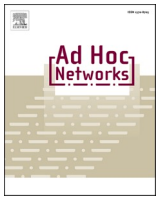
Ad Hoc Networks 152 (2024) 103322

Contents lists available at ScienceDirect

Ad Hoc Networks

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



Survey paper

Denial of service attacks in edge computing layers: Taxonomy, vulnerabilities, threats and solutions

Ryhan Uddin [a], Sathish A.P. Kumar [a,*], Vinay Chamola [b]

[a] Department of Electrical Engineering and Computer Science, Cleveland State University, 2121 Euclid Ave, Cleveland, OH, 44115, USA
[b] Department of Electrical and Electronics Eng., BITS-Pilani, Pilani, Rajasthan 333031, India

ARTICLE INFO

Keywords:
Edge computing
DDoS
Security
Vulnerabilities
Attacks

ABSTRACT

Edge computing has emerged as the dominant communication technology connecting IoT and cloud, offering reduced latency and harnessing the potential of edge devices. However, its widespread adoption has also introduced various security vulnerabilities, similar to any nascent technology. One notable threat is the denial of service (DoS) attack, including its distributed form, the distributed denial of service (DDoS) attack, which is the primary focus of this research. This paper aims to explore the impact of different types of DoS and DDoS attacks on edge computing layers by examining the vulnerabilities associated with various edge peripherals. Additionally, existing detection and prevention mechanisms are investigated to address these weaknesses. Furthermore, a theoretical architecture is proposed to mitigate distributed denial of service attacks targeting edge systems. By comprehensively analyzing and addressing the security concerns related to DoS and DDoS attacks in edge computing, this research aims to contribute to the development of robust and secure edge computing systems.

Abbreviation

ACK	Acknowledgment
AAI	Active ARP Inspection
ARP	Address Resolution Protocol
ASLR	Address Space Layout Randomization
AES	Advanced Encryption Standard
API	Application Programming Interface
AUC	Area under the Curve
AI	Artificial Intelligence
AAA	Authentication Authorization and Accounting
CPU	Central Processing Unit
CDL	Centralized Deep Learning
CLI	Command Line Interface
CNN	Convolutional Neural Network
CODE	COoperative Defense
CODE4MEC	COoperative DEfense for MEC
DTLS	Datagram Transport Layer Security
DNN	Deep Neural Network
DMZ	Demilitarized Zone
DoS	Denial of Service
DDL	Distributed Deep Learning
DDoS	Distributed Denial of Service
DPPC	Distributed Parallel Packet Classification
DNS	Domain Name System
EDIMA	Early Detection of IoT Malware Network Activity
FedAvg	Federated Averaging
FL	Federated Learning
5G	Fifth generation of broadband cellular network technology
FTP	File Transfer Protocol
4G	Fourth generation of broadband cellular network technology
FR-RED	Fractal Residual-Based Real-Time Detection
GHz	Giga Hertz
HTTP	Hypertext Transfer Protocol
IIoT	Industrial Internet of Things
GR-AD-KNN	Information Gain Ratio Average Distance KNN
IEEE	Institute of Electrical and Electronics Engineers
ICMP	Internet Control Message Protocol
IoT	Internet of Things
IP	Internet Protocol
IPSec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IRC	Internet Relay Chat

* Corresponding author.
E-mail address: s.kumar13@csuohio.edu (S.A.P. Kumar).

https://doi.org/10.1016/j.adhoc.2023.103322
Received 10 July 2023; Received in revised form 22 September 2023; Accepted 4 October 2023
Available online 7 October 2023

ISP	Internet Service Provider
IDPS	Intrusion Detection / Prevention System
IDS	Intrusion Detection System
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
KDTree	K-Dimensional Tree
k-NN	K-Nearest-Neighbor
LSVM	Lagrangian Support Vector Machine
LDL	Localized Deep Learning
LSTM	Long Short-Term Memory
LTE	Long-term Evolution
LDoS	Low-rate DoS
ML	Machine Learning
MITM	Man in the Middle
MAC	Media Access Control
MEC	Mobile Edge Computing
MEC	Multi-Access Edge Computing
MLP	Multilayer Perceptron
MTD	Moving Target Defense
MVF	Multi-View Visibility Framework
NFV	Network Function Virtualization
NN	Neural Network
NGFW	Next-Gen Firewalls
NCAS	Normalized Cumulative Amplitude Spectrum
NFGA	Network Flow Guard for ARP
OAI	Open Air Interface
OS	Operating System
PPS	Packet Per Seconds
P2P	Peer to Peer
POD	Ping of Death
PSD	Power Spectral Density
QuinDC	Quintile Deviation Checking
RF	Radio Frequency
RTVD	Real-time Volumetric Detection
ROC	Receiver Operating Characteristics
RNN	Recurrent Neural Network
RRL	Response Rate Limiter
RTO	Retransmission Timeout
SDNWISE	SDN wrapper over WSN stack
SSH	Secure Shell Protocol
SAVE-S	Security-Aware edge serVer sElection under stochastic
SOM	Self-Organizing Map
SDN	Software-Defined Network
SDP	Software-Defined Perimeter
SD-IoT	Software-Defined Internet of Things
SHE	Structured Exception Handler
SEHOP	Structured Exception Handler Overwrite Protection
SVM	Support Vector Machine
SYN	Synchronize
Tbps	Terabits per second
3G	Third generation of broadband cellular network technology
TAD	Topological Anomaly Detection
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VNF	Virtual Network Functions
VSF	Virtual Security Functions
WSN	Wireless Sensor Network

1. Introduction

Edge computing, a distributed computing model situated at the periphery (edge) of the network, has gained prominence in recent years [1]. Originally introduced to address the need for cost-effective bandwidth utilization for long-distance IoT devices, edge computing has evolved to provide numerous benefits in the era of the Internet of Things (IoT). It enables enhanced processing capabilities, faster outputs, and efficient data usage [1]. The core principle of edge computing revolves around performing computations on systems that are in close proximity to end-users, thereby reducing latency and improving overall performance [1]. With the increasing demand for real-time data and instant connectivity, edge computing has become crucial in ensuring efficient bandwidth utilization, storage efficiency, and low latency [1]. These characteristics have contributed to the success of edge computing in various domains. Industrial automation is one field that has greatly benefited from edge computing [2]. By bringing computation closer to industrial devices and machinery, edge computing enables real-time data processing, faster decision-making, and improved operational efficiency [2]. Additionally, edge computing has found applications in augmented and virtual reality (AR/VR) [3], where the processing of resource-intensive AR/VR applications is shifted to edge devices, resulting in lower latency and enhanced user experiences [3]. Furthermore, edge computing has played a significant role in the advancement of artificial intelligence (AI) systems. By enabling distributed AI processing at the edge, edge computing reduces the reliance on centralized cloud infrastructure and facilitates faster inference and decision-making [4]. It has also been instrumental in applications such as security and surveillance [4], facial recognition [5], and virtual assistants [6], where low latency and real-time processing are essential for efficient operation.

Over time, edge computing has expanded its capabilities, enabling smarter devices with increased processing power. Fig. 1 illustrates the architecture of edge computing, demonstrating how it empowers cloud computing by distributing computation, offloading tasks, and processing capabilities between the cloud layer and IoT devices. However, along with these advancements, there has been a simultaneous rise in threats and vulnerabilities across various computing environments, including edge computing [7–24]. One prominent threat that affects both traditional and edge computing environments is the denial of service (DoS) attack. Within the realm of edge computing, distributed denial of service (DDoS) attacks pose an even more severe and pervasive threat. These attacks can have significant financial repercussions for major enterprises. The primary objective of both DoS and DDoS attacks is to disrupt services by engaging in malicious activities such as generating excessive and irrelevant traffic, overwhelming networks with excessive requests, and exhausting vital resources like bandwidth and hardware. As a result, these attacks often lead to additional consequences, such as overheating and property damage. Given the increasing prevalence of these attacks, we have conducted a comprehensive exploration of both DoS and DDoS attack types within the context of edge computing. We have categorized and analyzed the findings for each attack type, outlining their taxonomy and implications. However, due to their distributed nature, DDoS attacks represent a more sophisticated and widespread form of threat to edge systems globally. According to Radware's 2021–2022 global threat analysis report, malicious DDoS activities increased by 37 % in 2021 compared to the previous year [25]. In the fourth quarter of 2021, Microsoft Azure experienced the largest DDoS attack ever recorded, with an attack volume of 3.47 terabits per second (Tbps). Subsequently, in December, two more attacks were observed, with volumes of 3.25 Tbps and 2.55 Tbps, respectively [26]. These incidents highlight the urgent need for further research and preparation to defend against such destructive cyberattacks. Following are our major contributions:

- A layered approach has been adopted in this study to classify and analyze the different categories of denial of service (DoS) and distributed denial of service (DDoS) attacks specifically targeting edge systems. This approach highlights the specific layers within edge computing that are susceptible to these attacks. Additionally, both layer-based and holistic strategies have been examined as potential approaches to effectively counter or mitigate such denial-of-service attacks in edge systems.
- In order to comprehensively address the threats posed by DoS attacks in edge computing, the inherent vulnerabilities and weaknesses exploited by each type of DoS attack have been identified. This analysis sheds light on the specific attack vectors that threat actors

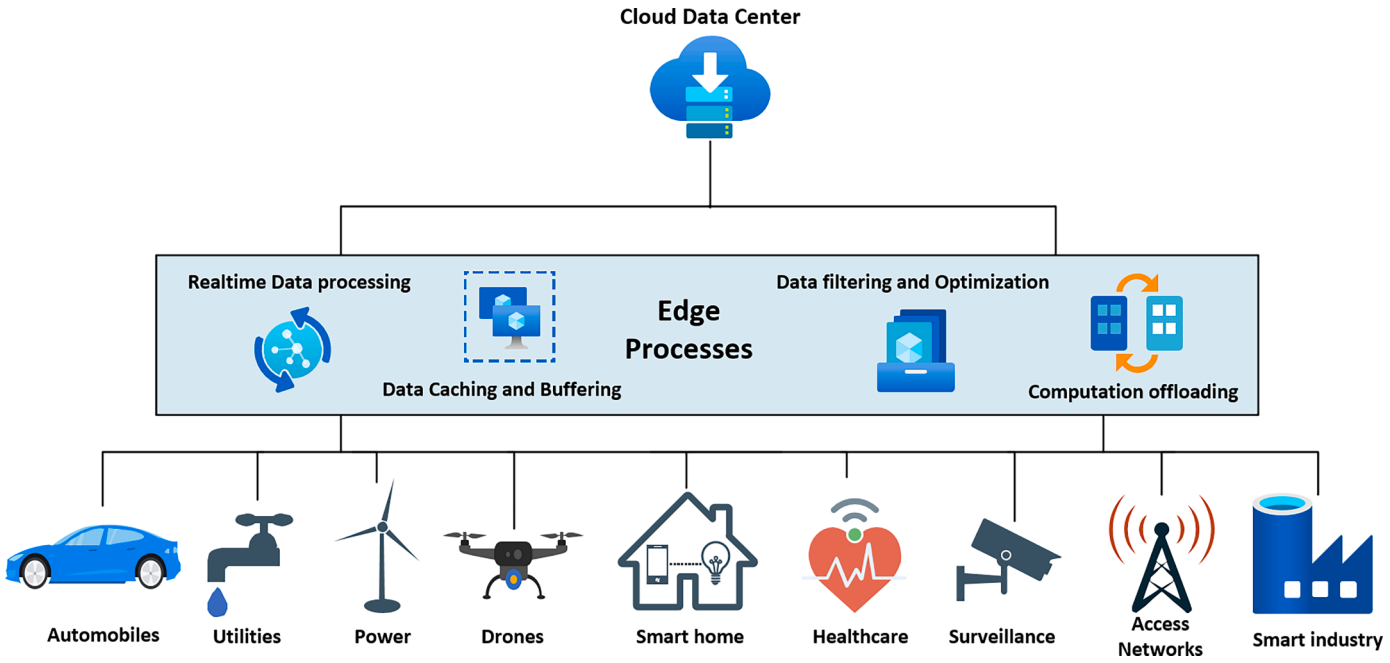


Fig. 1. Edge computing architecture.

may leverage to disrupt or compromise the availability and performance of edge computing systems.

- To understand the existing landscape of prevention and detection systems for DoS and DDoS attacks in edge computing, an investigation has been conducted on thirty state-of-the-art research-based systems. These systems have been analyzed to evaluate their efficacy in countering different types of DoS and DDoS attacks. The approaches employed by these systems have been carefully examined to identify their strengths, limitations, and areas of applicability in the context of edge systems.
- In response to the growing threat of distributed denial of service (DDoS) attacks in edge systems, a novel architecture based on federated learning and software-defined networking (SDN) has been proposed. This architecture serves as a robust detection and prevention mechanism specifically designed to mitigate the impact of DDoS attacks on edge systems. By leveraging federated learning techniques and the flexibility of SDN, this architecture aims to enhance the resilience of edge systems and ensure the continued availability and performance of critical edge computing resources.

The rest of the manuscript is organized as follows: In Section 2, an in-depth comparison of related literary works has been conducted, evaluating their research approaches, research focuses, and limitations concerning denial of service (DoS) attacks on edge systems. Section 3 outlines the approach and methodology employed in this survey. Section 4 provides an overview of various edge layers, which are then associated with various types of attacks affecting those edge layers in Section 6. Section 5 describes the general classifications of DoS and DDoS attack types in edge computing systems, associated threats and vulnerabilities, generic prevention / mitigation methods for each of those attack types, and also a brief exploration of the state-of-the-art research works to counter those attacks. Section 6 focuses on mapping the different attack types to individual edge layers. Tables are provided, compiling the threats and vulnerabilities associated with each attack type. Furthermore, an overview of 30 research-based solutions is presented, highlighting the platform types, prevention/mitigation techniques, testing metrics, and limitations of each solution. In Section 7, a novel security architecture for edge systems is proposed, primarily based on software-defined networking (SDN) and incorporating federated machine learning (ML) techniques. Section 8 offers recommendations for securing edge system-based networks, presenting both an edge layer-based approach and a holistic approach. Additionally, suggestions are provided for the future expansion of the security module to further enhance the protection of edge systems. Finally, in Section 9, the paper concludes with closing remarks, summarizing the key findings and contributions of the research. The following Fig. 2 offers an overview of the manuscript, segmented into nine primary sections, and highlights some major sub-sections.

2. Related works

Research studies and literature on edge computing have gained significant attention. However, it is noteworthy that only a small portion, approximately 7 %, of these studies are specifically focused on edge security [27]. This observation was made by the authors of an edge security survey, highlighting the limited emphasis on security in the context of edge systems. The survey was done by reviewing a large number of published papers on edge computing security from 2016 to early 2020. In their paper, the authors conducted comprehensive research, identifying and categorizing key security concerns in edge systems into five areas: access control, key management, attack mitigation, anomaly detection, and privacy protection. Each area presents its own set of challenges that need to be addressed to ensure the security of edge systems. When discussing attack mitigation challenges in edge computing, the authors specifically emphasized the growing concern of distributed denial of service (DDoS) attacks. These attacks are recognized as a significant bottleneck and hindrance to the potential of edge systems. Furthermore, the authors noted the scarcity of research works focused on DoS attacks in edge computing systems, which exacerbates the limitations in effectively countering these threats. While some published works exist on protecting edge systems from DoS attacks, the primary focus of these works has not been on DDoS attacks. Specifically, there is a lack of research that categorizes DDoS attacks based on individual edge layers, thoroughly examines the associated threats and vulnerabilities, explores standard countermeasures, and provides the latest solutions specifically targeting the categorized attack types in edge computing systems.

Several research studies have explored various security issues in edge

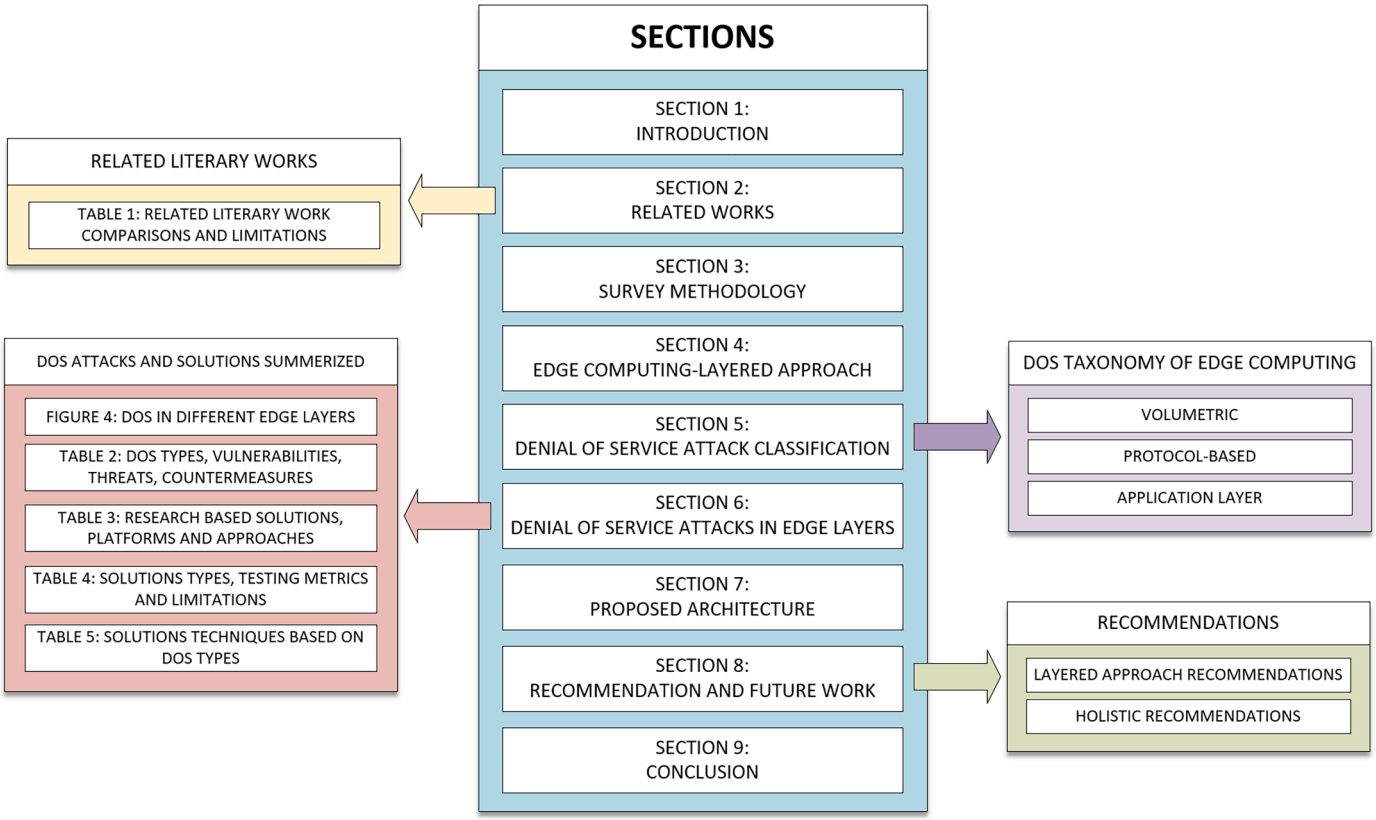


Fig. 2. Manuscript overview.

systems, providing general countermeasures to address these concerns. For instance, in a study conducted by the authors in [28], a review of the literature was conducted to survey DDoS attacks and task offloading in the domain of edge computing. The focus of this work was primarily on guiding future researchers in implementing real-life defense mechanisms against DDoS attacks on edge servers. While the study briefly demonstrated how to launch a DDoS attack on edge servers, it did not delve deeply into existing research on DDoS countermeasures. Another comprehensive work, described in [29], provides an extensive analysis of three paradigms - cloud, fog, and edge systems - and explores the fundamental components that govern these platforms. The study also highlights privacy and security concerns associated with these paradigms, taking into account metrics such as data privacy, data integrity, and data leakage. It identifies the challenges involved in achieving optimal platform security and data privacy in each paradigm. The authors further outline a layer-based approach, dividing the security

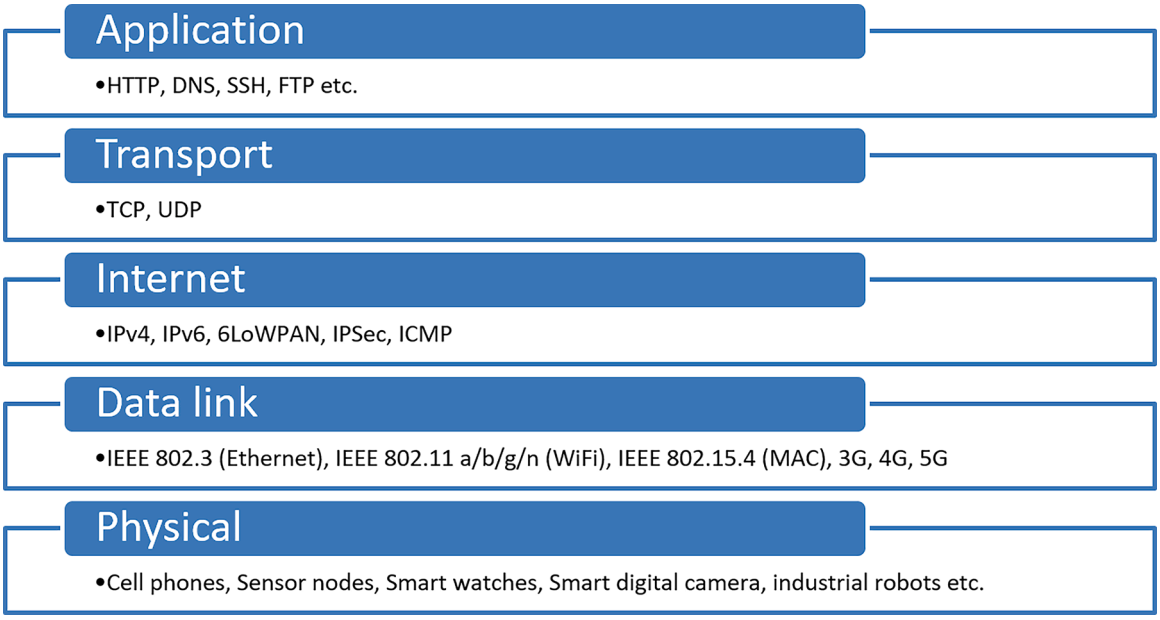


Fig. 3. Edge computing layers [29].

concerns into five common layers: application layer, session/presentation layer, transport layer, network layer, and PHY/MAC layer. This layer-based approach closely resembles our own approach presented in Fig. 3. The authors discuss security concerns affecting each layer and explore countermeasures for addressing these security issues. However, their survey primarily focuses on identifying commonalities among paradigms and examining generalized privacy and security issues. It does not delve deeply into specific DoS attack types for different edge layers, which was the primary objective of our survey.

