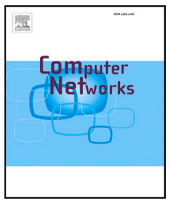
Computer Networks 270 (2025) 111433

Contents lists available at ScienceDirect

Computer Networks

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



Survey paper

Transformers model for DDoS attack detection: A survey

Euclides Peres Farias Junior [a,*], Anderson Bergamini de Neira [a], Ligia Fracielle Borges [b], Michele Nogueira [a,b]

[a] Department of Informatics, Federal University of Paran/'a (UFPR), Brazil
[b] Department of Computer Science, Federal University of Minas Gerais (UFMG), Brazil

ARTICLE INFO

Keywords:
DDoS attacks detection
Transformers model
Network security
IoT
SDN

ABSTRACT

Distributed Denial of Service (DDoS) attack detection through Transformer models is one of the innovative Deep Learning applications. DDoS attacks are hard to handle and there is no definitive solution. Therefore, detecting DDoS attacks based on the Transformer architecture are being widely explored because of its versatility and customization. Transformer architectures analyze network traffic and identify malicious patterns, given different advantages from these architectures, such as the processing capacity in long sequences, the attention mechanism (a.k.a., self-attention) aimed at capturing complex patterns in the identification of malicious traffic, real-time detection through parallelism, the generalization to new types of attacks and, finally, the complete integration with other artificial intelligence techniques. Therefore, this survey is an extensive literature review providing an overview of the Transformer Architecture through different applied models, strategies for data preprocessing, and applications in various types of data, including real-time, address different machine learning techniques and deep learning. Thus, it analyzed 45 papers that focus on detecting DDoS attacks. The F1-Score of the DDoS attack detection identified in the papers varies between 47.40% and 100.00%. This survey contributes to the understanding of relevant aspects in different models applied in transformer architecture and thus emphasizes open issues and research directions.

1. Introduction

Distributed denial of service (DDoS) attacks are one of the hardest cyber attacks to handle on the Internet. They are dynamic, acting in a distributed and coordinated manner to consume the total resources of both the network and the servers [1]. This type of attack follows different methodologies and architectures, causing damage to victims indiscriminately all over the world. The company Cloudflare published a study on DDoS attacks, showing that, in the first half of 2024, 8.5 million DDoS attacks were mitigated. Of these, 4.5 million occurred in the first quarter alone, and the other 4 million in the second quarter of 2024. The company also ranked the most attacked countries in the second quarter of 2024, highlighting China first, followed by Turkey, Hong Kong, Russia, Brazil, and Thailand. In addition to the global reach of DDoS attacks, the company Exploding Topics [2] projects that approximately 181 zettabytes of data will be transmitted and generated by the end of 2025. This massive volume of data requires the connection of numerous devices to manage the flow of information, which creates an environment vulnerable to DDoS attacks [3].

The application of AI, through the use of advanced Machine Learning (ML) and Deep Learning (DL) techniques, are promising resources in the implementation of intelligent solutions to detect DDoS attacks [4]. These resources are able to enable multiple intelligent operations such as the accurate classification of legitimate and malicious traffic, early attack detection, behavioral analysis of network traffic, real-time attack detection, and attack prediction, thus becoming one of the most assertive and efficient tools in detecting DDoS attacks [5,6]. However, cybercrimes also exploit AI, ML, and DL to improve, update, and build DDoS attacks, significantly increasing the concern for cybersecurity [7]. Hence, the use of Artificial Intelligence (AI) through various techniques, as Neural Networks (NN), has attracted attention, and it is becoming increasingly essential to detect DDoS attacks.

Transformer Architecture (TA) is the new generation of Neural Networks (NN) that has revolutionized the field of AI, mainly in the areas of Natural Language Processing (NLP) and Computer Vision [8], becoming one of the most innovative techniques explored in different cybersecurity domains [9]. This architecture has a variety of functionalities initiated by the structure at several levels that can be modified and customized at any time in both the Encoder and Decoder layers. In addition to allow parallel and distributed processing, significantly increasing its efficiency, mainly in actions that require real-time processing, factors that make TA a promising resource in the search

* Corresponding author.
E-mail address: epfjunior@inf.ufpr.br (E.P. Farias Junior).

https://doi.org/10.1016/j.comnet.2025.111433
Received 20 February 2025; Received in revised form 30 May 2025; Accepted 31 May 2025
Available online 23 June 2025

for solutions to detect DDoS attacks [10,11]. Hence, a review of the literature on TA applied to the detection of DDoS attacks shows the potential of this innovative architecture.

The literature has explored Transformer architectures as a primary tool for anomaly detection. However, the authors of this survey have not found a complete literature survey on Transformer models applied to DDoS attacks detection. Several authors have published papers exploring TA for building Intrusion Detection Systems (IDS), focusing their efforts on anomaly detection and including DDoS attacks [12–15]. The existing papers address attention-based models in a normal way (i.e., when there are no changes in TA) or in a hybrid way (i.e., when there is a diversity of ML or DL techniques together within the TA), exploring ML, DL, Adversarial Networks (GAN), Convolutional Neural Networks (CNN), unsupervised learning, semi-supervised learning, Large Language Models Scale (LLMs), Long Short Term Memory (LSTM), Computer Vision Transformer (ViTs), among others [16–19]. TA has also explored DDoS attack detection in different networks, such as the Internet of Things (IoT), Internet of Medical Things (IoTM), Industrial Internet of Things (IIoT) and Smart Grid (SG), among others [20–23].