In [30], the authors conducted an exploration of distributed denial of service (DDoS) flooding attacks, which they identified as a significant concern for network security. Their work delved into the scope and classifications of DDoS attacks, along with their associated countermeasures. The authors also discussed the motivations and incentives driving attackers to carry out these attacks. A specific focus was given to botnet-based DDoS attacks, which were further sub-classified into categories such as internet relay chat (IRC)-based, web-based, and peer-to-peer (P2P)-based attacks. The authors provided insights into both centralized and distributed countermeasures for mitigating DDoS attacks, offering a quantitative comparison of their respective advantages and disadvantages. Notably, the comparisons primarily centered around network and transport-level DDoS attacks, with consideration given to deployment locations. However, it is important to note that there are key distinctions between their work and our research. While their focus predominantly lies within traditional network architecture, our study centers specifically on edge computing systems. Moreover, their heavy emphasis on flooding-based attacks limits the inclusion of other critical attack types, such as low-rate denial of service (LDoS), ping of death (POD), zero-day DDoS attacks, and others.

In [31], the authors extensively discussed various security issues, including denial of service, side channels, malware injection, and authentication/authorization attacks. They provided detailed explanations of each issue and explored some of the available solutions. However, the key distinction between their work and ours lies in the specific focus and depth of exploration. Our research primarily centers on delving deeply into DoS and DDoS scenarios within the context of edge computing layers. We analyze each DoS attack category, mapping them to the corresponding edge layers, and provide a comprehensive list of the most recent solutions tailored to address these specific attack types. Additionally, we outline the security threats, vulnerabilities, and standard countermeasures associated with each category of DoS and DDoS attacks on edge systems. Table 1 in our work summarizes recent literature on DoS attacks in edge computing. It highlights the approach taken in each study, their primary focus, and the limitations of those approaches in comparison to our exploration of denial of service attacks in general. This provides a concise overview of existing research efforts, allowing for a comparative analysis and highlighting the unique contributions of our work.

3. Survey methodology

The survey conducted in this research followed a rigorous review process, involving extensive data analysis of papers extracted from prominent electronic databases, including IEEE-Explore, Scopus, Springer, Science Direct, and ResearchGate, among others. These databases were specifically selected for their comprehensive collection of publications and research works from reputable journals and conference proceedings, ensuring the inclusion of high-quality scholarly literature. To ensure the relevance and currency of the research, the focus was primarily placed on papers published from 2017 onward. This approach aimed to provide an up-to-date understanding of the most recent advancements in edge systems. Additionally, the inclusion criteria mandated that the literature be written in English, the most widely accessible language in the academic community. Over 30 independent research works were carefully incorporated into the survey, offering solutions to specific security vulnerabilities. For each problem, similar papers addressing the same issue were identified and reviewed, facilitating a comprehensive analysis of each attack type for edge systems. The classification of attack types into distinct branches, such as distributed and non-distributed, was conducted to form the DoS taxonomy. While edge computing layers are interconnected with fog and cloud systems, the scope of this survey was confined to edge systems alone. The research placed a stronger emphasis on edge-specific works, considering other platforms, such as hybrid or individual systems, to be beyond the scope of this study.

4. Edge computing – a layered approach

In edge computing, the architecture can be classified into multiple layers based on the associated protocols. Understanding these layers is crucial for comprehending the different types of attacks that can target each layer. The first layer in the edge computing architecture comprises a wide range of edge devices, including cell phones, sensors, smart watches, laptops, edge routers, industrial robots, surveillance systems, and more. These devices form the foundation of the edge computing ecosystem. The next layer, known as the data link layer, facilitates the transfer of data frames between devices using various mediums. This layer utilizes Mac addressing and acts as an intermediary stage for data transmission. The internet layer, which follows the data link layer, enables connectivity between edge devices, legacy systems, intermediate fog layers, and cloud infrastructures. This layer utilizes protocols such as IPv4, IPv6, 6LoWPAN, and others, along with Ethernet, Wi-Fi, cellular,

Table 1
Related literary work comparisons and limitations on DoS attacks in edge computing.

Related Work	Platform	Approach	Focus	Limitation
Zeyu et al. [27]	Edge computing	Analysis of security challenges	Defined 5 fields for the sources of security risks in edge computing	Did not explore in-depth DoS or DDoS classifications, hence no focused solutions for each category
Ahmad et al. [28]	Edge computing	DDoS attack initiation and guidance for future researchers	DDoS classifications, botnet-based DDoS attacks. Task offloading in edge systems.	Did not discuss research-based solutions for each DDoS category. Heavy focus on task offloading solutions.
Ometov et al. [29]	Cloud, fog, edge	Layered approach	Commonalities and heterogeneity of the security concerns on cloud, fog, and edge systems.	No elaboration on DDoS-specific attack categories. Generalized privacy /security concerns, very limited solutions.
Zargar et al. [30].	Legacy Network	Flooding-based DDoS attacks and countermeasures	DDoS and sub-categories of DDoS flooding attacks and incentive behind the attacks. Centralized and distributed countermeasures of DDoS attacks, advantages and disadvantages.	Work is solely based on traditional architecture, not edge systems. Heavy emphasis on flooding-based attacks, and no discussion on prominent attacks such as LDoS, POD, zero-day DDoS, etc.
Xiao et al. [31]	Edge computing	Generalized edge security	DDoS, Side channel attacks, Malware injection, and Authentication/authorization attacks.Root causes of attacks.	Missing DoS and DDoS categories such as Smurf, DNS reflection, Low-rate DoS (LDoS), etc., lack of in-depth analysis.

or other communication mediums. The transport layer provides the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) for efficient data transfer between hosts. It employs techniques like buffering and windowing (segmenting) to ensure smooth and reliable data transmission. The final layer is the edge applications layer, which encompasses various edge functionalities that leverage all the aforementioned layers. This layer performs processing tasks specific to the edge devices and applications. Fig. 3 illustrates the different layers involved in edge computing. It is important to note that all these layers are vulnerable to various types of attacks. In Section 6 of our research, we demonstrate the impact of different types of DoS/DDoS attacks on these individual layers. Prior to that, in Section 5, we provide a detailed explanation of the various attack types, ensuring a comprehensive understanding of their characteristics and implications.

5. Denial of service attack classifications

Denial of service (DoS) attacks have been prevalent in the digital landscape for quite some time, and they also pose a significant threat to the realm of edge computing [32]. These attacks can manifest in various forms, targeting individual devices, networks, or even entire systems. In some instances, a single system acts as the source of the attack, while in other cases, the attack originates from a distributed network of devices located in different locations. This latter type is known as a "distributed denial of service" (DDoS) attack. DDoS attacks present a greater challenge compared to conventional DoS attacks because they can camouflage themselves behind a multitude of unidentified devices, making them more difficult to mitigate. To provide a structured understanding of denial of service attacks in the context of edge computing, we have classified them into two categories: DDoS attacks and non-distributed DoS attacks. Fig. 4 presents the taxonomy of denial of service attacks in edge computing, offering an overview of the categories to which each attack type belongs. This taxonomy serves as a reference point for organizing and comprehending the different types of DoS attacks encountered in the edge computing landscape.

The taxonomy of denial of service (DoS) attacks, as depicted in Fig. 4, consists of two main types: distributed denial of service (DDoS) attacks and non-distributed denial of service attacks. DDoS attacks are further categorized into three distinct groups, namely volumetric, protocol-based, and application layer-based attacks [33]. Within the volumetric attack category, specific attack types include UDP flood, ICMP flood, DNS amplification, and MAC flood. These attacks aim to overwhelm the target system's resources, such as network bandwidth or memory, by inundating it with an excessive amount of malicious traffic. The protocol-based attack category encompasses attacks such as SYN flood, smurf DDoS, ping of death (POD), and low-rate denial of service (LDoS). These attacks exploit vulnerabilities in network protocols to exhaust system resources, disrupt communication, or cause system crashes. The application layer attack type focuses on attacks targeting the application layer of the network stack. Examples of application layer attacks include HTTP flood-based attacks, such as HTTP GET/POST flooding, slowloris, and zero-day DDoS attacks. These attacks aim to overwhelm the target application or server by exploiting its specific functionalities or resource limitations. On the other side of the taxonomy, non-distributed DoS attacks are classified into physical and protocol-based attacks. The physical attack type involves signal jamming, where the attacker disrupts wireless signals, causing communication interference and rendering the target devices unable to operate effectively. The protocol-based non-distributed DoS attack category includes attacks such as teardrop attacks and buffer overflow attacks. These attacks exploit vulnerabilities in network protocols or application software, aiming to cause system instability, crashes, or denial of service.

5.1. Distributed denial of service (DDoS)

As mentioned in the previous section, DoS attacks can be classified into distributed and non-distributed types. Distributed denial of service (DDoS) attacks have three sub-classifications, namely:

- Volumetric
- Protocol-based
- Application layer

In the subsequent sections, we present a detailed enumeration of these subcategories, providing concise descriptions, vulnerabilities, and threats associated with each variant of DDoS attacks. Furthermore, we discuss the standard countermeasures available based on existing tools and technologies. Additionally, we explore research-based solutions proposed by independent researchers worldwide, analyzing their approaches and mapping them according to the methods they employ.

5.1.1. Volumetric DDoS attacks

Volumetric DDoS attacks are characterized by their ability to overwhelm a system with a massive volume of malicious traffic, leading to service disruption and loss of availability. These attacks often leverage compromised or infected devices, which act as sources to generate a

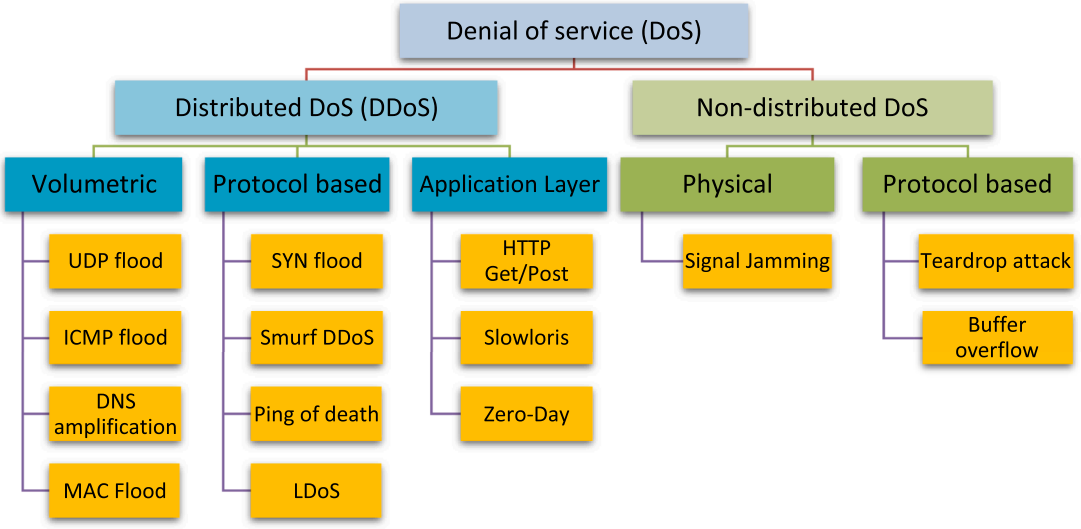


Fig. 4. DoS Taxonomy of edge computing.

high volume of random data packets. Among the various types of volumetric DDoS attacks, UDP floods and ICMP floods are the most commonly encountered. UDP floods involve sending a large number of User Datagram Protocol (UDP) packets to a target system or network. These packets are typically sent to random ports, causing the target to become overwhelmed by the sheer volume of incoming traffic. As a result, the system's resources, such as network bandwidth or processing power, become saturated, leading to a degradation of service or complete unavailability. ICMP floods, on the other hand, exploit the Internet Control Message Protocol (ICMP) to flood the target system with ICMP echo request (ping) packets. By sending an excessive number of ping requests to the target, the attacker aims to consume the system's resources, particularly its network bandwidth and processing capabilities. This influx of ICMP packets can lead to network congestion, latency, and unresponsiveness of the target system. In the subsequent sections, we will delve into further detail about these types of volumetric DDoS attacks, providing comprehensive descriptions of their attack mechanisms, vulnerabilities they exploit, potential threats they pose, and standard countermeasures available to mitigate their impact. Additionally, we will explore research-based solutions proposed by the research community to address these specific types of volumetric DDoS attacks.

a) UDP flood:

The UDP Flood DDoS attack (also known as the Fraggle Attack) usually occurs in the transport layer of edge computing systems. It uses User Datagram Protocol packets (UDP) and sends them out to the targeted server with the sole purpose of overwhelming its traffic and resources, which results in a massive load for the server and oftentimes exhausts the protective firewall as well. As a result, legitimate users face denial of service. Moreover, often attackers forge the IP of a victim and send requests to an edge server. The server then replies to the victim (instead of the attacker), which potentially launches a reflected DDoS attack.

5.1.1.1. Vulnerabilities. The UDP flood attack exploits a vulnerability inherent in the UDP (User Datagram Protocol) protocol. Unlike TCP (Transmission Control Protocol), which employs a 3-way handshake system for establishing connections, UDP is a connectionless protocol. This lack of connection establishment makes UDP susceptible to flooding attacks. In a UDP flood attack, the attacker inundates the target system with a high volume of UDP requests sent to various ports. Since UDP does not have a mechanism to verify or manage these requests, the target system becomes overwhelmed by the sheer number of incoming packets. This flood of UDP packets consumes the system's resources and communication bandwidth, resulting in degradation of service or even complete unavailability. To mitigate the impact of UDP flood attacks, it is crucial for systems and firewalls to have adequate transmission capacity. This capacity ensures that the network can handle the increased volume of incoming UDP packets without becoming overwhelmed. Implementing traffic monitoring and filtering mechanisms can help identify and block malicious UDP traffic, preventing it from reaching the target system.

5.1.1.2. Threats. UDP flood attacks can originate from malicious flooding agents, zombie systems, and compromised systems that have been hijacked by attackers [33]. These entities serve as sources to launch UDP assaults on targeted systems. In some cases, these attacks may utilize spoofed IP addresses to mask the true origin of the malicious traffic, making it difficult to trace back to the actual attackers. One significant challenge posed by UDP flood attacks is their ability to bypass resource-intensive firewalls or security measures that are not adequately equipped to handle such high-volume attacks. The sheer volume of UDP traffic generated by the attacking sources can overload system resources, leading to service disruption and unavailability. Furthermore, the emergence of botnets has further exacerbated the threat of UDP flood attacks. Botnets are networks of compromised devices that can be remotely controlled by attackers. These botnets, such as Mirai [34–39], Gafgyt [40–43], and BashLite [44–45], have been responsible for orchestrating large-scale UDP flood attacks. By leveraging the collective power of multiple compromised devices, botnets can generate massive amounts of UDP traffic, overwhelming targeted systems and amplifying the impact of the attacks.

5.1.1.3. Generic prevention / mitigation methods. To prevent or mitigate UDP flood attacks, several countermeasures can be implemented. The first line of defense is the implementation of filtration techniques to detect and handle large UDP packets and non-stateful UDP packets in critical ports. By monitoring network flow, edge routers and firewalls can identify unusual traffic patterns and detect spoofed packets using ingress filtering [46]. This helps to prevent malicious UDP packets from reaching their intended targets. Service providers can also play a crucial role in mitigating UDP flood attacks by utilizing deep packet inspection (DPI) techniques. DPI allows for the inspection and analysis of the content of network packets, enabling the detection of malicious traffic. In coordination with traffic rerouting, scrubbing servers can be employed to divert and filter out suspicious UDP packets, thus protecting the targeted systems [47]. At the client level, users can take proactive measures to defend their devices and intra-networks against UDP flood attacks. This can be achieved by utilizing scrubbing software, which helps to identify and eliminate malicious UDP traffic, ensuring the integrity and availability of the devices and the network.

5.1.1.4. Research-based solutions and limitations. Several solutions have been proposed to address UDP flood attacks in edge systems. In [48], the authors propose a solution for detecting DoS attacks in IoT environments, specifically focusing on 6LoWPAN devices. They utilize a Suricata-IDS probe combined with flood threshold controllers to identify incoming packets with suspicious parameters. However, the testing was limited to simulated attacks, and no real-world industrial network testing was conducted. Additionally, the solution may face challenges in handling dispersed sniffing from larger distributed networks. In [49], a lightweight queue shuffling technique is proposed to mitigate DDoS attacks on edge devices with limited resources. The technique involves discarding malicious flows to protect the system. However, the results are theoretical and simulation-based, lacking real-world data for application in data-centric or corporate networks. The authors in [50] propose SoftEdgeNet, a distributed architecture based on software-defined networking (SDN) and fog nodes. This solution aims to mitigate attacks by filtering data and providing fault tolerance. However, challenges related to latency and edge node placement are highlighted. In [51], machine learning algorithms are employed to classify benign IoT packets from malicious DDoS packets. The study includes the use of K-dimensional tree (KDTree), support vector machine (SVM), decision tree, random forest, and neural network algorithms. The experiments involved simulated TCP SYN flood, UDP flood, and HTTP GET attacks. While the detection technique shows promise, the authors acknowledge the possibility of compromised edge nodes being part of a DDoS attack, leaving the removal of the compromised devices as the remaining solution, which may not always be feasible. In [52], the authors propose a fog-assisted software-defined networking (SDN)-based intrusion detection/prevention system (IDPS) for edge networks. They utilize an SDN wrapper over a wireless sensor network (WSN) stack called SDNWISE, employing the Contiki OS-based Cooja simulator and the SDN emulator Mininet for testing. The system's intrusion detection capabilities are evaluated using the UNSW-NB15 dataset, and packet generation is done using IXIA PerfectStorm. The proposed system dynamically implements countermeasures throughout the network's edge nodes using an SDN controller and proper policies to mitigate malicious traffic. The

centralized architecture incorporates an E3ML multi-classifier with recurrent neural network (RNN), multilayer perceptron (MLP), and alternate decision tree (ADT) algorithms. Although the system shows promise, further testing in real networks is required, and the authors note that the performance of RNN and MLP can be unstable. In [53], the authors propose a software-defined Internet of Things (SD-IoT) system based on the SDx paradigm. They use the cosine similarity of vectors of incoming packets to differentiate between malicious and benign traffic flows. The algorithm identifies the infected node responsible for the attack and signals the controller to prevent further propagation of packets from that node. The proposed model outperforms the existing distributed parallel packet classification (DPPC) algorithm-based solution [54], as it can handle large amounts of traffic and quickly prevent such attacks. However, the system relies heavily on a singular controller, which introduces the risk of a single point of failure. Additionally, the experiment only considers UDP traffic, and the viability of the system for handling other types of attacks is not explored.

a) ICMP flood:

The ICMP flood DDoS attack, also known as the ping flood attack, is another type of flooding attack. ICMP (Internet Control Message Protocol) is a protocol that carries important information about IP, routing issues, and datagram processing errors. In an ICMP flood attack, the attacker leverages ICMP echo requests and sends them to the targeted layer 3 (internet layer) infrastructure devices. These devices typically respond with ICMP echo reply messages, acknowledging connectivity. However, when an overwhelming number of echo message responses flood the device, it becomes unresponsive to legitimate traffic. As a result, network administrators often choose to disable ICMP packet transmission to prevent congestion on critical layer 3 devices. When a host is targeted with an ICMP flooding attack, the excessive traffic overload renders the victim's devices unusable and can disrupt basic network functionalities [55].

5.1.1.5. Vulnerabilities. The ICMP protocol in itself poses a vulnerability in the context of ICMP flood attacks. The protocol provides valuable reconnaissance capabilities by carrying IP and routing information within a network. ICMP can be used to determine if a device (host) is online or not, and it provides other important details such as maximum transmission units, transmission limitations, default packet size, and more. Attackers exploit these parameters to launch malicious attacks on systems that lack proper packet filtering and are not adequately protected. In [56], Eden L. highlights ICMP as a versatile hacking tool that is often overlooked and misunderstood, emphasizing its potential for misuse in carrying out attacks.

5.1.1.6. Threats. ICMP flooding attacks can be initiated by various threat agents, including malicious botnets. These botnets send a large number of echo requests as part of the attack. Additionally, unmonitored broadcast pings from spoofed IP addresses and erroneous datagram headers originating from unverified sources can also contribute to the threat [57]. In the past, ICMP flooding attacks often involved the use of spoofed false IP addresses to mask the origin of the attack. However, a different trend has emerged in recent times, where attackers utilize an extensive network of un-spoofed bots to launch massive attacks on the targeted network or server. This approach focuses on overwhelming the target through sheer volume rather than relying solely on IP address spoofing [58].

5.1.1.7. Generic prevention / mitigation methods. To defend against an ICMP flood attack, one of the primary measures is to disable ICMP propagation within the network. By disabling ICMP acknowledgement on critical nodes and edge routers, the network can mitigate the impact of ICMP flooding. For non-critical nodes, it is recommended to limit the processing of ICMP requests and restrict the data packet size. This helps to prevent network congestion and potential disruptions. Additionally, implementing filtration mechanisms on routers can ensure that the network remains free from suspicious activities. Monitoring various ports on a schedule can help in overseeing the flow of traffic and detecting any abnormal patterns. Egress filtration on edge routers can also be employed to ensure that outbound traffic is within the expected limits and does not exceed the bandwidth threshold [59].

5.1.1.8. Research-based solutions and limitations. In [60] authors demonstrated an IoT-dense mechanism titled FlowGuard utilizing machine learning models Long short-term memory (LSTM) and convolutional neural network (CNN) for classifying malicious data and identifying malicious flows. In the proposed system, filtration rules are applied to identify malicious flows segregating from benign ones, a flow handler is used to identify flow variations, and rules are updated based on traffic observations. But the efficiency of the employed LSTM was not on par with standard IoT requirements as the experiment was done with a system with minimal configuration. Therefore, no data is present if that constraint can be overcome with a superior system. In [61] authors implemented a machine learning approach based on a self-organizing map (SOM) and used k-nearest-neighbor (k-NN), an instance-based non-parametric learning algorithm to detect ICMP attacks. It classifies traffic into separate groups (malicious and normal) and can mitigate attacks significantly in cloud platforms, but it lacks attack-handling capabilities in terms of intrinsic preventive measures. Moreover, it suffers from similar limitations that a centralized SDN system suffers. Therefore, in case the primary controller is compromised or overwhelmed, the full system suffers the aftermath.

a) DNS Amplification:

The DNS amplification attack, also known as the DNS reflection attack, targets multiple layers of edge computing systems, including the application, transport, and internet layers. In this type of attack, the attacker floods a cloud server, such as a DNS resolver, website, or host, with a large volume of fake DNS lookup requests. This overwhelms the server's resources, leading to service interruptions, system crashes, or the exhaustion of available bandwidth, potentially causing the targeted site to become unavailable. The attack takes advantage of connectionless protocols like UDP and manipulates a small DNS request into an amplified one by requesting numerous records associated with the targeted site, its subdomains, backup and mail servers, and aliases. This results in a response that is significantly larger, amplifying the load on the server by a factor of 10–50 [62]. Detecting and differentiating between fake and legitimate requests becomes challenging as spoofed IP addresses are used, making it difficult to identify the true source of the attack.

5.1.1.9. Vulnerabilities. DNS amplification/reflection attacks exploit various vulnerabilities in the network infrastructure. These attacks can be launched without the need for a botnet and require minimal resources. Several vulnerabilities contribute to the success of DNS amplification attacks. One vulnerability lies in the use of public resolvers instead of ISP-designated DNS servers by hosts or servers. Public resolvers can be more susceptible to abuse and amplification attacks compared to properly configured ISP resolvers. Inadequate configuration of perimeter firewalls is another vulnerability that can facilitate these attacks. If the firewall allows most traffic without proper filtration for DNS requests, it can inadvertently become a conduit for DNS amplification attacks. The absence of source IP verification methods on network gateway devices is another vulnerability that attackers can exploit. Without proper source IP verification, it becomes easier for attackers to spoof their IP addresses and disguise the origin of the attack. Additionally, the lack of client authorization recursion and the absence

of a response rate limiter (RRL) in a DNS resolver can make the resolver more vulnerable to amplification attacks [62].

5.1.1.10. Threats. Misconfigured DNS servers pose a significant threat to DNS amplification attacks. If a DNS server is not properly configured, it can be exploited by attackers to amplify their attack traffic. Unidentified traffic originating from external networks is another potential threat. Any traffic that cannot be attributed to legitimate sources should be carefully monitored and analyzed to identify possible malicious activity. DNS responses that do not originate from local DNS resolvers can also indicate a potential threat. If DNS responses are coming from unauthorised or unknown sources, it raises concerns about the integrity and authenticity of the responses. Malicious APIs and web applications can also be utilized by attackers to initiate DNS amplification attacks. These applications may have vulnerabilities that can be exploited to launch such attacks. According to the US Cybersecurity and Infrastructure Security Agency, there are approximately 27 million DNS resolvers in the internet. Alarmingly, it is estimated that around 25 million of these DNS resolvers are susceptible to being used as attack sources [63].

5.1.1.11. Generic prevention / mitigation methods. To counter DNS amplification attacks, several preventive measures can be implemented. One of the primary defenses is the verification of the source IP of inbound traffic, also known as ingress filtering. Internet Service Providers (ISPs) can employ filters to reject traffic coming from forged IP sources [64]. By verifying the authenticity of the source IP addresses, the ISP can prevent attackers from using spoofed IP addresses to initiate DNS amplification attacks. Another measure is to deactivate recursion on authoritative name servers. Recursive resolution of external users can be restricted, and the authoritative name servers can be configured to only serve as zone pointers for the respective domains [64]. This helps to prevent unauthorised recursive queries and minimizes the risk of DNS amplification attacks. Implementing a response rate limiter (RRL) on authoritative DNS servers is also effective. RRL limits the number of responses given to individual clients per second [64]. By imposing limits on the rate of responses, the impact of DNS amplification attacks can be mitigated. It is important to note that RRL is typically applied on authoritative DNS servers and does not impact internal DNS queries or recursive resolvers.

5.1.1.12. Research-based solutions and limitations. In [65], a two-stage framework is proposed for mitigating DDoS attacks using network function virtualization (NFV) and edge systems. The framework consists of a screening mechanism and a resource allocation mechanism. The screening mechanism filters traffic flow based on filtering algorithms, while the resource allocator assigns resources to virtual network functions (VNF) or virtual security functions (VSF) used for attack mitigation. However, the proposed scheme lacks actual hardware implementation and data to validate its effectiveness. In [66], a real-time volumetric detection (RTVD) scheme is introduced. This scheme comprises three components: a sliding time window for entropy calculation, a signal directional filter for early detection, and a quintile deviation checking (QuinDC) algorithm for identifying DDoS attacks. The scheme primarily focuses on volumetric attacks and may not be effective against other types of attacks, such as slowloris. The scheme utilizes public datasets, including the 1999 DARPA intrusion detection evaluation dataset, the 2009 DARPA DDOS dataset, and the UNB CIC DDoS 2019 evaluation dataset, which provide data on TCP SYN flood, UDP flood, and HTTP flood attack traffic. The IXIA BreakingPoint network tester is used for simulating attacks with varying frequencies. Although the scheme demonstrates accuracy with minimal delay, it requires enhancements to handle other types of DDoS attacks based on different methods, such as protocol-based or application-based attacks.

a) MAC flood:

Every network-enabled device is assigned a unique media access control (MAC) address, which is a 48-bit hexadecimal address provided by the manufacturer. In a switching network, a MAC table is maintained to facilitate communication between devices at the data link layer. However, attackers can flood the switch with a large number of invalid MAC addresses. This flood of invalid addresses overwhelms the switch, as it tries to update the MAC table with all these entries. As a result, the MAC table becomes full, preventing valid users from populating the table and leading to a denial of service.

5.1.1.13. Vulnerabilities. MAC flooding attacks can exploit vulnerabilities such as the lack of physical address authentication or validation systems, the absence of a counter on the total number of MAC addresses in a network, and the use of unmanaged switches and network hubs. These vulnerabilities make it easier for attackers to flood the MAC table of a switch and disrupt network connectivity. By addressing these vulnerabilities and implementing appropriate security measures, the risk of MAC flooding attacks can be mitigated.

5.1.1.14. Threats. MAC flooding attacks can be carried out using various tools, such as Macof, Ettercap, Yersinia, and THC Parasite. These tools flood vulnerable switch ports with malicious fake MAC addresses, overwhelming the MAC table and pushing legitimate MAC addresses out. This can cause the switch to enter fail-open mode, essentially functioning as a network hub. The flood of fake MAC addresses and the resulting broadcast storm disrupt regular network services and can lead to a denial of service.

5.1.1.15. Generic prevention / mitigation methods. To counter MAC flooding attacks, implementing port security on a manageable switch is considered a best practice. Port security allows administrators to specify the maximum number of MAC addresses allowed on each switch port, preventing unauthorised devices from flooding the MAC table [67]. Another effective measure is the use of an authentication, authorization, and accounting (AAA) server, such as TACACS or RADIUS. AAA servers authenticate users' identities, ensuring that only authorized devices can connect to the network [67]. Additionally, the implementation of IEEE 802.1X authentication can further enhance security against MAC flooding attacks. This mechanism verifies the identities of data-sending users based on their credential certificates, which are confirmed by a RADIUS server [68].

5.1.1.16. Research-based solutions and limitations. Since MAC flooding attacks are primarily associated with traditional network security issues and may not have specific research works focusing on edge computing systems, it is reasonable to omit independent research works in this context. The solutions and countermeasures for MAC flooding attacks can be addressed within the scope of traditional network security practices.

5.1.2. Protocol-based DDoS attacks

Protocol-based DDoS attacks exploit vulnerabilities in communication protocols by overwhelming targeted systems with malicious connection requests. Two commonly encountered types of protocol-based DDoS attacks are SYN flood and Smurf DDoS attacks. The following sections provide details on these attack types.

a) SYN flood:

In a SYN flood attack, the attacker takes advantage of the three-way handshake mechanism of the Transmission Control Protocol (TCP) to overwhelm the targeted system. The attacker initiates a large number of TCP connections by sending SYN (synchronization) packets to the victim's server. However, instead of completing the handshake by sending an ACK (acknowledgement) packet in response to the SYN-ACK

(synchronization-acknowledgement) packet from the server, the attacker simply leaves the connection in a half-open state without completing the final step of the handshake. By continuously sending a flood of SYN packets and not responding with ACK packets, the attacker consumes the system's resources, such as available TCP ports and memory buffers, preventing legitimate users from establishing new TCP connections. This leads to a denial of service, where the system becomes overwhelmed and unresponsive to legitimate traffic.

5.1.2.1. Vulnerabilities. According to the statistical data on DDoS attacks compiled by Kaspersky for the year 2020, it was observed that a significant majority of these attacks, specifically around 92.6 %, were categorized as SYN flooding attacks [69]. SYN flooding attacks primarily exploit the inherent vulnerabilities in the Transmission Control Protocol (TCP) three-way handshake connection establishment process. Systems that do not implement measures such as reduced SYN received timer or limited lifetime for half-open connections are particularly susceptible to such attacks. Additionally, systems or operating systems that do not adequately manage the backlog associated with TCP queues or fail to recycle the oldest half-open connections are also at risk of facing the adverse consequences of SYN flood attacks.

5.1.2.2. Threats. The prevalence of SYN flood attacks can be attributed to several prominent threat agents that initiate such malicious activities. These agents include unrecognized connection requests originating from foreign hosts whose identities and intentions are unknown. Additionally, the intrusion can be facilitated by a malicious botnet, such as the infamous Mirai botnet [70], which orchestrates coordinated attacks using compromised devices. Furthermore, the attack can be carried out by a group of infected host devices acting in a distributed manner. It is also worth noting that edge devices and sensor nodes with forged IP addresses can serve as potential sources for SYN flood attacks.

5.1.2.3. Generic prevention / mitigation methods. Various countermeasures can be employed to mitigate SYN flood attacks and ensure the robustness of network systems. One effective technique involves the implementation of micro records that are associated with each incoming SYN request [71]. This approach allows for the verification and tracking of SYN requests, thereby enabling the identification of legitimate connections and filtering out malicious ones. The use of cryptographic hashing techniques, specifically SYN cookies, can also play a vital role in countering SYN flood attacks [71]. By incorporating SYN cookies, the server can generate a unique token based on the client's SYN request, ensuring the authenticity of the request and preventing resource depletion due to excessive half-open connections. Another approach to mitigating SYN attacks is the utilization of TCP RESET cookies [72]. This mechanism involves a three-way handshake process where the client's legitimacy is validated before proceeding with the transmission of SYN data. By discarding spoofed IP packets, TCP RESET cookies provide an effective defense against SYN flood attacks. Administrators can further enhance security measures by implementing timeouts on TCP stacks, allowing for the timely release of memory occupied by existing connections [72]. Additionally, the strategic utilization of selective dropping on incoming connections can help alleviate the impact of SYN flood attacks by prioritizing and handling legitimate traffic while discarding malicious requests.

5.1.2.4. Research-based solutions and limitations. In their study, the authors of [73] presented a software-defined perimeter (SDP) based defense framework for multi-access edge computing (MEC). The framework utilized Open air interface (OAI) V 1.1 for the LTE core and radio network, along with an open-source SDP provided by Waverley Labs. Through simulations, they evaluated the effectiveness of the framework in mitigating SYN flood attacks by comparing network throughput with and without SDP. The results demonstrated promising DDoS attack mitigation capabilities. However, it is worth noting that the filtration process introduced a slight overhead (average delay of 0.0489 sec.), which may become more significant in real-world network infrastructures with high data transmission rates. Additionally, the framework's reliance on a centralized controller poses a single-point vulnerability, which should be carefully considered in practical deployments. In [74], the authors proposed a localized DDoS prevention framework called MECshield, designed to protect heterogeneous IoT networks. The framework employed self-organizing map (SOM) filters placed at the network edges, which were managed by a centralized controller responsible for traffic flow maintenance and traffic control policies. A local policy conductor facilitated the communication between the controller and the SOM filters, allowing for traffic mitigation based on the filter's training. The framework was evaluated using datasets such as CAIDA, NSL-KDD, and DARPA, and it demonstrated proficiency in countering various DDoS attack types, including flooding attacks, POD attacks, and botnet attacks. MECshield achieved improved detection rates and higher accuracy compared to existing SOM filters. However, the authors acknowledged limitations in the training duration of the SOM filters, which could impact the overall countermeasure process. Furthermore, the framework exhibited high CPU usage, although it managed to avoid bottleneck issues due to its distributed structure. Further optimization efforts may be necessary to address these limitations, especially when considering critical network nodes where interruptions in the countermeasure process could have catastrophic consequences.

a) Smurf DDoS:

In 1998, the University of Minnesota experienced a severe Smurf DDoS attack that had a lasting impact. The attack, which lasted for over an hour, resulted in significant data loss and network outages, causing disruptions across the state. This incident, often referred to as a "cyber-traffic jam," marked one of the early instances of a DDoS attack [75]. The Smurf DDoS attack exploits the use of a spoofed IP address to initiate ICMP packets sent to an IP broadcasting network. The network then responds with echo replies, which are broadcasted to all hosts within the network. As a result, each host sends a response to the spoofed IP host, leading to a massive influx of broadcast traffic that has the potential to overwhelm the network infrastructure.

5.1.2.5. Vulnerabilities. Systems and devices that lack adequate anti-malware solutions are particularly vulnerable to Smurf attacks. Additionally, misconfigured traffic monitoring policies on edge routers, which allow unrestricted IP-directed broadcast, and the absence of outbound IP filtration mechanisms further increase the susceptibility to Smurf attacks. These vulnerabilities create opportunities for attackers to exploit the network and launch devastating Smurf attacks.

5.1.2.6. Threats. Threat agents that can initiate Smurf attacks include the presence of Smurf malware, the utilization of Coremelt zombies, and the existence of unchecked IP broadcasting hosts with spoofed IP addresses. Coremelt zombies specifically contribute to distributed attacks, also referred to as Coremelt attacks. In such attacks, a network is targeted by multiple subverted machines that form separate zombie groups. These groups communicate with each other to coordinate and execute network-wide flooding, making it challenging to identify the precise origin of the attack due to the involvement of multiple zombie groups [76–77].

5.1.2.7. Generic prevention / mitigation methods. To prevent Smurf attacks, several countermeasures can be implemented. First and foremost, disabling IP-directed broadcast on edge routers is crucial. By disabling this feature, the network prevents the amplification of ICMP packets and eliminates the possibility of triggering a Smurf attack. Implementing

egress filtration on system devices can also contribute to preventing the vulnerability. With egress filtration, outgoing traffic is monitored and filtered, ensuring that any malicious packets are detected and blocked before they can leave the network perimeter. Additionally, enforcing strict traffic monitoring practices allows for the timely identification and mitigation of any suspicious or malicious activity. Disabling ICMP or blocking unauthorised ICMP requests on critical system devices provides an added layer of protection against Smurf DDoS attacks. By limiting the acceptance of ICMP packets, the network mitigates the risk of being exploited by this type of attack [76]. These preventive measures work in conjunction to safeguard network nodes from the detrimental impacts of Smurf attacks, fortifying network security and resilience.

5.1.2.8. Research-based solutions and limitations. In [78], the authors proposed a hybrid solution for mitigating DDoS attacks by leveraging the distributed computation capability of multi-access edge computing (MAEC) at edge nodes. The solution combines source-based methods and reactive mitigation techniques, utilizing the MACE-X controller to implement policies on the edge nodes. The controller collects traffic monitoring information from MACE-X clients and broadcasts warning messages to refine local policies for traffic regulation based on traffic anomalies. The proposed platform aims to mitigate volumetric attacks such as UDP floods and ICMP floods through prevention layers and a trust-based filtering system. It also addresses attacks like SYN flood, Ping of Death, and Smurf DDoS through adaptive policy implementation on the edge nodes. However, one limitation of this system is the potential processing delay, which could have significant consequences. Unfortunately, there is no empirical data provided in the research to assess the effectiveness of the proposed framework.

In [79], the authors developed a DDoS mitigation scheme for fog computing using a SCADA testbed. The scheme employed a three-layer data analysis architecture, including an inline traffic filter through the firewall, offline-based traffic analysis using virtualized network functions (NFV), and a centralized coordination system with a distributed local server for improved accuracy. The system was tested using a Modbus traffic simulator on Mero control systems. The results showed decent accuracy in detecting attacks; however, there was a 70 % accuracy rate with high latency (up to 235 ms) in fog-level Modbus traffic. This latency can be significant, especially for critical edge nodes or industrial valves. Consequently, further testing and optimization of the system are necessary to ensure its effectiveness in real-world scenarios.

a) Ping of death:

The Ping of Death (POD) attack is a type of internet layer attack that exploits a vulnerability in network devices, servers, or hosts by sending oversized ping packets. Normally, the maximum allowable size for an IPv4 packet with the IP header is 65,535 bytes. In a POD attack, the attacker sends a ping packet that exceeds this threshold size. When the target system attempts to reassemble the fragmented packet, it encounters an oversized packet, causing memory overload and potentially leading to system crashes. This attack takes advantage of the vulnerability in packet fragmentation and reassembly processes, exploiting the system's inability to handle excessively large packets.

5.1.2.9. Vulnerabilities. POD attacks can be launched by exploiting protocols that utilize IP datagrams such as ICMP, TCP, UDP, and others. Layer 3 devices that have ICMP response enabled are particularly susceptible to these attacks. Additionally, edge servers or systems that lack proper packet filtration mechanisms and have limited memory buffers unable to handle larger packets are also vulnerable to POD attacks. The combination of these factors creates a conducive environment for attackers to exploit the vulnerability and successfully execute POD attacks.

5.1.2.10. Threats. POD attacks can be initiated by hosts sending malformed or fragmented packets, as well as compromised systems that utilize spoofed IP addresses. To mitigate the risk of such attacks, system administrators should exercise strict control over the maximum packet size to prevent the fragmentation of large payloads by malicious actors. It is crucial for administrators to stay vigilant and promptly apply updates and patches released by developers to address any known vulnerabilities and strengthen the overall security of the system [80].

5.1.2.11. Generic prevention / mitigation methods. To prevent POD attacks, one effective measure is to disable ping responses, which eliminates the possibility of an attacker exploiting the ICMP protocol for such attacks. Advanced network mapping techniques available in layer 3 routers and system devices can be utilized to address any reconnaissance issues that may arise. Edge gateways can maintain oversight and control without the need for active ping responses. Preventive countermeasures include implementing external packet filtration to selectively block fragmented pings while allowing regular traffic to flow unhindered. A packet inspection system can be deployed to monitor incoming packet formation and verify compliance with conventional packet size constraints. Increasing the memory buffers in system devices can also facilitate this process, providing additional capacity to handle and process incoming packets [81].

5.1.2.12. Research-based solutions and limitations. In their study [82], the authors present an analytical framework for an intrusion detection system (IDS) that focuses on filtering packet lengths to identify suspicious traffic. The framework incorporates integer optimization techniques to minimize false alarms and includes considerations for missed detection probabilities. The authors propose that network administrators have the ability to adjust normal packet size thresholds, particularly during high-frequency attack scenarios, in order to reduce false alarm rates. However, a major limitation of the system is its reliance on manual adjustments, highlighting the need for an automated approach. Additionally, the authors suggest that future enhancements could involve implementing an edge server-based attack prevention mechanism. To validate the effectiveness of the framework, further empirical data from real networks is required beyond the current simulations conducted in the study.

a) Low-rate denial of service (LDoS):

LDoS (Low-rate Denial of Service) attacks represent a new breed of DDoS attacks that operate differently from traditional attacks. Unlike high-volume attacks, LDoS attacks exploit a vulnerability in the TCP congestion-control mechanism by periodically sending short bursts of packets over an extended period. This strategy aims to overflow the router's queue and degrade the overall quality of network traffic. LDoS attacks maintain a low traffic flow, typically around 10–20 % of the background traffic, making it challenging to differentiate these attacks from legitimate traffic using conventional detection methods. These attacks target all three layers of edge computing systems: application, transport, and internet. The attacker employs the Shrew attack technique, which induces significant packet loss and triggers retransmission timeout (RTO) for TCP connections. This leads to congestion and bottlenecks in network traffic, resulting in service unavailability for legitimate users.

5.1.2.13. Vulnerabilities. Lack of congestion control mechanisms in TCP routers makes them vulnerable to the effects of LDoS attacks [83]. Edge routers that lack RTO randomization and network flow monitoring mechanisms are also susceptible to the impact of LDoS attacks [84]. Additionally, operating systems that utilize default or lower minimum RTO values may be more prone to LDoS attacks [84].

5.1.2.14. Threats. Fake sessions, fragmented HTTP requests, and malicious TCP flows are some of the primary threat agents that can execute low-rate DoS attacks. These attackers utilize deceptive techniques to generate minimal traffic, making it difficult to differentiate their activities from legitimate network traffic. In the case of LDoS attacks, hackers exploit the slow-time scale and dynamic nature of TCP's retransmission time-out (RTO) mechanism. By rapidly transmitting packets and inducing repeated timeouts, they can disrupt normal network operations, leading to a denial of service situation.

5.1.2.15. Generic prevention / mitigation methods. One effective countermeasure against LDoS attacks is the implementation of a firewall at the network edge [85]. By monitoring inbound TCP traffic and blocking malicious packets, the firewall can prevent the attack from reaching the targeted system. Increasing the capacity of edge servers to handle incoming connection requests is another mitigation strategy, as it can help alleviate potential network congestion [85]. Edge routers with sufficient buffer storage capabilities can actively mitigate DDoS attacks by effectively managing network queues and packet flows, ensuring the smooth operation of TCP flows while protecting against malicious attacks [85]. To specifically address the vulnerability exploited by LDoS attacks, randomizing the retransmission time-out (RTO) value is recommended [86]. By introducing variability in the RTO value, the attacker's ability to predict TCP timeout instances is significantly hindered, making it more challenging to disrupt the system through repeated timeouts. Furthermore, having a larger buffer size to handle TCP traffic can also contribute to mitigating traffic congestion and improving network performance [86].

5.1.2.16. Research-based solutions and limitations. In [87], the authors proposed an algorithm that combines the power spectral density entropy function and support vector machine (SVM) learning to detect malicious low-rate denial of service (LDoS) traffic. Using the KDD99 dataset, which includes various types of DoS attacks, the authors computed the power spectral density entropy for the nearest four items and added eight features for detection. The features were normalized and combined with SVM, resulting in a 99.19 % detection rate for LDoS attacks with O (n log n) time complexity. However, this work focused only on smurf-based attacks and did not propose any prevention mechanisms. In [88], the authors employed multi-feature fusion and convolutional neural network (CNN) to recognize low-rate LDoS attacks from benign traffic. They combined several features to generate a feature map, which was then utilized by deep learning techniques. The method was validated using the NS2 simulation platform and compared with other detection methods. The results showed a detection rate above 88 % for most methods, with the highest false negative rate of 16.7 %. The authors acknowledged the need for further improvements and expressed interest in expanding their work to larger-scale real networks to account for network variables. It is worth noting that this work also focused solely on detection and did not propose prevention mechanisms. In [89], the authors proposed a method called FR-RED (fractal residual-based real-time detection) for LDoS attack detection. The approach analyzed fractal residuals of network traffic using the R/S algorithm to calculate Hurst parameters. Inspired by previous work [90], the FR-RED project consisted of modules for training, testing, and detection. The training module obtained the mean value of fractal residuals as a standard for regular traffic, while the testing module generated a decision eigenvector based on the training module. The detection module tracked LDoS timeframes by relating the extracted features to the other modules. The experiment conducted on the NS2 network simulator achieved a detection accuracy of 97.75 % with a false positive rate of 0.97 % and a false negative rate of 3.81 %. While this showed improvement over previous work, the small-scale nature of the network used necessitates further empirical data collection in real-time scenarios to validate the approach's effectiveness.