Despite this broad use of TA addressing DDoS attack detection, they mostly present challenges ranging from model complexity, lack of practical use, lack of quality in training data, real-time implementations, model interpretation, low performance on imbalanced data, and dependence on labeled data. Those are the main criticisms raised in the literature. Therefore, the lack of a survey focused on showing how TA and its variants have been employed for DDoS attack detection reveals a significant gap, as these two topics are highly relevant to the literature.

Hence, this survey presents a comprehensive literature review focusing on the use of TA in detecting DDoS attacks. The relevance of this study is based on the fact that, although TA was introduced in 2017 [11], its use in cybersecurity, mainly for detecting DDoS attacks, started to be explored only in 2018. Knowing that the application of TA in cybersecurity is an emerging field, this paper aims to fill a significant gap in the literature by providing a detailed analysis of works that use TA techniques and variants to identify and mitigate DDoS attacks. The importance of carrying out this study is due to the growing need for advanced and effective solutions to combat increasingly sophisticated cyber threats. Hence, this survey examines the use of Transformer models for detecting DDoS attacks. It classifies existing papers on applying the Transformer Architecture to detect DDoS attacks.

Also, this survey delves into various machine learning and deep learning approaches, as well as preprocessing methods necessary for effectively implementing Transformer Models. This survey explores the application of Transformer models in various computing environments, such as computer networks, SDN, IoT, IoTM, IIoT, and SG. Providing a comprehensive understanding of DDoS attacks and the various techniques proposed to detect them through the Transformer Architecture is one of the main contributions of this research. Frequently used acronyms are listed in Table 1. The main contributions of this survey are as follows.

- A systematic analysis and a review of papers from 2018 to 2024 on the use of TA to detect DDoS attacks, highlighting its evolution since its beginning in 2017.
- A new classification framework for detection solutions based on the Transformer Architecture, emphasizing the analysis of diverse data types (e.g., labeled, real-time, and static data), which provides insights into the optimization of DDoS detection strategies.
- An investigation of machine learning, deep learning, and preprocessing methods essential for TA implementation, presenting a critical assessment of their effectiveness in detecting DDoS attacks. An exploratory view of the deployment of Transformer models in various computing contexts, such as computer networks, SDN, IoT, IoTM, IIoT, and SG, emphasizing their adaptability and robustness against DDoS attacks. A detailed analysis of open issues, learned lessons, and challenges in applying TA to detect DDoS attacks. This offers a critical and comprehensive view, helping identify opportunities for theoretical and practical advancement, both in the use of TA and network security.

This survey proceeds as follows. Section 2 presents the background for the topic. Section 3 explains the methodology followed in this work and the proposed classification for the existing works. Section 4 provides a comprehensive overview of the selected papers related to the application of the Transformer Model for detecting DDoS attacks. Section 5 builds on the findings to analyze the application of Transformer models to DDoS detection. Section 6 presents and discusses the open issues, challenges, and research opportunities that have not yet been fully explored. Finally, Section 7 concludes the survey, highlighting research directions and issues. Table 1 summarizes the acronyms and notations used in this survey.

2. Background

This section presents the main concepts to understand transformer models applied to DDoS attack detection. It presents also an overview of the main characteristics of DDoS attacks and their impact on the Internet, as well as it presents the challenges that involve the process of detecting DDoS attacks. This section highlights the main characteristics of traditional DDoS detection methods for comparison purposes. They may use ML and DL. Furthermore, this section details the emergence and evolution of transformer models, as well as the properties and why transformer models have become a promising alternative in the analysis for detecting attacks in real-time. Finally, this section provides the opportunity for a solid understanding of the historical, technical, and scientific context that can justify the study of Transformer models.

2.1. DDoS attacks

Denial of Service Attack (DoS) aims to prevent legitimate users from accessing services and resources, such as network resources on a website, web service, or even on a computer system [7,24]. Distributed Denial of Service (DDoS) attack is an evolution of the DoS attack [24]. DDoS attacks have become increasingly precise, aggressive, and efficient, as highlighted by [1]. Attackers now exploit specific application vulnerabilities, exemplified by attacks on the application layer that deplete web server resources. Additionally, using botnets, consisting of numerous compromised devices, enables attackers to generate immense traffic volumes, further complicating defensive measures. Fig. 1 illustrates the basic principle of DoS and DDoS attacks, which involve overwhelming a server with a large amount of malicious traffic. These attacks target various types of services, such as networks, web servers, cloud hosting, DNS (Domain Name System), online gaming, streaming platforms, VoIP (Voice over Internet Protocol), as well as financial services like banking and online payments [7]. The illustration includes three scenarios across two network topologies: Local Area Network (LAN) and Wide Area Network (WAN). For a better understanding in Fig. 1, the connections are represented in three different types: solid black arrow, which means the connections for both LAN and WAN in normal state. The dotted arrows and three red dashes indicate connections to LAN and WAN in a DoS attack state. Finally, the solid red lines, without dots, represent DDoS attacks for LAN and WAN. The first scenario shows a network with a real user and no attack, connected through LAN and WAN to access the application server. The second scenario demonstrates a DoS attack via LAN and WAN, where one computer sends excessive data loads to deplete the server's resources. Lastly, the DDoS scenario shows multiple hosts coordinating to attack the application server in a distributed and service-based manner.