5.1.3. Application layer DDoS attacks

Application layer DDoS attacks, specifically HTTP DDoS attacks, operate at Layer 7 of the OSI model. These attacks utilize the Hypertext Transfer Protocol (HTTP) to flood targeted servers or web applications with malicious HTTP requests. Attackers often leverage botnets to amplify their attack by sending a large volume of HTTP requests to overwhelm the target's resources [91]. These attacks can result in partial denial of service or completely exhaust the device's capacity to handle legitimate HTTP connections. The following segments shed some light on such application layer DDoS attack types.

a) HTTP GET/POST:

The HTTP GET/POST attack, also known as the HTTP flood attack, is a prevalent type of application layer DDoS attack. This Layer 7 attack targets edge servers or web-based applications by flooding them with HTTP requests. Attackers utilize malicious botnets, such as Mirai, Gafgyt, and BashLite, to amplify their attacks. In an HTTP GET attack, coordinated requests are sent from multiple infected devices to access or download files from the targeted server. This flood of requests overwhelms the server, leading to a loss of availability for legitimate users. In an HTTP POST attack, the attacker sends numerous POST requests through online forms, which are then processed by the server's database. The high volume of POST requests saturates the database processing capacity, causing service disruptions [92].

5.1.3.1. Vulnerabilities. HTTP attacks can be challenging to detect because they often resemble legitimate URL requests made by regular users. Traditional rate-centric defense mechanisms that rely on monitoring traffic volume are ineffective against these types of attacks since they involve relatively low traffic levels. Web portals and applications that lack JavaScript-based bot detection mechanisms, such as CAPTCHAs and server/client-side behavioral analysis, are particularly vulnerable to HTTP POST attacks. Additionally, servers like Apache and Nginx that do not implement measures such as HTTP timeouts, connection limiters, header size restrictions, or user backlogs are more susceptible to HTTP attacks.

5.1.3.2. Threats. Botnets play a significant role in orchestrating HTTP flood attacks, leveraging a large number of compromised devices that can range from thousands to millions of zombie hosts. Notable botnets like Mirai, Gafgyt, and BashLite are commonly associated with these types of attacks, penetrating through the application layers. In addition to botnets, various other threat agents contribute to HTTP flood attacks in edge systems. These include malware such as Trojan horses, unsupported and unauthorised HTTP requests, malicious payload injection through HTTP headers, host override headers, blacklisted domains, and servers that fall victim to web cache poisoning. These threat agents exploit vulnerabilities in the HTTP protocol and web application infrastructure to launch massive floods of HTTP requests, leading to service disruptions or unavailability [93].

5.1.3.3. Generic prevention / mitigation methods. To prevent HTTP attacks, several countermeasures can be implemented. First, maintaining a connection/IP database can be effective in blocking connections from blacklisted IPs automatically. By identifying and blacklisting malicious IPs, the system can prevent them from accessing the network or application. Setting an appropriate HTTP connection timeout is another crucial measure. The connection timeout value should be determined based on connection length statistics, ensuring it is slightly greater than the median lifetime of legitimate client connections. This helps to terminate idle or prolonged connections, reducing the impact of potential attacks. Implementing a well-defined incoming HTTP data rate control mechanism can be effective in preventing HTTP attacks. By defining a threshold rate for incoming HTTP data, connections that

exceed this rate can be dropped, preventing excessive traffic from overwhelming the server or application. Furthermore, the use of packet filters and connection backlogs for incoming traffic can contribute to preventing HTTP attacks. Packet filters can track and filter out malicious traffic, identifying requests that exhibit suspicious behavior or characteristics [94]. Connection backlogs can help manage and prioritize incoming connections, allowing the system to track and filter potential malicious traffic effectively [95]. Additionally, utilizing SDN platform-based IP traceback methods can prove to be effective in countering HTTP attacks. These methods leverage the capabilities of Software-Defined Networking (SDN) to trace the origin of malicious traffic, enabling better identification and mitigation of attacks.

5.1.3.4. Research-based solutions and limitations. In [96], the authors presented CODE4MEC, a defense framework for MEC nodes that utilizes a combination of control functions to enforce the defense mechanism [97]. The system tracks network traffic changes and employs an online combinational auction method for code scheduling, a vIPS-orthogonal CODE coordination scheme, and detection schemes SENTRY and BotBuster to mitigate HTTP flood attacks. However, the authors acknowledged the need for improvements in the response delay of the non-local CODE environment setup as part of their future work. In [98], the authors introduced ShadowNet, an architecture that leverages web services and edge functions to identify malicious DDoS packets. The ShadowNet web service, stored in the cloud, establishes connections with edge nodes through the ShadowNet fast path. The edge functions send data profile sketches of incoming packets to the ShadowNet web service through the fast path and aggregate IoT traffic information. The system demonstrated faster UDP attack detection compared to existing models, but it has limitations. Multiple network segment implementations are necessary to be effective, which may introduce geographical replications affecting fast-path assumptions. Additionally, the system prioritizes detection speed over accuracy, making it unable to differentiate between a legitimate attack and a flash crowd. Proposed techniques are being explored to address this limitation. Regarding botnet attacks on edge systems, [99] presented a sparsity representation framework for botnet attack detection on IoT edge devices. The framework utilized reconstruction error thresholding and outperformed a single hidden layer autoencoder using the N-BaIoT dataset. [100] constructed the N-BaIoT dataset and proposed a botnet detection method using deep autoencoders. In [101], the authors introduced EDIMA, a machine learning-based botnet detection framework designed to detect botnets before they launch attacks. The framework utilized supervised machine learning algorithms, including Gaussian naive Bayes, support vector machine (SVM), and random forest. The experiments conducted using multiple IoT devices connected to an edge router showed low false positive rates, and the random forest model achieved the best results.

a) Slowloris:

The slowloris attack is a type of application layer attack that involves the use of partial HTTP requests. It is characterized by the attacker sending out numerous HTTP connection requests and intentionally keeping them open without completing them. This attack exploits the limitation of server's ability to handle a limited number of simultaneous connections. The goal of the slowloris attack is to exhaust the server's resources and prevent it from accepting new connections. By keeping the connections open, the attacker consumes server resources such as connection slots, memory, and processing power. This leads to a denial of service condition where legitimate users are unable to establish new connections or access the targeted service. The unique aspect of the slowloris attack is its ability to inflict damage with minimal bandwidth and resources. The attacker strategically sends partial requests, keeping the connections alive with minimal data transfer. This makes it challenging to detect and mitigate the attack using traditional rate-based defense mechanisms. To defend against slowloris attacks, various techniques can be employed. One approach is to implement server-side configurations that limit the number of connections allowed per client or enforce timeouts on idle connections. Additionally, network-level mitigation techniques such as traffic filtering, load balancing, and rate limiting can be effective in mitigating slowloris attacks.

5.1.3.5. Vulnerabilities. Servers that lack proper connection restrictions and timeout settings are vulnerable to slowloris attacks. One major vulnerability is the allowance of a single IP to establish multiple connections. When servers do not enforce limitations on the number of connections from a single IP address, attackers can exploit this by opening multiple connections and keeping them open, consuming server resources and preventing new connections from being accepted. Additionally, servers that do not have mechanisms in place to restrict the number of concurrent incoming HTTP connections are at risk. Slowloris attackers take advantage of this by opening numerous connections without completing them, leading to resource exhaustion. Another vulnerability is the absence of timeout settings on HTTP headers. When servers do not configure timeout settings for HTTP headers received from clients, they are unable to terminate idle connections, making them more susceptible to slowloris attacks.

5.1.3.6. Threats. Slowloris attacks can be initiated by threat agents such as incomplete HTTP connections, malicious HTTP botnets, Android botnets like WireX [102], and web application-based Trojans [103]. In these attacks, the attacker takes advantage of vulnerable systems by keeping numerous HTTP connections open without completing them, effectively tying up server resources and preventing new connections from being established. This type of attack can cause significant disruption and impact the availability of targeted services. In addition to the traditional slowloris attack, variations have been observed, such as the case where thousands of connections were opened to the Gmail API with message-sending requests, and then all the connections were completed simultaneously. This resulted in Gmail sending out a large volume of bulk messages [104], potentially causing email service disruptions and impacting user experience.

5.1.3.7. Generic prevention / mitigation methods. To prevent or mitigate slowloris attacks, several common countermeasures can be implemented. One effective approach is to increase the client capacity of an edge server, allowing it to handle a larger number of simultaneous connections. At the same time, implementing connection lifetime restrictions can ensure that connections from individual clients have a specified duration and are not kept open indefinitely [105]. This helps prevent attackers from monopolizing server resources by keeping connections open for extended periods. Another important countermeasure is to limit the maximum number of connections allowed from a single IP address. By implementing this restriction, organizations can prevent attackers from launching slowloris attacks using multiple connections from the same source [105]. In addition, enforcing a minimum bandwidth requirement for packet transfer rate on each connection can be effective in mitigating slowloris attacks. By ensuring that a certain level of data is being transferred within a specified time frame, organizations can identify and block connections that exhibit unusually slow data transmission rates, which are indicative of a slowloris attack [105]. Furthermore, deploying a reverse proxy server immediately after the network firewall can provide an additional layer of defense. The reverse proxy server can analyze inbound connections, verify their legitimacy, and direct legitimate requests to the appropriate destination servers, while filtering out potentially malicious slowloris attack traffic [105].

5.1.3.8. Research-based solutions and limitations. In [106], the authors presented a mechanism that utilized side-channel information to identify traffic anomalies on IoT devices. They conducted a simulation using

a Raspberry Pi 4 as an IoT device and Amazon's Alexa as an IoT application. The attack deployment was performed using another Raspberry Pi device running Kali Linux, which launched various attacks including slowloris attacks. The collected data was then processed using a support vector machine (SVM) algorithm for attack classification. However, the system lacked solid preventive mechanisms, and the SVM algorithm achieved an accuracy of only 77.5 %, indicating a need for further improvement. In [107], the authors demonstrated a distributed cloud-native edge architecture that aimed to provide constant multi-view visibility using onion-ring visualization. This architecture relied on open-source tools and offered resource visibility while maintaining data integrity during connection failures. The system utilized the smart multi-view visibility framework (MVF) to reduce network load, making it an effective tool for countering application-based attacks such as slowloris. The combination of IO visor and PerfSONAR's active monitoring allowed administrators to detect resource issues, while Apache Zookeeper facilitated distributed synchronization. In [108], the authors employed long short-term memory (LSTM)-based detection techniques and fine-tuned hyperparameters to develop a robust deep learning model. The detection model successfully identified DoS attack simulators like GoldenEye, Heartbleed, Hulk, as well as slow HTTP DoS attacks like slowloris. The performance of the model was evaluated using metrics such as area under the curve (AUC), receiver operating characteristics (ROC), and F1 scores, which are crucial for evaluating the performance of machine learning models with imbalanced datasets. However, the system did not provide any prevention techniques and exhibited lower accuracy in detecting botnet attacks.

a) Zero-day DDoS:

A zero-day DDoS attack is a sophisticated form of DDoS attack that takes advantage of security vulnerabilities in software, hardware, or protocols that were previously unknown to experts, developers, or the community. The attacker identifies and exploits loopholes in the system, residing either on an edge device or a cloud server, to carry out the attack. By exploiting these vulnerabilities, the attacker can cause memory corruption or trigger specific conditions that result in the shutdown of the entire system. What makes zero-day DDoS attacks particularly challenging is that they leverage previously unknown exploits, which means that there are no established countermeasures or patches available to protect against them. As a result, detecting and mitigating these attacks becomes extremely difficult, as traditional defense mechanisms are not designed to address these specific vulnerabilities. It often requires a proactive and dynamic approach to identify and respond to zero-day DDoS attacks, involving continuous monitoring, vulnerability assessment, and rapid response to newly discovered exploits [109].

5.1.3.9. Vulnerabilities. In a zero-day DDoS attack, the attacker takes advantage of an unintentional software glitch, code error, or hardware flaw that was previously unknown to the system developers or security experts. By exploiting these flaws, the attacker can launch a series of attacks and potentially compromise an entire network [110]. Systems that lack proper mechanisms for vulnerability checking or rollback methods are particularly susceptible to such attacks. Additionally, networks that do not have effective web application traffic monitoring in place are more vulnerable to zero-day DDoS attacks, as they may not be able to detect and respond to the malicious traffic in a timely manner. Therefore, it is crucial for organizations to implement robust security measures, such as regular vulnerability assessments, timely software updates, and proactive monitoring, to mitigate the risks associated with zero-day DDoS attacks.

5.1.3.10. Threats. Malicious web browser extensions, malware payloads delivered through phishing emails or malicious web links, and compromised systems with undetected exploits are significant threat agents that can initiate zero-day DDoS attacks. These agents typically target unsuspecting users who have inadequate security measures in place. Once inside the system, they exploit existing vulnerabilities, including software flaws and glitches, which can result in the complete failure of the targeted system. It is important for users to exercise caution when interacting with unknown websites or email attachments, keep their systems and applications up to date with the latest security patches, and use reputable antivirus and anti-malware software to detect and prevent such threats.

5.1.3.11. Generic prevention / mitigation methods. To effectively counter zero-day DDoS attacks, it is crucial to be prepared and proactive in addressing vulnerabilities that may be exploited. Implementing rollback systems allows for swift resolution of identified glitches or flaws through rapid patch deployment. Additionally, deploying a web application firewall at the network edge provides a strong defense against incoming malicious traffic, helping to prevent the exploitation of unknown vulnerabilities. Another valuable countermeasure is the use of runtime application self-protection (RASP), which detects and distinguishes between safe and malicious application request payloads. Employing secure web extensions, regularly updating software, and avoiding opening unknown links are also important practices to mitigate the risk of falling victim to phishing attacks.

5.1.3.12. Research-based solutions and limitations. In [111], the authors proposed the use of federated learning (FL) as a countermeasure against zero-day botnet attacks, addressing concerns regarding data privacy. They preferred FL over centralized deep learning (CDL) models to ensure the privacy of data. The project utilized a deep neural network (DNN) for traffic classification, which proved effective in detecting traffic anomalies in edge and IoT networks [112–114]. The federated averaging (FedAvg) algorithm was employed to aggregate multiple remotely coordinated DNN models from various IoT devices and generate a global DNN model. The researchers conducted simulations of zero-day botnet attacks using datasets such as BotIoT and N-BaIoT, comparing the performance of FL with other models including CDL, LDL, and DDL. The results demonstrated the promise of the FL approach.

5.2. Non-Distributed denial of service (DoS)

In the non-distributed DoS attack category several attack variations can be observed. These attacks originate from a single point and can attack a single system or an edge node thwarting its functionalities and causing a denial of service. Following are some of the variations:

a) Signal Jamming:

A jamming attack is a physical type of DoS attack where external means such as high-range signals or electromagnetic energy are used causing interference to disrupt communication among wireless devices [115]. There are instances where unintentional interference might occur due to external noise, radio frequency (RF) interference, or collision. However, in the case of a jamming attack, this is mostly the result of a signal that is high in energy, efficient, and anti-jamming resistant, with very low detection probabilities [116]. This kind of attack can be catastrophic, as it can be performed with very simple tools [117]. Jamming attacks can be directed toward sensor nodes to disrupt physical signals, deplete node resources such as battery life, and bandwidth, and force users to re-authenticate, potentially causing interruptions and a state of denial of service. It can even lead to other types of attacks, such as offline dictionary attacks and man-in-the-middle (MITM) attacks. [118].

5.2.1. VulnerabilitiesE

Usage of non-resilient RF devices, usage of RF devices with similar

bands, and having no anti-jamming mechanisms are some of the major vulnerabilities that can lead to signal jamming attacks.

5.2.2. Threats

Various agents can be considered threats for launching signal jamming attacks or can cause signal jamming in general. Anonymous RF devices can cause active signal jamming, which can be used by a hacker to cause communication interruptions. On the other hand, various appliances and everyday devices can cause signal jamming with RF interference, such as microwaves, cordless phones, fluorescent lights, RF video cameras, etc.

5.2.3. Generic prevention / mitigation methods

To mitigate the risk of jamming attacks in wireless sensor networks (WSNs), several measures can be implemented. One approach is to ensure that WSN devices are not placed in close proximity to devices that can generate interference, such as microwave ovens, fluorescent lights, RF video cameras, and cordless phones. By keeping a distance from potential sources of interference, the likelihood of jamming incidents can be reduced. Another mitigation strategy involves utilizing the 5 GHz band for devices located closer to the edge router. The 5 GHz band offers a shorter range but provides faster transmission rates. By deploying devices that are less susceptible to interference on the 5 GHz band, the network can minimize the impact of potential jamming attacks. Employing jamming-resistant receivers is another important measure to safeguard WSNs against jamming attacks. These receivers are designed to detect and filter out jamming signals, allowing the network to maintain proper communication despite the presence of interference. Furthermore, deploying redundant sensor nodes in critical areas can enhance the network's resilience against jamming attacks. Redundancy ensures that even if certain nodes are affected by jamming, there are alternative paths and nodes available to maintain network connectivity and functionality.

5.2.4. Research-based solution and limitation

In [119], the authors proposed a solution to counter stochastic jamming in edge systems using the multi-armed bandit (MAB) architectural framework. The proposed framework, called SAVE-S (Security-Aware edge serVer sElection under stochastic jammer), leverages the MAB algorithm to select suitable edge servers for offloading computational tasks [120–121]. The algorithm aims to maximize resource utilization and mitigate the impact of jamming attacks without relying on specific spectrum or data transmission capabilities. The authors implemented a prototype of the SAVE-S algorithm and conducted tests using both real and synthetic datasets. The results demonstrated promising performance in mitigating the effects of stochastic jamming attacks. However, the authors acknowledge the need for more comprehensive testing on a larger scale, considering different types of jamming attacks such as reactive attacks, constant attacks, random attacks, and periodic jamming attacks.

a) Teardrop Attack:

A teardrop attack is a type of DoS attack that exploits a vulnerability in the TCP/IP protocol stack, specifically targeting the internet layer. The attack involves sending fragmented packets to a targeted host device or server. In the IP headers of these packets, there is a field called "fragment offset" that indicates the position of the fragmented data relative to the original data packet. In a teardrop attack, the attacker manipulates this field in a way that causes overlapping of the fragments. When the targeted system attempts to reassemble these fragmented packets, the overlapping fragments create inconsistencies in the reassembly process. This leads to memory corruption or system crashes, resulting in a denial of service condition. Teardrop attacks were more prevalent on older Windows systems and Linux kernel versions before 2.1.63, as these systems had vulnerabilities that allowed such attacks to be successful. However, it is important to note that teardrop attacks have resurfaced and can still be an effective attack vector if the targeted system has not been properly patched and protected.

5.2.5. Vulnerabilities

Older systems with outdated operating systems, unpatched systems with data assembly bugs, devices lacking regular system updates, and networks lacking ingress filtering are some of the vulnerabilities that can be exploited to launch teardrop attacks on edge systems.

5.2.6. Threats

Anonymous packet senders with unknown IP origins and fragmented or bug-laden data packets are two of the major threats that might cause teardrop attacks. These can cause packet-overlapping errors, frame alignment issues, and re-assembly bugs that result in OS crashes or the shutdown of the application that is trying to handle the packet.

5.2.7. Generic prevention / mitigation methods

To prevent teardrop attacks, organizations can employ the following countermeasures. Firstly, implementing a proper firewall that employs ingress filtering at the network layer can effectively discard malicious or junk packets, thereby preventing teardrop attacks from reaching the targeted systems. It is crucial to ensure that systems are running the latest operating systems and have the necessary patches and updates applied. Modern operating systems often include security enhancements and fixes that make them more resilient against teardrop attacks. Therefore, organizations should regularly update their systems to ensure they have the latest protections in place. For older and obsolete systems that may still be in use, it is recommended to replace them with newer, more secure systems. Outdated systems are more likely to have vulnerabilities that can be exploited by teardrop attacks and other forms of cyber threats. In critical network segments, deploying a secure proxy server can be an effective preventive measure against teardrop attacks. The secure proxy can filter and validate incoming packets before they reach the target systems, helping to block any malicious or fragmented packets that could potentially trigger a teardrop attack. In the event of an ongoing teardrop attack, having a backup caching server can be beneficial. This server can serve static backup contents, ensuring that essential operations can continue even if the primary systems are experiencing disruptions or failures caused by the attack.

5.2.8. Research-based solution and limitation

In [117], the authors presented a teardrop attack detection system based on machine learning techniques. The experiment focused on an IPv6 network and compared the proposed system with two traditional algorithms: TAD-KNN (topological anomaly detection-k-nearest neighbors) and GR-AD-KNN (information gain ratio average distance KNN). The authors highlighted a limitation of the traditional KNN algorithm, which is the instability and misjudgment in classifying smaller groups when the k value is not properly selected. They referred to this limitation as the "small group classification disadvantage." To address this issue, they introduced the GR-AD-KNN algorithm, which improved the detection performance of DoS attacks, specifically teardrop attacks in this case. The experimental results demonstrated that the GR-AD-KNN algorithm achieved a higher F1 score compared to the TAD-KNN algorithm in the detection of teardrop attacks. The authors conducted multiple averages to ensure the robustness of the results. However, one limitation of this work is that the evaluation focused solely on teardrop attacks and did not consider more severe DoS threats. Teardrop attacks are relatively less severe compared to other types of DoS attacks. Therefore, further evaluation and consideration of a wider range of DoS threats would enhance the effectiveness and applicability of the proposed detection system.

a) Buffer overflow

A buffer overflow attack is a type of protocol-based DoS attack that targets a bug or vulnerability in software, specifically in the application layer. It takes advantage of the absence of array-bounds checking in programming languages like C/C++. When a program attempts to write more data to a buffer than it can handle, the excess data overflows into neighboring memory addresses. Hackers exploit this vulnerability by intentionally overloading the buffer with excessive data, causing it to overwrite adjacent memory elements. By manipulating the contents of the buffer, they can alter the program's execution path and potentially gain unauthorised access to sensitive user data. In some cases, these attacks can lead to a system crash or compromise critical security services. It is worth noting that certain programming languages like Java, Perl, and JavaScript have built-in mechanisms to prevent buffer overflow vulnerabilities, making them less susceptible to such attacks.

5.2.9. Vulnerabilities

Systems based on C/C++ or Fortran, system code that is reliant on external data, and errors or vulnerabilities in codes are some of the major elements that can act as catalysts for buffer overflow attacks.

5.2.10. Threats

Various threat agents can cause buffer overflow attacks. Malicious code that exploits programs while triggering new actions and programs that flood memory space [123] are some of the major types that can cause major buffer overflow issues in an unprotected system.

5.2.11. Generic prevention / mitigation methods

Using programming languages with built-in countermeasures against buffer overflow attacks, such as Python or Java, is indeed a good preventive measure [123]. These languages have mechanisms in place to handle memory management and array bounds checking, reducing the risk of buffer overflow vulnerabilities. Modern operating systems also provide runtime protections that can significantly mitigate buffer overflow attacks. One such mechanism is Address Space Layout Randomization (ASLR) [124]. ASLR randomizes the memory locations of key components of a program, including the stack, heap, libraries, and executables, making it difficult for an attacker to predict and exploit specific memory addresses. Another method is the use of non-executable memory areas. By marking certain memory regions as non-executable, the operating system prevents malicious code from being executed from those areas, reducing the impact of buffer overflow attacks. Structured Exception Handler Overwrite Protection (SEHOP) is another defense mechanism employed by modern operating systems. It protects against exploits that target the Structured Exception Handler (SEH), which is a component responsible for handling exceptions in Windows systems. SEHOP prevents malicious actors from overwriting SEH records and executing arbitrary code [125].

5.2.12. Research-based solution and limitation

In [126], the authors present a system called CloudSEC, which aims to address lateral movement attacks in hierarchical network environments within the edge-cloud architecture. The system consists of two main components: EventTracker and AlertCorrelator. The EventTracker component monitors network activities and intrusions by analyzing event logs. It tracks and analyzes various events occurring within the network to detect potential security threats. The AlertCorrelator component aggregates alerts generated by distributed intrusion detection sensors in the edge-cloud environment. It collects and correlates these alerts to provide a comprehensive view of the network security status and identify potential attacks. The authors conducted experiments using different datasets to evaluate the system's performance. In the first category, they used the LLDOS1.0 and LLDOS2.0.2 datasets generated by MIT Lincoln Laboratories. These datasets simulated network environments consisting of multiple hosts, DMZs, and various operating systems such as Windows, Linux, SunOS, and Solaris. The second category utilized the Treasure Hunt dataset generated by the University of California Santa Barbara, which included subnetworks with MySQL servers, file servers, and web servers. The experiments focused on detecting buffer overflow attacks and various DDoS attacks. The evaluation metrics were based on confidence intervals and events per day. The results demonstrated the effectiveness of the CloudSEC system in detecting these attacks and providing a robust detection framework.

6. Denial of service attacks in edge layers

In this section, we provide a comprehensive overview of the attack types discussed in earlier sections and map them to their respective edge computing layers. The figure (Fig. 5) illustrates the mapping, while tables (2, 3, 4, and 5) summarize the attack types, vulnerabilities, threats, countermeasures, recent research-based solutions, and their approaches and limitations. Starting from the top layer, the application layer is vulnerable to volumetric attacks such as HTTP flood and DNS amplification attacks. It is also prone to botnet attacks due to weak OS security, unreliable software, and access to malicious websites. Additionally, it can be targeted by less frequent attacks like buffer overflow, slowloris, and LDoS attacks. Moving to the transport layer, it is also susceptible to LDoS and DNS amplification attacks. However, the most common attack types at this layer are volumetric attacks like UDP flood and SYN flood attacks. According to Verisign's Q2 2018 DDoS report, UDP-based attacks accounted for approximately 52 % of DDoS attacks, while TCP-based attacks (such as SYN flood) accounted for around 26 % [127]. The Internet layer is affected by attacks such as ICMP flood, ping of death, and teardrop attacks [128]. These attacks exploit vulnerabilities associated with fragmented IP packets, erroneous datagrams, and spoofed IP broadcasts. To protect this layer, proper ingress filtration of IP packets is crucial. Moving down to the data link layer (layer 2), MAC flooding is a significant concern. Attacks targeting layer 2 devices, such as switches and network hubs, can cause area-wide network outages if proper authentication systems or countermeasures are not in place. Finally, at the physical layer, the primary target is jamming-type DoS attacks. These attacks disrupt communication by emitting high-range signals or electromagnetic energy, resulting in interference and network disruption. Fig. 5 provides an overview of the attacks categorized by their respective edge computing layers. Tables 2, 3, 4, and 5 further summarize the attack types, vulnerabilities, threats, countermeasures, recent research-based solutions, and their approaches and limitations, providing a comprehensive reference for understanding and addressing these attacks in edge computing environments.

Now that we have demonstrated the attacks by layers, we can summarize the solutions addressing each of them. In the following Table 2, we have compiled all the denial of service (DoS) and distributed denial of service (DDoS) attack categories, classifications, vulnerabilities, and threats in edge systems based on each edge computing layer. We have also included the solutions provided by recent research works associated with each attack type. The next table (Table 3), outlines the approach/techniques associated with each research-based solution with the associated references. The following table (Table 4) outlines the research-based solution list, mechanism, platform type, and the limitations of their proposed frameworks. Finally, Table 5 associates each research-based solution technique to respective denial of service attacks.

From Table 5, it is evident that, according to recent state-of-the-art research works, most volumetric attacks can be countered using ML-based classifiers [60] with flow handlers, self-organizing maps [61] often in conjunction with SDN environments [74]. The hybrid solution provided by MACE-X [78] that combines source-based methods and reactive mitigation techniques has proven effective in preventing most denial-of-service attack types. Its adaptive responsive framework offers data buffering and cashing imbued with pre-transcoding capabilities, yet it suffers from processing delays that impact the detection rate. For protocol-based attacks, a solution can be imposed based on the associated edge layer. In terms of Smurf DDoS (transport layer) an inline filtration is more apropos [79]; on the other hand, for ping of death

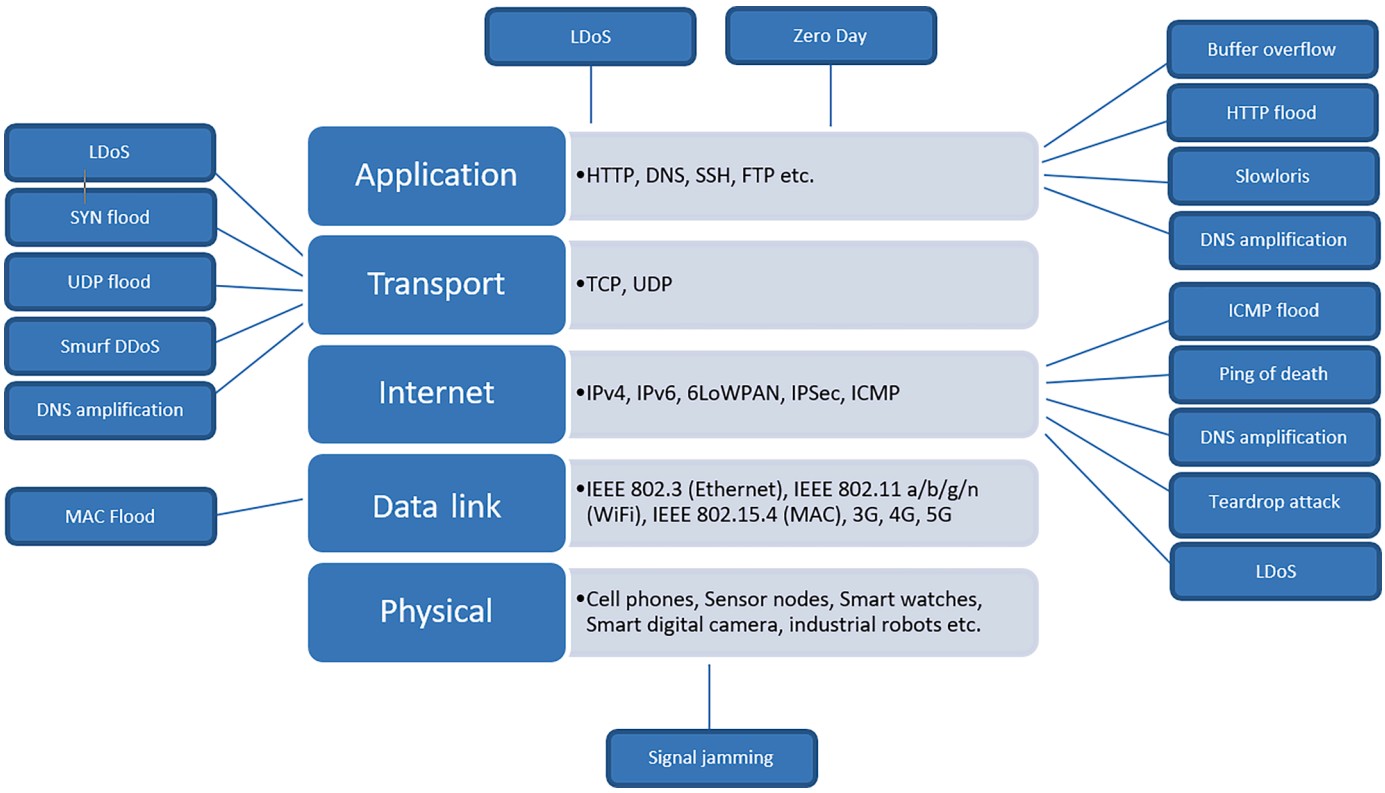


Fig. 5. Denial of service attacks on different edge layers.

(internet layer), packet length filtration is more efficacious [82]. In the case of application layer attacks, many of the supervised machine learning-based approaches are highly effective. Even privacy-preserving techniques such as federated learning approaches can be used in association with neural network combinations to thwart application layer attacks, which we tried to incorporate in our proposed architecture. In our architecture, we have conceptualized an SDN-based system that employs ACL policies imposed by a virtual firewall and a centralized controller that provides flow-handling capabilities. In the following segment, we discuss our proposed architecture with a brief visualization.

7. Proposed architecture

Our conceptual architecture (Fig. 6) addresses the limitations of existing solutions by combining federated learning (FL) and software-defined networking (SDN) principles. In the SDN platform, the control plane provides robust management capabilities, while the data plane offers flexibility for fine-grained control over network elements. The architecture includes a primary industry-standard controller that serves as the central management entity, enabling network supervision capabilities regulated by the SDN controller itself. The controller leverages several techniques to perform this supervision. They are as follows:

a Hybrid packet inspection:

The controller employs hybrid packet inspection method that combines both per-packet-based inspection and also statistical-based approaches. In both cases, network traffic goes through several filtration processes, such as hop count filtration, packet length filtration, source IP traceback, flow tracing, flow duration logging, etc. The statistical analysis of these parameters helps to identify malicious flow patterns from benign ones by feeding the information to the global ML model.

a Resilient AAA framework:

The utilization of the SDN platform grants the benefit of resilient AAA framework that performs three major operations. Robust authentication through password-protected user access, an ACL to authorize user privilege levels, and data logging to perform the accounting on network traffic [130]. Since these functionalities are performed within the controller itself, therefore no overhead is generated through additional hardware.

a Link supervision:

To ascertain the legitimacy of edge-node pairs' mutual authentication, channel probing is employed in order to gauge channel state information, which involves comparing the previous source node with the current one [131]. However, this is not an issue in our framework, as SDN by definition, has individual node information and uses way fewer physical nodes than a traditional physical network. Moreover, MAC validation mechanisms can be easily employed by extracting individual node information, which can be used to subvert MAC flooding-based attacks. Additionally, ARP inspection-based node validation can ensure that traffic is flowing from legitimate users in the network.

Additionally, the proposed system employs a honeypot server that acts as a decoy, diverting and capturing malicious intruders. The overall system is protected by a virtual inline firewall defined by the SDN controller. To avoid additional middleware and minimize system throughput impact, we rely on the virtual firewall as a defense mechanism [132,133]. During normal network operations or potential DDoS attacks, benign traffic is monitored and load-balanced (utilizing a weighted round-robin load balancer [134] to ensure smooth network activities, while malicious traffic is either discarded or directed to the honeypot server for statistical analysis and reference.

The controller extracts traffic data from each node in the network and converts it into Tensors objects with searchable attributes [135, 136]. Individual Tensor objects are identified with object tags that are shared with the controller. The controller converts these objects into local models. These local model weights and biases are then aggregated

Table 2
Denial of service attacks types, vulnerabilities, threats, and standard countermeasures in edge Systems.

Category	Classification	Layer	Vulnerabilities	Threats	Solutions
Volumetric	UDP flood	Transport	• UDP is a connectionless protocol • No systems to authenticate or filter inbound connections • Systems lacking provisions for flood mitigation • Firewalls lacking countermeasures are often overborne	• Malicious flooding agents, zombie systems, and compromised systems • Hosts embedded with spoofed IP addresses	[48,49,50,52,53, 54,60,61,78,51, 66,111,98]
Volumetric	ICMP flood	Internet	• ICMP reconnaissance capabilities with IP and routing information in a network. • ICMP provides other important information such as max transmission units, transmission limitations, default Packet size, etc. • An unmeasured system that lacks proper packet filtering	• Malicious botnets sending echo requests • Unmonitored broadcast pings from spoofed IP addresses • Erroneous datagram header originating from unverified sources etc. [60]	[60,61,78,74]
Volumetric	DNS amplification	Application, transport, internet	• Requires very little resource to launch an attack and does not need the help of a botnet • A substandard host or a server that uses public resolvers over ISP-designated DNS • A Poorly configured perimeters firewalls that allow most traffic without proper filtration for DNS requests • Lack of source IP verification methods on a network gateway device • Absence of client authorization recursion and response rate limiter (RRL) in a DNS resolver [62]	• A misconfigured Domain name system (DNS server), • Unidentified traffic originating from external network • DNS response not using local DNS resolvers • Malicious API and web applications • 25 million DNS resolvers out of the 27 million (worldwide) [63]	[60,66,74]
Volumetric	MAC flood	Data link	• Lack of physical address authentication or validation system • No counter on total MAC addresses in a network • Unmanaged switches or network hubs	• Malicious packets containing fake or spoofed MAC addresses • Mac of, Ettercap3, Yersinia4, THC Parasite5 [129]	N/A
Protocol-based	SYN flood	Transport	• Exploits TCP three-way handshake connection system • Systems that do not use reduced SYN received timer or limited lifetime for half-open connections • Systems or OS that does not use proper backlog (associated with the ports) for TCP queues • Systems that do not recycle the oldest half-open connections	• Unrecognized connection requests from unknown foreign hosts • Obfuscated intrusion from a Malicious botnet (ex. Mirai botnet [64,34–39],) • Intrusion from an infected group of host devices (i.e. Distributed attack) • Edge devices with forged IP addresses.	[52,60,61,66,73, 74,78,79,111]
Protocol-based	Smurf DDoS	Transport	• Systems and devices without Anti-Malware solutions • Edge routers have misconfigured traffic monitoring policies, allowing unchecked IP-directed broadcast and • Lack of outbound IP filtration mechanisms	• Smurf malware, • Coremelt zombies [76–77] • Unchecked IP broadcasting hosts (equipped with spoofed IP)	[74,78,79]
Protocol Based	Ping of death	Internet	• Protocols that use IP datagram (i.e. ICMP, TCP, UDP etc.) • Critical layer 3 devices with ICMP response on • An edge server or a system is not equipped to drop the malformed data packet • Systems with very little memory buffer that are incapable of handling larger packets • Lack of filtration system	• Hosts sending malformed fragmented packets • Compromised systems with spoofed IP addresses	[74,78,82]
Protocol Based	LDoS	Application, transport, internet	• A TCP router with no mechanism to deal with traffic congestion [83] • Edge routers lacking RTO (Retransmission timeout) randomization and network flow monitoring mechanism • Operating systems using default or lower minimum RTO values [84]	• Fake sessions • Fragmented HTTP requests • Malicious TCP flow that causes link congestion	[87,88,89]
Application layer	HTTP GET/ POST	Application	• Mimics standard URL requests therefore very difficult to detect • Traditional rate-centric defense mechanisms are ineffective • Web portals and applications that do not use JavaScript-based bot detection mechanism (that contains CAPTCHAs, server and client-side behavioral analysis) • Apache, Nginx servers that do not use the HTTP timeout, connection limit, header size, or user backlog	• Botnets (ex: Mirai, Gafgyt, BashLite, etc.) • Malware like Trojan Horse • Unsupported and unauthorised HTTP requests • HTTP header with malicious payload injection • Host override headers, • Blacklisted domains • A server victim of web cache poisoning [93].	[66,73,51,78,94, 95,104,105]
Application layer	Slowloris	Application	• Allowing single IP to instantiate multiple connections • No restriction on incoming HTTP connections • No timeout settings on HTTP header from clients	• Incomplete HTTP connections • Malicious HTTP botnets • WireX (android botnet) • Web application-based Trojans	[106,107,108]
Application layer	Zero-day	Application	• An inadvertent software or hardware system flaw • Lack of rollback systems in Patch deployment • Networks having unmonitored web application traffic • Outdated web application software	• Malicious web browser extensions • Malware payloads stemming from phishing emails or web links	[78,[111]

(*continued on next page*)

Table 2 (*continued*)

Category	Classification	Layer	Vulnerabilities	Threats	Solutions
				• A compromised system with an undetected exploit	
Physical	Signal jamming	Physical	• Non-resilient RF devices • Usage of RF devices with similar band • No Anti-jamming mechanisms	• Anonymous RF devices • Microwaves, cordless phones, fluorescent lights, RF video cameras, etc.	[87,88,89,119]
Protocol-based	Teardrop attack	Internet	• Older systems with outdated operating systems • Unpatched systems with data assemble bugs • Devices lacking regular system updates • Networks lacking ingress filtering	• Anonymous packet sender with unknown IP origin • Fragmented or bug-laden packets	[122]
Protocol-based	Buffer overflow	Application	• Systems based on C/C++ or Fortran • System code that is reliant on external data • Errors or vulnerabilities in codes	• Malicious codes that exploit programs while triggering new actions • Programs that flood memory space [123]	[126]

and averaged to create a global model. Over time, this model trains to predict traffic patterns and inform the controller about any malicious activities occurring in the network or on the edge nodes. The controller can then take appropriate actions, such as blocking traffic from the identified malicious node or isolating its traffic until a proper countermeasure can be instantiated. The system operates in a secure manner, but it should be noted that the training period of the global model may be a limiting factor that requires further investigation and optimization. Additionally, the architecture currently relies on a single controller, which poses a single point of failure. To address this, the introduction of a backup or secondary controller in the event of a catastrophic failure would enhance system resilience and continuity.

8. Recommendations and future work

Based on the survey and review of existing solutions for edge systems, we provide some key recommendations that are crucial to safeguarding and averting potential disasters in edge networks. These recommendations can be classified into two categories: One is the layered approach, and the other is the holistic approach. The layered approach recommendations are primarily based on the five edge computing layers discussed in Section 4 and deal with the issues prevalent in each specific layer only, while the holistic approach mainly offers generalized recommendations for safeguarding edge systems as a whole.

8.1. Edge systems – layered approach recommendations

The layered approach recommendations highlight techniques focused on the five edge layers. They are listed follows:

a Application layer

The application layer is the layer built on a container-based infrastructure [137], where individual edge applications, middleware, and common software are executed. An edge server's (such as Multi-Access Edge Compute or MEC) data footprint dictates the complexity of those applications. Therefore, any sort of data overflow can cause massive disruption in this layer, when a flooding attack is initiated. That is why, securing this layer is imperative to keep the edge applications functional. To stop flooding attacks, two methods can be employed. One is per-packet-based detection, and the other is statistics-based detection [31]. Even though the primary cause of flooding-based DDoS attacks is mostly the protocol vulnerability itself, due to the limitations of traditional network architecture, the best solution is to trace attacks by implementing packet-filtering mechanisms. However, this kind of mechanism also comes with additional overhead, requiring the tracing of legitimate packet sources, individual IP tracing, maintaining ever-expansive IP/MAC tables, and so on. The statistics-based detection mainly employs entropy-based machine learning tools [138–139], which observe traffic flow and identify the pattern of traffic for the declaration of a DDoS flooding attack. However, the limitation is the sheer volume of traffic it needs to observe, which is often very taxing in terms of attack detection time and initial model training duration before any sort of detection [140]. Therefore, a hybrid form would be a great countermeasure, combining both per-packet-based and statistical-based approaches for the best result. Moreover, the application layer is heavily affected by malicious botnets. Therefore, countermeasures should be employed to curtail botnet activities in this layer, which was elaborated in the Application layer DDoS attacks (Section 5.1.3).

a Transport layer

In the transport layer, UDP flooding is a very common form of attack. As discussed earlier (in Section 5), there should be a filtering mechanism for UDP packets on critical ports, both for stateful and non-stateful UDP packets. The reduction of UDP packet response time can also mitigate DoS attacks on the edge system by a decent margin [141]. Moreover, for flood attacks, SYN cookies can be utilized for cryptographic hashing of the data and TCP reset cookies for precision handshakes.

a Internet layer

The network layer persistently deals with various IP protocols like IPv4, IPv6, 6LoWPAN, etc. Therefore, having an AAA (authentication, authorization, and accounting) mechanism in place can prevent a large portion of DoS attacks. There are existing works offering resilient AAA frameworks for edge systems. In [142] Moosavi et al. have proposed an authentication and authorization architecture that is based on the certificate-based datagram transport layer security (DTLS) handshake protocol and utilizes distributed smart gateways to safeguard distributed networks [143]. They claim that their framework architecture is more secure than the state-of-the-art delegation-based architecture and offers 26 % less overhead. On top of this, an efficient migration system can be implemented in case the AAA mechanism fails to make the overall system fail-proof [144].

a Datalink layer

The data link layer primarily deals with MAC addresses, which are associated with MAC tables. Therefore, the employment of strong MAC validation mechanisms can subvert various DoS attacks like address resolution protocol (ARP) spoofing, MAC flooding, etc. Fake ARP requests can cause the MAC cache to be filled with forged entries, essentially corrupting it and causing ARP poisoning, thereby triggering a denial of service to the real host [145]. Therefore, the implementation of anti-ARP spoofing mechanisms can help protect edge systems against DDoS attacks caused by ARP spoofing. A network that has network flow

Table 3
Research-based solutions, platforms, and approach/techniques.

Solution	Platform	Approach/Techniques	Ref.
DoS Detection Architecture for 6LoWPAN	6LoWPAN	Attack matrix-based IDS	[48]
Light-Weight DDoS mitigation scheme	Legacy	Queue shuffling	[49]
SoftEdgeNet	SDN	ACL based filtration	[50]
FlowGuard	ML	ML-based classifier with flow handler	[107]
Self-Organizing Map-based Approach, k-NN Algorithm	SDN/ML	SOM with KNN	[61]
Five-Layers SDP-Based MEC	SDN	SDP based MEC	[73]
Multi-Access Edge Computing-X	5G	Source-based method and reactive mitigation (Hybrid Solution)	[78]
Machine Learning DDoS Detection	ML	KDTree, LSVM, DT, RF and NN	[51]
Cooperative Defense Framework on MEC	Legacy	Code scheduling algorithm, control functions, sentry, and bot busters	[96]
Timing Side-Channel and Machine Learning	ML	Side channel information with SVM	[106]
Analytic Framework Intrusion Detection System	6LoWPAN	Packet length filtration (with Analytic framework)	[82]
Real-Time Volumetric Detection Scheme	SDN	Real-time volumetric detection (with Sliding time window, signal directional filter, and QuinDC algorithm)	[66]
Fog-Assisted SDN Controlled Framework	SDN/ML	E3ML multi-classifier with RNN, MLP, and ADT	[52]
SmartX multi-view visibility for OF@TEIN+dist. cloud-native edge boxes	SDN	Smart multi-view visibility framework (with active resource monitoring and distributed synchronization)	[60]
LSTM-based Network Attack Detection	ML	LSTM (with fine-tuned hyper-parameters)	[111]
Fog computing-based approach to DDoS mitigation in IIoT systems	Legacy/ VNF	Inline filtration (with NFV-based traffic analyzer)	[79]
LDoS Detection Using PSD-Based Entropy and Machine Learning	ML	PSD entropy with SVM	[87]
LDoS Attack Detection Based on Multi-feature Fusion and CNN	ML	Multi-feature fusion with CNN	[88]
FR-RED: Fractal Residual-Based Real-Time Detection of the LDoS Attack	ML	Fractal residual-based real-time detection	[89]
Federated Deep Learning for zero-day botnet Attack Detection	FL/ML	FL (FedAvg) with DNN	[111]
DDoS Attack Detection and Mitigation with SD-IoT and cosine similarity	SDN	Cosine similarity of the vectors of incoming packets	[53]
Mobile Edge Computing Shield for Heterogeneous IoT	SDN	SOM in SDN	[74]
ShadowNet	SDN/Web	Web-based filtration, susceptible to the flash crowd and geographical replication	[98]
Secure edge Computing in IoT via Online Learning	ML	Security-Aware edge serVer sElection algorithm to counter stochastic jamming	[119]
DoS attack Detection over IPv6 Network Based on KNN Algorithm	ML	GR-AD-KNN algorithm	[122]
Real-Time Lateral Movement Detection Based on Evidence Reasoning Network	ML	Evidence-based reasoning and Event tracking to thwart lateral movement attacks	[126]

control capabilities can help safeguard edge systems from ARP spoofing attacks by utilizing active ARP inspection and host certification models supervised by a centralized controller. Many of the systems are already existing, such as D-ARPSpoof [146], Network Flow Guard for ARP (NFGA) [147], Active ARP Inspection (AAI) [148], etc. Additionally, the usage of manageable switches and the standard use of static ARP tables can largely mitigate many of the DoS attacks on edge systems. Furthermore, data transmission in this layer can be secured by the use of the advanced encryption standard (AES) which ensures wireless security. Nevertheless, data link layer-based moving target defense (MTD) [103] can also be implemented that can randomize varying frame sizes, frame structures, randomized MAC addresses, and various encoding schemes to safeguard systems and devices by narrowing the attack window.

a Physical layer

For the physical layer, the standard recommendation is to make use of interference-resilient devices to avoid RF interruptions. In terms of imposed security, multiple authentication techniques can be employed to safeguard edge networks [149]. One such measure is channel-based authentication, which utilizes the unique properties of RF channels of legitimate nodes to detect trespassers. The method uses channel probing to estimate channel state information by comparing the old source node with the new one and declares the legitimacy of mutual authentication of user pairs [150]. Another method is the utilization of power spectral densities to ascertain intruders from legitimate users [151] which can work in tandem with the aforementioned channel-based authentication technique. Moreover, radio frequency fingerprinting can also be a great way to authenticate legitimate RF devices, which is based on the extraction of discriminating attributes from intrinsic physical properties of different hardware. These attributes can be extracted by identifying the variations of carrier frequency offset of RF devices, node variations based on spectral analysis, common phase error, in-phase/quadrature imbalance, etc. [150].

8.2. *Edge systems as a whole – holistic recommendations*

After the exploration of layer-based recommendations, we can now enumerate the holistic approach to edge system security. They are as followings:

a More research on edge systems:

The first step towards securing a system is to explore the vulnerabilities of the system and find its loopholes. Since edge computing is a relatively new paradigm and still evolving, exploration of its limitations and identification of possible vulnerabilities are imperative. Therefore, more research work is needed to ensure an in-depth analysis of individual elements within an edge system and also the subsidiaries like cloud and fog systems, which are expansive parts of edge computing. That way, it can not only help to prevent existing major threats like DDoS attacks, but it can even identify future attack patterns and unknown threats before they compromise the whole platform.

a Universal encryption standards between devices:

Edge systems boast a broad range of devices that establish communication links with each other. Therefore, the immediate next step should be the assurance of data security through secure communication protocols and advanced encryption methods to carry forward the data to end nodes [152]. As edge systems also have low-power devices, there can be an initiative to design a universal edge computing-based lightweight key encryption algorithm that can be effective for not only low-power devices but also for resource-heavy cloud servers.

Table 4
Research-based Solutions, types, platforms, testing metrics, and limitations.

Solution	Solution Type		Platform	Testing Metrics	Limitations	Ref.
	Detection	Prevention				
DoS Detection Architecture for 6LoWPAN	✔		6LoWPAN	Accuracy	Unable to handle dispersed sniffing from a larger distributed network	[48]
Light-Weight DDoS mitigation scheme	✔	✔	Legacy	Traffic flow rate	Only simulation-based, lack of empirical data	[49]
SoftEdgeNet	✔	✔	SDN	Traffic flow rate	Latency and node placement issues	[50]
FlowGuard	✔	✔	ML	Accuracy, precision, recall, F1 score	The efficiency of employed LSTM is not on par with standard IoT requirements	[60]
Self-Organizing Map-based Approach, k-NN Algorithm	✔		SDN/ML	Density, fault threshold	Single-point vulnerability lacks attack-handling capabilities in terms of intrinsic preventive measures	[61]
Five-Layers SDP-Based MEC		✔	SDN	CPU usage, packet delay	Single-point vulnerability generates overhead	[73]
Multi-Access Edge Computing-X	✔	✔	5G	N/A	Processing delay, lack of empirical data	[78]
Machine Learning DDoS Detection	✔		ML	Accuracy, precision, recall, F1 score	No mechanism to recover compromised nodes	[51]
Cooperative Defense Framework on MEC	✔	✔	Legacy	Accuracy, efficiency, traffic rate	Response delay from non-local CODE environment	[96]
Timing Side-Channel and Machine Learning	✔		ML	Accuracy, precision, recall, F1 score	lacks a prevention mechanism, with only 77.5 % accuracy	[106]
Analytic Framework Intrusion Detection System	✔		6LoWPAN	Accuracy, queue length	requires external packet modification, lack of empirical data	[82]
Real-Time Volumetric Detection Scheme	✔		SDN	Traffic flow rate	Only works against volumetric attacks	[66]
Fog-Assisted SDN Controlled Framework	✔	✔	SDN/ML	Accuracy, precision, recall, F1 score	MLP and RNN performances are unstable, and lack of empirical data	[52]
SmartX multi-view visibility for OF@TEIN+dist. cloud-native edge boxes	✔		SDN	Traffic flow rate	Lack of empirical data in the real-time data network	[107]
LSTM-based Network Attack Detection	✔		ML	AUC, detection rate, detection accuracy, FPR, FNR, F1 score	Does not provide any prevention mechanism, suffers in accuracy against botnet attacks	[108]
Fog computing-based approach to DDoS mitigation in IIoT systems	✔	✔	Legacy/ VNF	Detection rate, end-to-end delay	only 70 % accuracy in fog level in terms of Modbus, high latency in detection	[79]
LDoS Detection Using PSD-Based Entropy and Machine Learning	✔		ML	Detection rate, detection accuracy	Does not provide any prevention mechanism	[87]
LDoS Attack Detection Based on Multi-feature Fusion and CNN	✔		ML	Accuracy, FPR, FNR	High false negative rate, done on a small scale, lack of empirical data	[88]
FR-RED: Fractal Residual-Based Real-Time Detection of the LDoS Attack	✔		ML	Accuracy, F1 score, FPR, FNR	Done on a small scale, lack of empirical data	[89]
Federated Deep Learning for zero-day botnet Attack Detection	✔		FL/ML	Accuracy, precision, recall, F1 score	Done with only 5 devices (very small scale), global model training takes a long time	[111]
DDoS Attack Detection and Mitigation with SD-IoT and cosine similarity	✔	✔	SDN	Traffic flow rate	Only performed on UDP transmissions using Scapy, single point vulnerability, Done on a small scale	[53]
Mobile Edge Computing Shield for heterogeneous IoT	✔	✔	SDN	Accuracy, precision	Filter training delay, very High CPU usage	[74]
ShadowNet	✔	✔	SDN/Web	Detection time, packet per second (PPS)	Geographical replications affect fast path and aggregation performance, slow detection process with overheads	[98]
Secure edge Computing in IoT via Online Learning	✔		ML	Regret analysis	More experiments are needed considering various types of signal jamming attacks and interferences	[119]
DoS attack Detection over IPv6 Network Based on KNN Algorithm	✔		ML	F1 Score	Evaluation is only based on teardrop attacks which are comparatively less severe, with no prevention mechanism	[122]
Real-Time Lateral Movement Detection Based on Evidence Reasoning Network	✔		ML	Confidence interval, events per day	Does not provide any prevention mechanism, lacks empirical data	[126]

a Firewalls based on network type:

A network without a point firewall is like a treasure chest without any locks for an invader. Therefore, the use of a firewall is mandatory for any network, be it an enterprise-level or small home network. Various types of firewalls can be used to protect a network entry point, depending on network types and requirements. It can be an ingress packet filtering firewall for incoming packet inspection, next-gen firewalls (NGFW) for blocking malware attacks and external threats, stateful inspection firewalls ensuring three-way handshake ensuring end-to-end secure connectivity through tracking IP source and destination for each connection, a well-configured SDN-controlled virtual firewall [153–154] that ensures firewall functionalities with zero overhead cost, etc. These firewalls can be the primary differentiator between a secured enterprise network and a catastrophic distributed denial of service (DDoS) disaster.

a Network Supervision and fine-grained access control

The inherent lack of supervision and access control in unsupervised networks poses significant security vulnerabilities. In the context of edge computing, where numerous low-powered IoT devices are interconnected, the primary focus is often on performance rather than fine-grained access control [155]. Furthermore, the deployment of edge and IoT devices in a fragmented manner throughout the network exacerbates the security challenges. The passive nature of defense mechanisms in such a disjointed environment further exposes the system to potential threats [31]. To address these vulnerabilities and enhance network supervision and access control, a centralized mechanism is needed [156]. This can be achieved through the use of a

Table 5
Solution techniques based on DoS types.

Category	Classification	Layer	Approach/ Techniques	Research-based Solutions
Volumetric	UDP flood	Transport	Attack matrix-based IDS, Queue shuffling, ACL-based filtration, ML-based classifier with flow handler, SOM with KNN, Hybrid solution, KDTree, LSVM, DT, RF and NN, Real-time volumetric detection, E3ML multi-classifier, FL with DNN, Cosine similarity of the vectors of incoming packets, Web-based filtration	[48,49,50, 60,61,78, 51,66,52, 111,53,98]
Volumetric	ICMP flood	Internet	ML-based classifier with flow handler, SOM with KNN, Hybrid Solution, SOM in SDN	[60,61,78, 74]
Volumetric	DNS amplification	Application, transport, internet	ML-based classifier with flow handler, SOM with KNN, SOM in SDN	[60,66,74]
Volumetric	MAC flood	Data link	N/A	N/A
Protocol-based	SYN flood	Transport	ML-based classifier with flow handler, SOM with KNN, SDP-based MEC, Hybrid Solution, Real-time volumetric detection, E3ML multi-classifier, Inline filtration, FL with DNN, SOM in SDN	[60,61,73, 78,66,52, 79,111,74]
Protocol-based	Smurf DDoS	Transport	Hybrid Solution, Inline filtration, SOM in SDN	[78,79,74]
Protocol Based	Ping of death	Internet	Hybrid Solution, Packet length filtration, SOM in SDN	[78,82,74]
Protocol Based	LDoS	Application, transport, internet	PSD entropy with SVM, Multi-feature fusion with CNN, Fractal residual-based real-time detection	[87,88,89]
Application layer	HTTP GET/ POST	Application	SDP-based MEC, Hybrid Solution, KDTree, LSVM, DT, RF and NN, Code scheduling algorithm, control functions, sentry and bot busters, Real-time volumetric detection, Smart multi-view visibility framework, LSTM, Web-based filtration	[73,78,51, 96,66,107, 108,98]
Application layer	Slowloris	Application	Side channel information with SVM, MVF, LSTM	[106,108, 108]
Application layer	Zero-day	Application	Hybrid Solution, FL with DNN	[78,108]
Physical	Signal jamming	Physical	PSD entropy with SVM, Multi-feature fusion with CNN, Fractal residual-based real-time detection, Security-Aware edge serVer sElection algorithm	[87],88,89, 119]
Protocol-based	Teardrop attack	Internet	GR-AD-KNN algorithm	[122]
Protocol-based	Buffer overflow	Application	Evidence-based reasoning and Event tracking to thwart lateral movement attacks	[126]

software-defined network (SDN) management tool. SDN offers several advantages, including the separation of the control and data planes, which allows for flexible network orchestration. It also enables centralized supervision and access control of individual network elements, effectively mitigating the limitations of the heterogeneous distributed environment [157]. Therefore, in our proposed architecture, we have chosen to leverage the SDN platform to overcome the coarse-grained nature of edge devices and gain superior control over the network elements. The utilization of SDN brings additional benefits, such as the ability to prioritize network requests, perform IP traceback, and implement active flow filtering. These capabilities can be instrumental in preventing and mitigating malicious traffic flows within the network [158]. By employing SDN's holistic control over network flow, we can enhance the security of the edge computing environment and ensure the integrity and availability of critical services.

a Secure communication of IoT devices:

In the context of IoT-based systems, securing communication between IoT devices is of paramount importance [159]. Edge computing plays a crucial role in enabling powerful capabilities for IoT devices. However, the diverse nature of IoT communication schemes, including unicast, multicast, and broadcast, poses a challenge in finding a universal solution [160]. To address this, it is essential to assess the vulnerabilities and apply appropriate countermeasures [161,162]. One effective method is IP traceback, which helps identify the origin of incoming packets. Traditional IP traceback techniques such as link testing, ICMP messaging, logging, packet marking, hop-to-hop tracing, and hop count filtering can be employed to expose the source of malicious traffic and prevent DoS attacks [102]. As mentioned briefly in Section 8.1***d***, Moving Target Defense (MTD) can also serve as an effective approach against DoS attacks [163]. MTD involves dynamically reconfiguring network statistics with random values to confuse attackers. For instance, dynamic randomization of the IP or MAC address

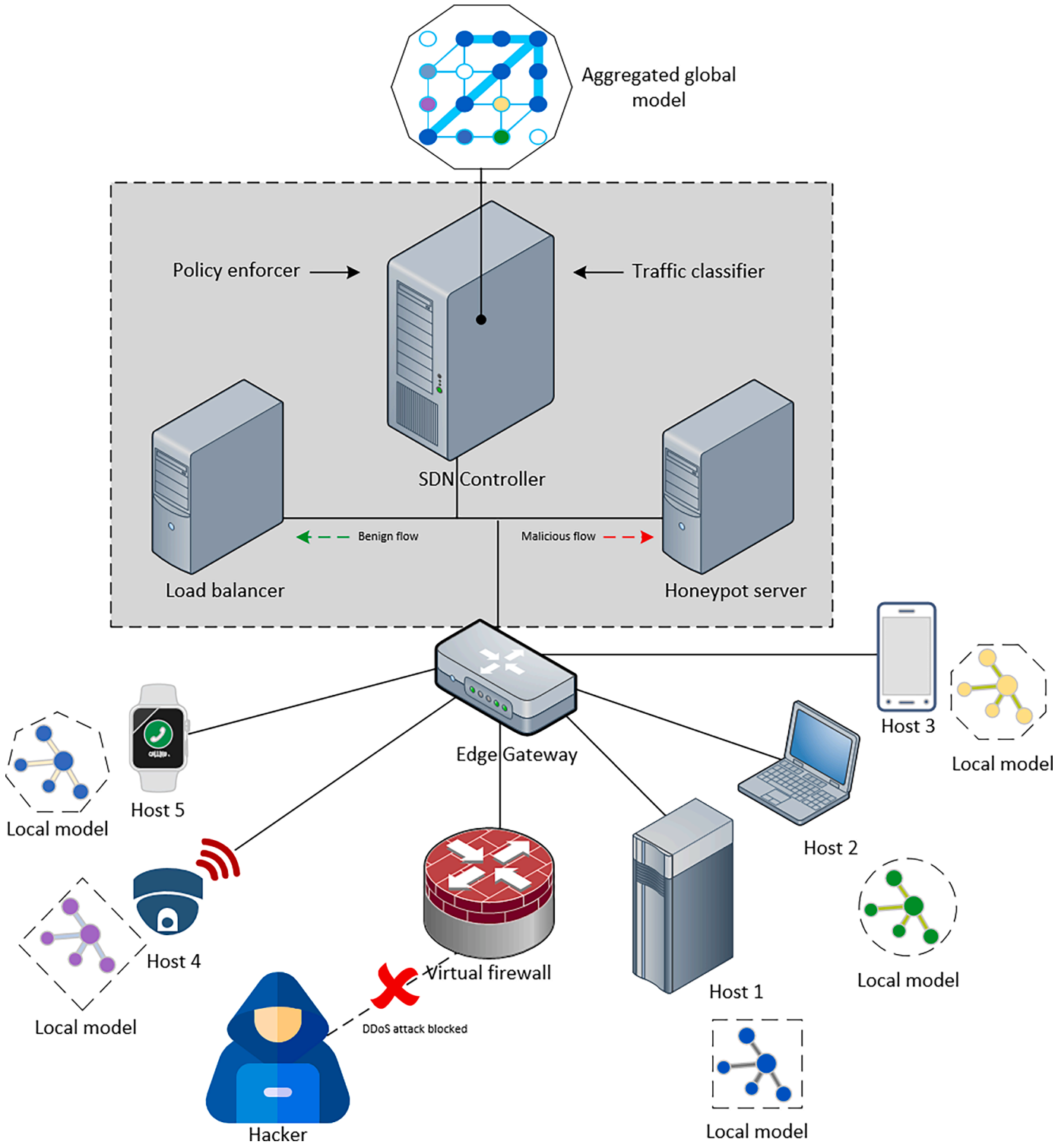


Fig. 6. Proposed DDoS security architecture.

of a host can make it extremely challenging for hackers to launch successful attacks [164]. Diversification and redundancy-based MTD techniques can also be applied, utilizing proxy-based or active mitigation methods to combat DoS attacks. However, it is important to consider the resource limitations of IoT devices, including computational power, network overhead, throughput, and response time, before implementing these methods. A balanced approach is crucial to ensure that the benefits of the security measures outweigh their limitations [165,166]. In this regard, SDN-based infrastructures are particularly advantageous, as they do not suffer from the constraints of traditional networks [167,168]. Thus, in our proposed theoretical architecture, we have chosen SDN as the preferred approach.

a Utilization of AI with federated learning:

In this research paper, we have explored various machine learning-based models that have been utilized by researchers for preventing denial of service (DoS) and distributed denial of service (DDoS) attacks.

We have highlighted the potential of federated learning (FL) techniques to enhance the security of edge systems by facilitating secure data sharing among major hosts, while still leveraging the protection provided by traditional machine learning techniques for regular data streams [169–172]. This approach allows for a flexible network security framework where different security layers can be maintained based on the confidentiality levels of the data, employing partial or fully federated learning-based models [173].

For our future work, we plan to extend our research by implementing the FL-based framework and conducting simulations in a real-world research test bed [174]. We will perform extensive evaluations of our prototype to assess its effectiveness in preventing DDoS attacks in edge systems [175]. Furthermore, we aim to incorporate multiple machine learning algorithms such as FedSGD [176], FedProx [177] and FedDANE [178] etc. and compare their performance using standard evaluation metrics across various intrusion patterns. Additionally, we intend to explore the integration of other platforms such as fog computing, cloud computing, and hybrid approaches to evaluate the scalability and effectiveness of our framework in multi-platform scenarios. These future endeavors will provide valuable insights into the capabilities and limitations of our proposed framework and contribute to the advancement of secure network architectures in edge computing environments.

9. Conclusion

The emergence of edge computing has brought about significant advancements in integrating Internet of Things (IoT) and cloud systems. However, to fully leverage the potential of this technology, it is crucial to ensure robust security measures and minimize vulnerabilities. Unfortunately, the research focus on security, particularly in the context of denial of service (DoS) attack types and their countermeasures in edge computing, remains insufficient, highlighting the need for further investigation. In this paper, we have taken a preliminary step to address this research gap by providing a comprehensive overview of DoS and distributed denial of service (DDoS) attack types specifically targeting edge systems across different layers. We have examined the existing state-of-the-art solutions and discussed their limitations. Furthermore, we have identified common countermeasures, threats, and vulnerabilities prevalent in edge systems. To tackle DDoS attacks in edge systems, we have proposed a novel architecture based on the utilization of federated learning within a Software-Defined Networking (SDN) platform. Our architecture combines the power of machine learning with the flexibility of SDN to enhance the security of edge systems. However, the effectiveness of this proposed system can only be evaluated through practical implementation on a data-centric network infrastructure, which represents the next phase of our research. By shedding light on the security challenges and proposing a potential solution, this work aims to stimulate further exploration and development in securing edge computing environments. Future research endeavors will focus on implementing and evaluating the proposed architecture in real-world scenarios, which will contribute to the ongoing efforts in strengthening the security of edge systems.

Declaration of Competing Interest

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

Data availability

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

Acknowledgement

This study was supported by funding through the US National Science Foundation Award number 2028397.

References

[1] Y.C. Hu, M. Patel, D. Sabella, N. Sprecher, V. Young, Mobile edge computing - a key technology towards 5G, ETSI (11) (2015). Available, https://www.etsi.org/images/files/etsiwhitepapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf [Accessed 20 March 2022].
[2] "Edge Computing in Industrial Automation", Mh-monitoringcontrol.com, 2022. [Online]. Available: https://mh-monitoringcontrol.com/sites/default/files/downloads/mh_edge_computing_in_industrial_automation.pdf. [Accessed: 20- May- 2022].
[3] "How 5G and edge computing will transform AR & VR use cases", STL Partners, 2021. [Online]. Available: https://stlpartners.com/articles/edge-computing/how-5g-and-edge-computing-will-transform-ar-vr-use-cases/. [Accessed: 20- May- 2022].
[4] D. Aishwarya, R.I. Minu, Edge computing based surveillance framework for real-time activity recognition, ICT Express 7 (2) (2021) 182–186, https://doi.org/10.1016/j.icte.2021.04.010. Available [Accessed 20 May 2022].
[5] Y. Mao, S. Yi, Q. Li, J. Feng, F. Xu, S. Zhong, A privacy-preserving deep learning approach for face recognition with edge computing", HotEdge (2018). '18Available, https://www.usenix.org/system/files/conference/hotedge18/hotedge18-papers-mao.pdf [Accessed 20 May 2022].
[6] A. Murray, How edge computing makes voice assistants faster and more powerful, Netw. World (2022) [Online]. Available, https://www.networkworld.com/article/3262105/how-edge-computing-makes-voice-assistants-faster-and-more-powerful.html [Accessed: 20- May- 2022].
[7] S.A. Kumar, T. Vealey, H. Srivastava, Security in internet of things: challenges, solutions and future directions, in: 2016 49th Hawaii International Conference on System Sciences (HICSS), IEEE, 2016, pp. 5772–5781.
[8] J. Chelladhurai, P.R. Chelliah, S.A. Kumar, Securing docker containers from denial of service (dos) attacks, in: 2016 IEEE International Conference on Services Computing (SCC), IEEE, 2016, pp. 856–859.
[9] S. Kumar, B. Xu, Vulnerability assessment for security in aviation cyber-physical systems, in: 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), IEEE, 2017, pp. 145–150.
[10] S. Alampalayam, A. Kumar, An adaptive and predictive security model for mobile ad hoc networks, Wirel. Pers. Commun. 29 (2004) 263–281.
[11] S. Sullivan, A. Brighente, S. Kumar, M. Conti, 5G security challenges and solutions: a review by OSI layers, IEEE Access 9 (2021) 116294–116314.
[12] S. Alampalayam, A. Kumar, Predictive security model using data mining, in: IEEE Global Telecommunications Conference, 2004. GLOBECOM'04 4, IEEE, 2004, pp. 2208–2212.
[13] M. Gohil, S. Kumar, Evaluation of classification algorithms for distributed denial of service attack detection, in: 2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), IEEE, 2020, pp. 138–141.
[14] S.A. Kumar, Classification and review of security schemes in mobile computing, Wirel. Sens. Netw. 2 (06) (2010) 419.
[15] S. Alampalayam, A. Kumar, Security model for routing attacks in mobile ad hoc networks, in: 2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall 3, IEEE, 2003, pp. 2122–2126. IEEE Cat. No. 03CH37484.
[16] J. Harvey, S. Kumar, A survey of intelligent transportation systems security: challenges and solutions, in: 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), IEEE, 2020, pp. 263–268.
[17] S. Alampalayam, E.F. Natsheh, Multivariate fuzzy analysis for mobile ad hoc network threat detection, Int. J. Bus. Data Commun. Netw. (IJBDCN) 4 (3) (2008) 1–30.
[18] D. Eastman, S. Kumar, A simulation study to detect attacks on internet of things, in: 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress, IEEE, 2017, pp. 645–650. DASC/PiCom/DataCom/CyberSciTech.
[19] S. Kumar, S. Velliangiri, P. Karthikeyan, S. Kumari, S. Kumar, M.K. Khan, A survey on the blockchain techniques for the Internet of Vehicles security, Trans. Emerg. Telecommun. Technol. (2021) e4317.
[20] A. Bikos, S. Kumar, Securing digital ledger technologies-enabled IoT devices: taxonomy, challenges, and solutions, IEEE Access 10 (2022) 46238–46254.
[21] C. Liptak, S. Mal-Sarkar, S. Kumar, Power analysis side channel attacks and countermeasures for the internet of things, in: 2022 IEEE Physical Assurance and Inspection of Electronics (PAINE), IEEE, 2022, pp. 1–7.
[22] A. Bikos, S. Kumar, Reinforcement learning-based anomaly detection for Internet of Things distributed ledger technology, in: 2021 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2021, pp. 1–7.
[23] S. Kumar, B. Bhargava, R. Macêdo, G. Mani, Securing iot-based cyber-physical human systems against collaborative attacks, in: 2017 IEEE International Congress on Internet of Things (ICIOT), IEEE, 2017, pp. 9–16.

[24] S. Alampalayam, A. Kumar, S. Srinivasan, Mobile ad hoc network security-a taxonomy, in: The 7th International Conference on Advanced Communication Technology 2, IEEE, 2005, pp. 839–844. ICACT 2005.2005.

[25] S. Haim, DDoS Protection in the Age of 5G Networks, Edge Computing and Explosive Bandwidth Growth, Security Boulevard, 2022 [Online]. Available, https://securityboulevard.com/2022/03/ddos-protection-in-the-age-of-5g-networks-edge-computing-and-explosive-bandwidth-growth/ [Accessed: 01- Jul- 2022].

[26] S. Gatlan, Microsoft Mitigates Largest DDoS Attack 'ever reported in History, BleepingComputer, 2022 [Online]. Available, https://www.bleepingcomputer.com/news/security/microsoft-mitigates-largest-ddos-attack-ever-reported-in-history/ [Accessed: 01- Jul- 2022].

[27] H. Zeyu, X. Geming, W. Zhaohang, Y. Sen, Survey on edge computing security, in: 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), 2020, https://doi.org/10.1109/icbaie499/2020.00027. Available[Accessed 24 June 2022].

[28] I. Ahmad, A Survey on DDoS Attacks in Edge Servers, University of Texas at Arlington, 2022.

[29] A. Ometov, O. Molua, M. Komarov, J. Nurmi, A survey of security in cloud, edge, and fog computing, Sensors 22 (3) (2022) 927, https://doi.org/10.3390/s22030927. Available[Accessed 24 June 2022].

[30] S. Zargar, J. Joshi, D. Tipper, A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks, IEEE Commun. Surv. Tutor. 15 (4) (2013) 2046–2069.

[31] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, W. Lv, Edge computing security: state of the art and challenges, Proc. IEEE 107 (8) (2019) 1608–1631, https://doi.org/10.1109/jproc.2019.2918437. Available[Accessed 25 February 2022].

[32] S. Singh, R. Sulthana, T. Shewale, V. Chamola, A. Benslimane, B. Sikdar, Machine-learning-assisted security and privacy provisioning for Edge computing: a survey, IEEE Internet of Things J. 9 (1) (2022) 236–260.

[33] "What is A DDoS Attack?", www.akamai.com, 2022. [Online]. Available: https://www.akamai.com/our-thinking/ddos. [Accessed: 10- Feb- 2022].

[34] J. Fruhlinger, The Mirai botnet explained: how IoT devices almost brought down the internet, CSO Online (2018) [Online]. Available, https://www.csoonline.com/article/3258748/the-mirai-botnet-explained-how-teen-scammers-and-cctv-cameras-almost-brought-down-the-internet.html [Accessed: 30- May- 2022].

[35] M. Salim, S. Rathore, J. Park, Distributed denial of service attacks and its defenses in IoT: a survey, J. Supercomput. 76 (7) (2019) 5320–5363, https://doi.org/10.1007/s11227-019-02945-z. Available[Accessed 29 May 2022].

[36] C. Williams, Today the web was broken by countless hacked devices – your 60-second summary", Theregister.com (2016) [Online]. Available, https://www.theregister.com/2016/10/21/dyn_dns_ddos_explained/ [Accessed: 31- May- 2022].

[37] B. Krebs, "Did the Mirai Botnet really take Liberia offline? – Krebs on security", Krebsonsecurity.com, 2016. [Online]. Available: https://krebsonsecurity.com/2016/11/did-the-mirai-botnet-really-take-liberia-offline/. [Accessed: 31- May- 2022].

[38] L. Pascu, British hacker-for-hire goes to prison for liberian telecom, Deutsche Telekom Mirai attack, Hot Secur. (2019) [Online]. Available, https://www.bitdefender.com/blog/hotforsecurity/british-hacker-for-hire-goes-to-prison-for-liberian-telecom-deutsche-telekom-mirai-attack [Accessed: 30- May- 2022].

[39] T. Seals, "Mirai Botnet sees big 2019 growth, shifts focus to enterprises", Threatpost.com, 2019. [Online]. Available: https://threatpost.com/mirai-botnet-sees-big-2019-growth-shifts-focus-to-enterprises/146547/. [Accessed: 30- May- 2022].

[40] D. Kortepeter, TechGenix, 2021.

[41] S. Stelfox, What is Gafgyt malware? Smart Home Cybersecur. News (2019) [October 2019 edition]", Minim.com[Online]. Available, https://www.minim.com/blog/smart-home-cybersecurity-news-roundup-what-is-gafgyt-malware-october-2019-edition [Accessed: 01- Jun- 2022].

[42] "How IoT Devices Are Impacting Edge Architecture | NETSCOUT", NETSCOUT, 2022. [Online]. Available: https://www.netscout.com/blog/iot-impacting-edge-architecture. [Accessed: 02- Jun- 2022].

[43] R. Kawasoe, C. Han, R. Isawa, T. Takahashi, J. Takeuchi, Investigating behavioral differences between IoT malware via function call sequence graphs, in: Proceedings of the 36th Annual ACM Symposium on Applied Computing, 2021, pp. 1674–1682. Available, https://dl.acm.org/doi/10.1145/3412841.3442041 [Accessed 2 June 2022].

[44] "BASHLITE", DBpedia, 2022. [Online]. Available: https://dbpedia.org/page/BASHLITE. [Accessed: 01- Jun- 2022].

[45] A. Marzano, et al., The evolution of Bashlite and Mirai IoT botnets, in: 2018 IEEE Symposium on Computers and Communications (ISCC), 2018, https://doi.org/10.1109/iscc.2018.8538636. Available[Accessed 2 June 2022].

[46] "UDP-Based Amplification Attacks | CISA", Cisa.gov, 2014. [Online]. Available: https://www.cisa.gov/uscert/ncas/alerts/TA14-017A. [Accessed: 14- Feb- 2022].

[47] "What is a UDP Flood | DDoS Attack Glossary | Imperva", Learning center, 2022. [Online]. Available: https://www.imperva.com/learn/ddos/udp-flood/. [Accessed: 13- Feb- 2022].

[48] P. Kasinathan, C. Pastrone, M. Spirito, M. Vinkovits, Denial-of-Service detection in 6LoWPAN based Internet of Things, in: 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2013, https://doi.org/10.1109/wimob.2013.6673419. Available[Accessed 09 February 2022].

[49] R. Yaegashi, D. Hisano, Y. Nakayama, Light-weight DDoS mitigation at network edge with limited resources, in: 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), 2021, https://doi.org/10.1109/ccnc49032.2021.9369635. Available[Accessed 10 February 2022].

[50] P. Sharma, S. Rathore, Y. Jeong, J. Park, SoftEdgeNet: SDN based energy-efficient distributed network architecture for edge computing, IEEE Commun. Mag. 56 (12) (2018) 104–111, https://doi.org/10.1109/mcom.2018.1700822. Available [Accessed 10 February 2022].

[51] R. Doshi, N. Apthorpe, N. Feamster, Machine learning DDoS detection for consumer internet of things devices, in: 2018 IEEE Security and Privacy Workshops (SPW), 2018, https://doi.org/10.1109/spw.2018.00013. Available [Accessed 13 February 2022].

[52] Q. Shafi, A. Basit, S. Qaisar, A. Koay, I. Welch, Fog-assisted SDN controlled framework for enduring anomaly detection in an IoT network, IEEE Access 6 (2018) 73713–73723, https://doi.org/10.1109/access.2018.2884293. Available [Accessed 27 February 2022].

[53] D. Yin, L. Zhang, K. Yang, A DDoS attack detection and mitigation with software-defined internet of things framework, IEEE Access 6 (2018) 24694–24705, https://doi.org/10.1109/access.2018.2831284. Available[Accessed 27 May 2022].

[54] N. Dao, J. Park, M. Park, S. Cho, A feasible method to combat against DDoS attack in SDN network, in: 2015 International Conference on Information Networking (ICOIN), 2015, https://doi.org/10.1109/icoin.2015.7057902. Available [Accessed 28 May 2022].

[55] N. Gupta, A. Jain, P. Saini, V. Gupta, DDoS attack algorithm using ICMP flood, in: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2016.

[56] L. Eden, The truth about ICMP, in: Global information assurance certification paper, SANS Institute, 2002. Available, https://www.giac.org/paper/gsec/719/truth-about-icmp/101601 [Accessed 12 February 2022].

[57] N. Tuan, P. Hung, N. Nghia, N. Tho, T. Phan, N. Thanh, A DDoS attack mitigation scheme in ISP networks using machine learning based on SDN, Electronics (Basel) 9 (3) (2020) 413, https://doi.org/10.3390/electronics9030413. Available [Accessed 12 February 2022].

[58] "What is an ICMP flood attack?," NETSCOUT. [Online]. Available: https://www.netscout.com/what-is-ddos/icmp-flood. [Accessed: 14-Dec-2022].

[59] Y. Jia, F. Zhong, A. Alrawais, B. Gong, X. Cheng, FlowGuard: an intelligent edge defense mechanism against IoT DDoS attacks, IEEE Internet of Things J. 7 (10) (2020) 9552–9562, https://doi.org/10.1109/jiot.2020.2993782. Available [Accessed 12 February 2022].

[60] J. Firch, How To Prevent A ICMP Flood Attack, PurpleSec, 2022 [Online]. Available, https://purplesec.us/prevent-ping-attacks/ [Accessed: 17- Feb- 2022].

[61] M. Hossain, H. Ochiai, D. Fall, Y. Kadobayashi, LSTM-based network attack detection: performance comparison by hyper-parameter values tuning, in: 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud, 2020, https://doi.org/10.1109/cscloud-edgecom49738.2020.00020. EdgeComAvailable[Accessed 27 February 2022].

[62] D. Walkowski, What is a DNS amplification attack? F5 Labs (2019) [Online]. Available, https://www.f5.com/labs/articles/education/what-is-a-dns-amplification-attack- [Accessed: 12- Feb- 2022].

[63] "DNS Amplification Attacks | CISA", Cisa.gov, 2003. [Online]. Available: https://www.cisa.gov/uscert/ncas/alerts/TA13-088A. [Accessed: 12- Feb- 2022].

[64] "DNS Amplification Attacks | CISA", Cisa.gov, 2014. [Online]. Available: https://www.cisa.gov/uscert/ncas/alerts/TA13-088A. [Accessed: 14- Feb- 2022].

[65] J. Firch, How To Prevent A Domain Name Server (DNS) Amplification attack, Purplesec, 2021 [Online]. Available, https://purplesec.us/prevent-dns-amplification-attack/ [Accessed: 12- Feb- 2022].

[66] J. Li, M. Liu, Z. Xue, X. Fan, X. He, RTVD: a real-time volumetric detection scheme for DDoS in the Internet of Things, IEEE Access 8 (2020) 36191–36201, https://doi.org/10.1109/access.2020.2974293. Available[Accessed 25 February 2022].

[67] R. Das, A. Karabade, G. Tuna, Common network attack types and defense mechanisms, in: 2015 23nd Signal Processing and Communications Applications Conference (SIU), 2015, https://doi.org/10.1109/siu.2015.7130435. Available [Accessed 25 June 2022].

[68] "What is 802.1X? How Does it Work?", SecureW2, 2022. [Online]. Available: https://www.securew2.com/solutions/802-1x#:~:text=802.1X%20is%20a%20network,confirmed%20by%20the%20RADIUS%20server. [Accessed: 25- Jun- 2022].

[69] V. Chinnasamy, 2021. What is SYN (synchronize) attack? How the attack works and how to prevent the SYN attack. [online] www.indusface.com. Available at: https://www.indusface.com/blog/what-is-syn-synchronize-attack-how-the-attack-works-and-how-to-prevent-the-syn-attack/. [Accessed 12 February 2022].

[70] "What is the Mirai Botnet?", www.cloudflare.com, 2021. [Online]. Available: https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/. [Accessed: 07- Apr- 2022].

[71] C. Easttom, Network Defense and Countermeasures, 2nd ed., Pearson, 2006, p. 44.

[72] P. Goldschmidt, "TCP Reset Cookies – a heuristic method for TCP SYN Flood mitigation", 2019. Available: http://excel.fit.vutbr.cz/submissions/2019/057/57.pdf. [Accessed 18 February 2022].

[73] J. Singh, Y. Bello, A. Hussein, A. Erbad, A. Mohamed, Five-layers SDP-based hierarchical security paradigm for iot multiaccess edge computing, IEEE Internet of Things J. 8 (7) (2021) 5794–5805, https://doi.org/10.1109/jiot.2020.3033265. Available[Accessed 12 February 2022].

[74] N. Dao, et al., Securing heterogeneous IoT with intelligent DDoS attack behavior learning, IEEE Syst. J. (2019). Available, https://arxiv.org/abs/1711.06041 [Accessed 27 May 2022].

[75] P. Festa, Smurf" Attack Hits Minnesota, CNET, 1998 [Online]. Available, https://www.cnet.com/news/smurf-attack-hits-minnesota/ [Accessed: 13- Feb- 2022].

[76] E. Georgescu, What Is a Smurf Attack and How to Prevent It, Heimdal Security Blog, 2021 [Online]. Available, https://heimdalsecurity.com/blog/smurf-attack-ddos/ [Accessed: 11- Mar- 2022].

[77] A. Studer, A. Perrig, The Coremelt attack, Comput. Secur. – ESORICS (2009) 37–52, 2009.

[78] N. Dao, D. Vu, Y. Lee, M. Park, S. Cho, MAEC-X: DDoS prevention leveraging multi-access edge computing, in: 2018 International Conference on Information Networking (ICOIN), 2018, https://doi.org/10.1109/icoin.2018.8343118. Available[Accessed 13 February 2022].

[79] L. Zhou, H. Guo, G. Deng, A fog computing based approach to DDoS mitigation in IIoT systems, Comput. Secur. 85 (2019) 51–62, https://doi.org/10.1016/j.cose.2019.04.017. Available[Accessed 28 February 2022].

[80] "What is the Ping of Death (pod)? definition, damage & defense," Okta. [Online]. Available: https://www.okta.com/identity-101/ping-of-death/. [Accessed: 14- Dec-2022].

[81] "Ping of death DDoS attack", www.cloudflare.com, 2021. [Online]. Available: https://www.cloudflare.com/learning/ddos/ping-of-death-ddos-attack/. [Accessed: 16- Feb- 2022].

[82] A. Abdollahi, M. Fathi, An intrusion detection system on ping of death attacks in IoT networks, Wirel. Pers. Commun. 112 (4) (2020) 2057–2070. Available, https://link.springer.com/article/10.1007/s11277-020-07139-y [Accessed 18 February 2022].

[83] W. Zhijun, L. Wenjing, L. Liang, Y. Meng, Low-rate DoS attacks, detection, defense, and challenges: a survey, IEEE Access 8 (2020) 43920–43943, https://doi.org/10.1109/access.2020.2976609. Available[Accessed 28 February 2022].

[84] R. Mathew, V. Katkar, Survey of low rate DoS attack detection mechanisms, in: Proceedings of the International Conference & Workshop on Emerging Trends in Technology - ICWET '11, 2011, https://doi.org/10.1145/1980022.1980227. Available[Accessed 28 February 2022].

[85] S. Sarat, A. Terzis, On the effect of router buffer sizes on low-rate denial of service attacks, in: Proceedings. 14th International Conference on Computer Communications and Networks, ICCCN, 2005, https://doi.org/10.1109/icccn.2005.1523867, 2005., 2005. Available[Accessed 5 March 2022].

[86] Guang Yang, M. Gerla, M. Sanadidi, Defense against low-rate TCP-targeted denial-of-service attacks, in: Proceedings. ISCC 2004. Ninth International Symposium on Computers And Communications, 2022, https://doi.org/10.1109/iscc.2004.1358428. IEEE Cat. No.04TH8769Available[Accessed 5 March].

[87] N. Zhang, F. Jaafar, Y. Malik, Low-rate DoS attack detection using PSD based entropy and machine learning, in: 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud, 2019, https://doi.org/10.1109/cscloud/edgecom.2019.00020. EdgeCom)Available[Accessed 28 February 2022].

[88] D. Tang, L. Tang, W. Shi, S. Zhan, Q. Yang, MF-CNN: a new approach for LDoS attack detection based on multi-feature fusion and CNN, Mob. Netw. Appl. 26 (4) (2020) 1705–1722, https://doi.org/10.1007/s11036-019-01506-1. Available [Accessed 19 May 2022].

[89] D. Tang, Y. Feng, S. Zhang, Z. Qin, FR-RED: fractal residual based real-time detection of the LDoS attack, IEEE Trans. Reliab. 70 (3) (2021) 1143–1157, https://doi.org/10.1109/tr.2020.3023257. Available[Accessed 19 May 2022].

[90] I. Barsukov, A. Bobreshov, M. Riapolov, Fractal analysis based detection of DoS/LDoS network attacks, in: 2019 International Russian Automation Conference (RusAutoCon), 2019, https://doi.org/10.1109/rusautocon.2019.8867618. Available[Accessed 19 May 2022].

[91] "What is a DDoS Botnet | Common Botnets and Botnet Tools | Imperva", Learning center, 2021. [Online]. Available: https://www.imperva.com/learn/ddos/botnet-ddos/. [Accessed: 29- May- 2022].

[92] "HTTP flood attack", cloudflare.com, 2022. [Online]. Available: https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/. [Accessed: 13- Feb- 2022].

[93] J. Kettle, Practical web cache poisoning, PortSwigger Res. (2018) [Online]. Available, https://portswigger.net/research/practical-web-cache-poisoning [Accessed: 13- Feb- 2022].

[94] X. Liu, G. Zhou, M. Kong, Z. Yin, X. Li, L. Yin, W. Zheng, Developing multi-labelled corpus of twitter short texts: a semi-automatic method, Systems 11 (8) (2023) 390, https://doi.org/10.3390/systems11080390, doi.

[95] S. Wang, H. Sheng, Y. Zhang, D. Yang, J. Shen, R. Chen, Blockchain-empowered distributed multi-camera multi-target tracking in edge computing, IEEE Trans. Ind. Inf. (2023), https://doi.org/10.1109/TII.2023.3261890.

[96] H. Li, et al., A cooperative defense framework against application-level DDoS attacks on mobile edge computing services, IEEE Trans. Mob. Comput. (2021) 1, https://doi.org/10.1109/tmc.2021.3086219, 1Available[Accessed 13 February 2022].

[97] Y. Wang, X. Han, S. Jin, MAP based modeling method and performance study of a task offloading scheme with time-correlated traffic and VM repair in MEC systems, Wirel. Netw. (2022), https://doi.org/10.1007/s11276-022-03099-2.

[98] K. Bhardwaj, J.C. Miranda, A. Gavrilovska, Towards IoT-DDoS Prevention Using Edge Computing, 18, HotEdge', Boston, MA, USA, 2018.

[99] C. Tzagkarakis, N. Petroulakis, S. Ioannidis, Botnet attack detection at the IoT edge based on sparse representation, 2019 Global IoT Summit (GIoTS) (2019), https://doi.org/10.1109/giots.2019.8766388. Available[Accessed 3 June 2022].

[100] Y. Meidan, et al., N-BaIoT—network-based detection of IoT botnet attacks using deep autoencoders", IEEE Pervasive Comput. 17 (3) (2018) 12–22, https://doi.org/10.1109/mprv.2018.03367731. Available[Accessed 3 June 2022].

[101] A. Kumar, M. Shridhar, S. Swaminathan, T. Lim, Machine learning-based early detection of IoT botnets using network-edge traffic, Comput. Secur. 117 (2022), 102693, https://doi.org/10.1016/j.cose.2022.102693. Available[Accessed 4 June 2022].

[102] H. Liao, Z. Zhou, X. Zhao, L. Zhang, S. Mumtaz, A. Jolfaei, S.H. Ahmed, A. K. Bashir, Learning-based context-aware resource allocation for edge-computing-empowered industrial IOT, IEEE Internet of Things J. 7 (5) (2020) 4260–4277.

[103] S.A. Kumar, T. Vealey, H. Srivastava, Security in internet of things: challenges, solutions and future directions, in: 2016 49th Hawaii International Conference on System Sciences (HICSS), IEEE, 2016, pp. 5772–5781.

[104] J. Firch, How To Prevent A ICMP Flood Attack, PurpleSec, 2022 [Online]. Available, https://purplesec.us/prevent-ping-attacks/ [Accessed: 17- Feb- 2022].

[105] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, W. Lv, Edge computing security: state of the art and challenges, Proc. IEEE 107 (8) (2019) 1608–1631, https://doi.org/10.1109/jproc.2019.2918437. Available[Accessed 25 February 2022].

[106] K. Sahu, R. Kshirsagar, S. Vasudeva, T. Alzahrani, N. Karimian, Leveraging timing side-channel information and machine learning for IoT security, in: 2021 IEEE International Conference on Consumer Electronics (ICCE), 2021, https://doi.org/10.1109/icce50685.2021.9427585. Available[Accessed 13 February 2022].

[107] M. Rathore, M. Usman, J. Kim, Maintaining SmartX multi-view visibility for OF@TEIN+ distributed cloud-native edge boxes, Trans. Emerg. Telecommun. Technol. 32 (6) (2020), https://doi.org/10.1002/ett.4101. Available[Accessed 26 February 2022].

[108] M. Hossain, H. Ochiai, D. Fall, Y. Kadobayashi, LSTM-based network attack detection: performance comparison by hyper-parameter values tuning, in: 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud, 2020, https://doi.org/10.1109/cscloud-edgecom49738.2020.00020. EdgeComAvailable[Accessed 27 February 2022].

[109] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, W. Lv, Edge computing security: state of the art and challenges, Proc. IEEE 107 (8) (2019) 1608–1631, https://doi.org/10.1109/jproc.2019.2918437. Available[Accessed 25 February 2022].

[110] "What is a Zero-Day Exploit | Protecting Against 0day Vulnerabilities | Imperva", Learning center, 2021. [Online]. Available: https://www.imperva.com/learn/application-security/zero-day-exploit/. [Accessed: 04- Mar- 2022].

[111] S. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, O. Jogunola, Federated deep learning for zero-day botnet attack detection in IoT-edge devices, IEEE Internet of Things J. 9 (5) (2022) 3930–3944, https://doi.org/10.1109/jiot.2021.3100755. Available[Accessed 19 May 2022].

[112] A. Rangisetti, R. Dwivedi, P. Singh, Denial of ARP spoofing in SDN and NFV enabled cloud-fog-edge platforms, Cluster Comput. 24 (4) (2021) 3147–3172, https://doi.org/10.1007/s10586-021-03328-x. Available[Accessed 23 June 2022].

[113] J. Cox, R. Clark, H. Owen, Leveraging SDN For ARP Security, SoutheastCon, 2016, https://doi.org/10.1109/secon.2016.7506644, 2016Available[Accessed 24 June 2022].

[114] J. Xia, Z. Cai, G. Hu, M. Xu, An active defense solution for ARP spoofing in OpenFlow network, Chin. J. Electr. 28 (1) (2019) 172–178, https://doi.org/10.1049/cje.2017.12.002. Available[Accessed 24 June 2022].

[115] O. Osanaiye, A. Alfa, G. Hancke, A statistical approach to detect jamming attacks in wireless sensor networks, Sensors 18 (6) (2018) 1691, https://doi.org/10.3390/s18061691. Available[Accessed 26 June 2022].

[116] K. Pelechrinis, M. Iliofotou, S. Krishnamurthy, Denial of service attacks in wireless networks: the case of jammers, IEEE Commun. Surv. Tutor. 13 (2) (2011) 245–257, https://doi.org/10.1109/surv.2011.041110.00022. Available[Accessed 26 June 2022].

[117] M. Li, I. Koutsopoulos, R. Poovendran, Optimal jamming attacks and network defense policies in wireless sensor networks, in: IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications, 2007, https://doi.org/10.1109/infcom.2007.155. Available[Accessed 26 June 2022].

[118] "Wireless Security - Layer 1 DoS", Tutorialspoint.com, 2022. [Online]. Available: https://www.tutorialspoint.com/wireless_security/wireless_security_layer1_dos.htm. [Accessed: 26- Jun- 2022].

[119] B. Li, T. Chen, X. Wang, G. Giannakis, Secure edge computing in IoT via online learning, in: 2018 52nd Asilomar Conference on Signals, Systems, and Computers, 2018, https://doi.org/10.1109/acssc.2018.8645223. Available[Accessed 27 June 2022].

[120] X. Dai, Z. Xiao, H. Jiang, M. Alazab, J.C.S. Lui, S. Dustdar, J. Liu, Task co-offloading for D2D-assisted mobile edge computing in industrial internet of things, IEEE Trans. Ind. Inf. 19 (1) (2023) 480–490, https://doi.org/10.1109/TII.2022.3158974.

[121] H. Jiang, X. Dai, Z. Xiao, A.K. Iyengar, Joint task offloading and resource allocation for energy-constrained mobile edge computing, IEEE Trans. Mob. Comput. (2022), https://doi.org/10.1109/TMC.2022.3150432ZZFP23.

[122] Y. Alharbi, A. Alferaidi, K. Yadav, G. Dhiman, S. Kautish, Denial-of-service attack detection over IPv6 network based on KNN algorithm, Wirel. Commun. Mob. Comput. (2021) 1–6, https://doi.org/10.1155/2021/8000869, 2021Available [Accessed 1 July 2022].

[123] "What is a Buffer Overflow | Attack Types and Prevention Methods | Imperva", Learning center, 2022. [Online]. Available: https://www.imperva.com/learn/application-security/buffer-overflow/. [Accessed: 03- Jul- 2022].

[124] "IBM Docs", Ibm.com, 2022. [Online]. Available: https://www.ibm.com/docs/en/zos/2.4.0?topic=overview-address-space-layout-randomization. [Accessed: 03-Jul- 2022].
[125] R. Sheldon, Discover three key exploit protection features in Windows 10, SearchEnterpriseDesktop (2018) [Online]. Available, https://www.techtarget.com/searchenterprisedesktop/tip/Discover-three-key-exploit-protection-features-in-Windows-10#:~:text=randomized%20memory%20allocations.-,Structured%20Exception%20Handling%20Overwrite%20Protection,managing%20hardware%20and%20software%20exceptions [Accessed: 03- Jul- 2022].
[126] Z. Tian, et al., Real-time lateral movement detection based on evidence reasoning network for edge computing environment, IEEE Trans. Ind. Inf. 15 (7) (2019) 4285–4294, https://doi.org/10.1109/tii.2019.2907754. Available[Accessed 4 July 2022].
[127] "Q2 2018 DDoS Trends Report: 52 Percent of attacks employed multiple attack types", Circleid.com, 2018. [Online].Available:https://circleid.com/posts/20180927_q2_2018_ddos_trends_report_52_percent_of_attacks_multiple_types. [Accessed: 16- May- 2022].
[128] R.F. Hayat, S. Aurangzeb, M. Aleem, G. Srivastava, Lin JC. ML-DDoS, A blockchain-based multilevel DDoS mitigation mechanism for IoT environments, IEEE Trans. Eng. Manag. (2022 May 13).
[129] "Introducing the mac of Tool - Port Security - Cisco Certified Expert", Cisco certified expert, 2022. [Online]. Available: https://www.ccexpert.us/port-security/introducing-the-macof-tool.html. [Accessed: 26- Jun- 2022].
[130] X. Liu, T. Shi, G. Zhou, M. Liu, Z. Yin, L. Yin, W. Zheng, Emotion classification for short texts: an improved multi-label method, Humanit. Soc. Sci. Commun. 10 (1) (2023) 306, https://doi.org/10.1057/s41599-023-01816-6.
[131] D. Wang, B. Bai, K. Lei, W. Zhao, Y. Yang, Z. Han, Enhancing information security via physical layer approaches in heterogeneous IoT with multiple access mobile edge computing in smart city, IEEE Access 7 (2019) 54508–54521.
[132] H. Zhang, Y. Mi, Y. Fu, X. Liu, Y. Zhang, J. Wang, J. Tan, Security defense decision method based on potential differential game for complex networks, Comput. Secur. 129 (2023), 103187, https://doi.org/10.1016/j.cose.2023.103187.
[133] S. Lu, M. Liu, L. Yin, Z. Yin, X. Liu, W. Zheng, X. Kong, The multi-modal fusion in visual question answering: a review of attention mechanisms, PeerJ Comput. Sci. 9 (2023) e1400, https://doi.org/10.7717/peerj-cs.1400.
[134] R. Uddin, F. Monir, Performance evaluation of ryu controller with weighted round Robin load balancer, Commun. Comput. Inf. Sci. (2021) 115–129.
[135] R. Uddin, S.A. Kumar, SDN-based federated learning approach for satellite-IOT framework to enhance data security and privacy in space communication, IEEE J. Radio Freq. Identif. 7 (2023) 424–440.
[136] R. Uddin, S. Kumar, Federated learning based intrusion detection system for satellite communication, in: 2023 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW), 2023.
[137] "IBM Developer", Developer.ibm.com, 2022. [Online]. Available: https://developer.ibm.com/articles/edge-computing-application-and-device-layer/. [Accessed: 28- Jul- 2022].
[138] "IBM Docs", Ibm.com, 2022. [Online]. Available: https://www.ibm.com/docs/en/zos/2.4.0?topic=overview-address-space-layout-randomization. [Accessed: 03-Jul- 2022].
[139] R. Sheldon, Discover three key exploit protection features in Windows 10, SearchEnterpriseDesktop (2018) [Online]. Available, https://www.techtarget.com/searchenterprisedesktop/tip/Discover-three-key-exploit-protection-features-in-Windows-10#:~:text=randomized%20memory%20allocations.-,Structured%20Exception%20Handling%20Overwrite%20Protection,managing%20hardware%20and%20software%20exceptions [Accessed: 03- Jul- 2022].
[140] J. Li, Y. Deng, W. Sun, W. Li, R. Li, Q. Li, Z. Liu, Resource orchestration of cloud-edge–based smart grid fault detection, ACM Trans. Sen. Netw. 18 (3) (2022), https://doi.org/10.1145/3529509grid.
[141] P. Sinha, V. Jha, A. Rai, B. Bhushan, Security vulnerabilities, attacks and countermeasures in wireless sensor networks at various layers of OSI reference model: a survey, in: 2017 International Conference on Signal Processing and Communication (ICSPC, 2017.
[142] S. Moosavi, T. Gia, A. Rahmani, E. Nigussie, S. Virtanen, J. Isoaho, H. Tenhunen, SEA: a secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways, Procedia Comput. Sci. 52 (2015) 452–459.
[143] Z. Liu, P. Qian, J. Yang, L. Liu, X. Xu, Q. He, X. Zhang, Rethinking smart contract fuzzing: fuzzing with invocation ordering and important branch revisiting, IEEE Trans. Inf. Forensics Secur. 18 (2023) 1237–1251, https://doi.org/10.1109/TIFS.2023.3237370.
[144] H. Kim, E. Kang, D. Broman, E. Lee, Resilient authentication and authorization for the internet of things (IoT) using edge computing, ACM Trans. Internet of Things 1 (1) (2020) 1–27.
[145] "What is ARP Spoofing and How to Prevent it?", www.ctemplar.com, 2022. [Online]. Available: https://ctemplar.com/what-is-arp-spoofing-and-how-to-prevent-it/. [Accessed: 23- Jun- 2022].
[146] A. Rangisetti, R. Dwivedi, P. Singh, Denial of ARP spoofing in SDN and NFV enabled cloud-fog-edge platforms, Cluster Comput. 24 (4) (2021) 3147–3172, https://doi.org/10.1007/s10586-021-03328-x. Available[Accessed 23 June 2022].
[147] J. Cox, R. Clark, H. Owen, Leveraging SDN For ARP Security, SoutheastCon, 2016, https://doi.org/10.1109/secon.2016.7506644, 2016Available[Accessed 24 June 2022].
[148] J. Xia, Z. Cai, G. Hu, M. Xu, An active defense solution for ARP spoofing in OpenFlow network, Chin. J. Electr. 28 (1) (2019) 172–178, https://doi.org/10.1049/cje.2017.12.002. Available[Accessed 24 June 2022].
[149] K. Cao, et al., Enhancing physical layer security for IoT with non-orthogonal multiple access assisted semi-grant-free transmission, IEEE Internet of Things J. (2022), https://doi.org/10.1109/JIOT.2022.3193189.
[150] D. Wang, B. Bai, K. Lei, W. Zhao, Y. Yang, Z. Han, Enhancing information security via physical layer approaches in heterogeneous IoT with multiple access mobile edge computing in smart city, IEEE Access 7 (2019) 54508–54521.
[151] Y. Liu, J. Li, A. Petropulu, Destination assisted cooperative jamming for wireless physical-layer security, IEEE Trans. Inf. Forensics Secur. 8 (4) (2013) 682–694.
[152] F. Qiao, Z. Li, Y. Kong, A privacy-aware and incremental defense method against GAN-based poisoning attack, IEEE Trans. Computationapol Soc. Syst. (2023), https://doi.org/10.1109/TCSS.2023.3263241.
[153] R. Uddin, M.F. Monir, Performance analysis of SDN based firewalls: pox vs. ODL, in: 2019 5th International Conference on Advances in Electrical Engineering (ICAEE), 2019.
[154] R. Uddin, M.F. Monir, Evaluation of four SDN controllers with firewall modules, in: Proceedings of the International Conference on Computing Advancements, 2020.
[155] Y. Song, R. Xin, P. Chen, R. Zhang, J. Chen, Z. Zhao, Identifying performance anomalies in fluctuating cloud environments: a robust correlative-GNN-based explainable approach, Fut. Gener. Comput. Syst. 145 (2023) 77–86, https://doi.org/10.1016/j.future.2023.03.020.
[156] B. Li, X. Zhou, Z. Ning, X. Guan, K.C. Yiu, Dynamic event-triggered security control for networked control systems with cyber-attacks: a model predictive control approach, Inf. Sci. (Ny) 612 (2022) 384–398, https://doi.org/10.1016/j.ins.2022.08.093.
[157] R. Firouzi, R. Rahmani, A distributed SDN controller for distributed IoT, IEEE Access 10 (2022) 42873–42882.
[158] Y. Ding, W. Zhang, X. Zhou, Q. Liao, Q. Luo, L.M. Ni, FraudTrip: taxi fraudulent trip detection from corresponding trajectories, IEEE Internet of Things J. 8 (16) (2021) 12505–12517, https://doi.org/10.1109/JIOT.2020.3019398.
[159] M. Majid, S. Habib, A.R. Javed, M. Rizwan, G. Srivastava, T.R. Gadekallu, J.C. Lin, Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: a systematic literature review, Sensors 22 (6) (2022) 2087. Mar 8.
[160] B. Cheng, D. Zhu, S. Zhao, J. Chen, Situation-aware IoT service coordination using the event-driven SOA paradigm, IEEE Trans. Netw. Serv. Manag. 13 (2) (2016) 349–361, https://doi.org/10.1109/TNSM.2016.2541171.
[161] W. Zheng, P. Deng, K. Gui, X. Wu, An abstract syntax tree based static fuzzing mutation for vulnerability evolution analysis, Inf. Softw. Technol. (2023), 107194, https://doi.org/10.1016/j.infsof.2023.107194.
[162] J. Zhang, S. Peng, Y. Gao, Z. Zhang, Q. Hong, APMSA: adversarial perturbation against model stealing attacks, IEEE Trans. Inf. Forensics Secur. (2023) 18, https://doi.org/10.1109/TIFS.2023.3246766.
[163] J. Tan, H. Jin, H. Hu, R. Hu, H. Zhang, H. Zhang, WF-MTD: evolutionary decision method for moving target defense based on Wright-Fisher process, IEEE Trans. Dependable Secure Comput. (2022), https://doi.org/10.1109/TDSC.2022.3232537.
[164] H.Y. Jin, Z. Wang, Asymptotic dynamics of the one-dimensional attraction-repulsion Keller-Segel model, Math. Methods Appl. Sci. 38 (3) (2015) 444–457, https://doi.org/10.1002/mma.3080.
[165] J. Sakhnini, H. Karimipour, A. Dehghantanha, R.M. Parizi, G. Srivastava, Security aspects of Internet of Things aided smart grids: a bibliometric survey, Internet of Things 14 (2021 Jun 1), 100111.
[166] R.K. Dhanaraj, R.H. Jhaveri, L. Krishnasamy, G. Srivastava, P.K. Maddikunta, Black-hole attack mitigation in medical sensor networks using the enhanced gravitational search algorithm, Int. J. Uncertainty, Fuzziness Knowl.-Based Syst. 29 (2) (2021) 297–315. DecSuppl.
[167] M. Monir, R. Uddin, D. Pan, Behavior of NAPT middleware in an SDN environment, in: 2019 4th International Conference on Electrical Information and Communication Technology (EICT), 2019, https://doi.org/10.1109/eict48899.2019.9068752. Available[Accessed 17 June 2022].
[168] M.F. Monir, R. Uddin, D. Pan, Implementation of a click based ids on SDN-NFV architecture and performance evaluation, in: 2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), 2021.
[169] J. Singh, M. Wazid, A.K. Das, V. Chamola, M. Guizani, Machine learning security attacks and defense approaches for emerging cyber physical applications: a comprehensive survey, Comput. Commun. 192 (2022) 316–331.
[170] V. Hassija, V. Chamola, B.C. Bajpai, Naren, S. Zeadally, Security issues in implantable medical devices: fact or fiction? Sustain. Cities Soc. 66 (2021), 102552.
[171] M. Wazid, A.K. Das, V. Chamola, Y. Park, Uniting cyber security and machine learning: advantages, challenges and future research, ICT Express 8 (3) (2022) 313–321.
[172] A. Vangala, A.K. Das, V. Chamola, V. Korotaev, J.J. Rodrigues, Security in IOT-enabled smart agriculture: architecture, security solutions and challenges, Cluster Comput. 26 (2) (2022) 879–902.
[173] K. Cao, B. Wang, H. Ding, L. Lv, R. Dong, T. Cheng, F. Gong, Improving physical layer security of Uplink NOMA via energy harvesting jammers, IEEE Trans. Inf. Forensics Secur. 16 (2021) 786–799, https://doi.org/10.1109/TIFS.2020.3023277.
[174] S. Han, H. Ding, S. Zhao, S. Ren, Z. Wang, J. Lin, S. Zhou, Practical and robust federated learning with highly scalable regression training, IEEE Trans. Neural Netw. Learn. Syst. (2023), https://doi.org/10.1109/TNNLS.2023.3271859.
[175] Y. Yao, J. Zhao, Z. Li, X. Cheng, L. Wu, Jamming and eavesdropping defense scheme based on deep reinforcement learning in autonomous vehicle networks,

IEEE Trans. Inf. Forensics Secur. 18 (2023) 1211–1224, https://doi.org/10.1109/TIFS.2023.3236788.
[176] H. McMahan, E. Moore, D. Ramage, S. Hampson, B. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: 20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA, 2017.
[177] T. Li, A. Kumar Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, "Federated optimization in heterogeneous networks", 2018 [Online]. Available: https://arxiv.org/abs/1812.06127. [Accessed: 14- Jul- 2022].
[178] T. Li, A. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar and V. Smithy, "FedDANE: a federated Newton-type method", 2019.

Ryhan Uddin received the bachelor's degree in electronics and telecommunication engineering from North South University, Dhaka, Bangladesh, in 2016 and master's degree in computer science from American International University Bangladesh, in 2020. Currently he is pursuing Ph.D. degree while working as a research assistant at the Intelligent Secure Cyber Systems and Applications Research (ISCAR) Lab at Cleveland State University, Ohio, USA. Earlier in his career, he served as a system engineer for about 7 years at Grameen CyberNet Ltd., Dhaka, Bangladesh, where he worked on data center networks, Linux platforms and virtual machines. His-research interests include Software Defined Networking (SDN), computer networks and security, machine learning and edge computing

Sathish A.P. Kumar is an Associate Professor of Computer Science in the Department of Electrical Engineering and Computer Science at the Cleveland State University, Cleveland, Ohio, USA. He is directing Intelligent Secure Cyber-Systems Analytics and Applications Research (ISCAR) Lab at the Cleveland State University. He earned his PhD degree in Computer Science and Engineering from the University of Louisville, Kentucky, USA in 2007. He is a Senior Member of IEEE. His-current research interests are in cybersecurity, machine learning, big data analytics and secure distributed systems and their applications. He has published more than 70 technical papers in international journals and conference proceedings. He currently serves as an Associate editor for IEEE Access, PLOS One, Elsevier Machine Learning with Applications and as an editorial board member for Nature Scientific Reports.

Vinay Chamola received the B.E. degree in electrical and electronics engineering and the master's degree in communication engineering from the Birla Institute of Technology and Science, Pilani, Pilani, India, in 2010 and 2013, respectively, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2016. In 2015, he was a Visiting Researcher with the Autonomous Networks Research Group (ANRG), University of Southern California, Los Angeles, CA, USA. He also worked as a Postdoctoral Research Fellow at the National University of Singapore. He is currently an Associate Professor with the Department of Electrical and Electronics Engineering, BITS-Pilani, where he heads the Internet of Things Research Group/Laboratory. He is also the Co-Founder and the President of a healthcare startup Medsupervision Pvt. Ltd. His-research interests include the IoT security, blockchain, UAVs, VANETs, 5G, and healthcare. He serves as an Area Editor for the Ad Hoc Networks journal, Elsevier, and the IEEE Internet of Things Magazine. He also serves as an Associate Editor for IEEE Transactions on Intelligent Transportation Systems, IEEE Networking Letters, IEEE Consumer Electronics Magazine, IET Quantum Communications, IET Networks, and several other journals. He serves as the Co-Chair for various reputed workshops like IEEE GLOBECOM Workshop 2021, IEEE INFOCOM 2022 Workshop, IEEE ANTS 2021, and IEEE ICIAfS 2021. He is listed in the World's Top 2 % Scientists identified by Stanford University. He is a Fellow of the IET.