The cybersecurity community, research centers, universities, and Internet service providers have focused significant attention on addressing the threat of DDoS attacks [25,26]. Over the years, this type of

Table 1
List of acronyms and notations used in this survey.

Notation	Description	Notation	Description
ACL	Access Control Lists	*ADL*	Adversarial Deep Learning
ADT	Adaptive Transformer	*AET*	Attention-Enhanced Transformer
AET	Attention-Enhanced Transformer	*AI*	Artificial Intelligence
AIS	Artificial Immune System	*ANN*	Artificial Neural Network
API	Application Programming Interface	*BC*	Behavior Cloning
BERT	Bidirectional Encoder Representations from Transformer	*BLSTM*	Bidirectional Long Short-Term Memory
BN	Batch Normalization	*CAN*	Controller Area Networks
CART	Classification and Regression Trees	*CDN*	Content Delivery Network
CIDS	Combined Intrusion detection System	*CNN*	Convolutional Neural Network
CPS	Cyber–physical Systems	*CQL*	Conserve Q-Learning
CSL	Custom Transformer	*CVAE*	Conditional Variational AutoEncoders
DDoS	Distributed Denial of Service	*DL*	Deep Learning
DM	Data Mining	*DNN*	Deep Neural Network
DNS	Domain Name System	*DoS*	Denial of Service
DP	Distributed Processing	*DR*	Detection Rate
DRL	Deep Reinforcement Learning	*DT*	Decision Tree
ECOD	Ensemble Classifier for Outliers Detection	*EQ*	Evaluation Questions
FEDformer	Frequency Enhanced Decomposed Transformer	*FFN*	Fuzzy Neural Network
FPR	False Positive Rate	*GCNN*	Graph Convolutional Neural Networks
GFEM	Global Feature Extraction	*GRU*	Gated Recurrent Units
HBT	Hybrid Transformer	*HSI*	Hyperspectral Image
HTTP	Hypertext Transfer Protocol	*ICMP*	Internet Control Message Protocol
IDS	Intrusion detection System	*IF*	Isolation Forest
IIoT	Industrial Internet of Things	*IoT*	Internet of Things
IoTM	Internet of Medical Things	*IP*	Internet Protocol
IPS	Intrusion Prevention System	*IRC*	Internet Relay Chat
K-NN	K-Nearest Neighbor	*LAN*	Local Area Network
LDA	Linear Discriminant Analysis	*LFEM*	Local Feature Extraction Module
LLMs	Large Language Models Scale	*LR*	Logistic Regression
LSTM	Long Short-Term Memory	*MCU*	Microcontroller Unit
MDS	Misbehavior Detection System	*MITM*	Man-in-the-Middle
ML	Machine Learning	*MLP*	Multi-layer Perceptron
MTNN	Modified Transformer Neural Network	*NB*	Naïve Bayes
NLP	Natural Language Processing	*NN*	Neural Network
OEIF	On-demand Evolving Isolation Forest	*P2P*	Peer-to-peer Protocol
PatchTST	Patch Time Series Transformer	*PCAP*	Packet Capture
PWT	Packet Window Transformer	*RC*	Residual Connections
ReLU	Rectified Linear Unit	*RF*	Random Forest
RL	Reinforcement Learning	*RNN*	Recurrent Neural Network
ROC	Receive Operating Characteristic	*RPT*	Real-Time Processing Transformer
SDN	Software-Defined Networking	*SG*	Smart Grid
SLR	Systematic Literature Review	*SMB*	Server Message Block
SMOTE	Synthetic Minority Over-Sampling Technique	*SMTP*	Simple Mail Transfer Protocol
SRL	Systematic Literature Review	*SSH*	Secure Shell
STD	Standard Transformer	*SVM*	Support Vector Machine
TA	Transformer Architecture	*TAN*	Transformer Attention Networks
TCP	Transmission Control Protocol	*TLC*	Transformer LSTM CNN
TL	Transfer Learning	*TM*	Transformer Model
TPR	True Positive Rate	*UDP*	User Datagram Protocol
VC	Variant Classification	*ViT*	Vision Transformer
VoIP	Voice over Internet Protocol	*WAF*	Web Application Firewall
WAN	Wide Area Network	*XGB*	eXtreme Gradient Boosting
XSS	Cross-Site Scripting	–	–

attack has continued to claim countless victims indiscriminately and can be directed at any system connected to the network, regardless of its nature or purpose [27,28]. This characteristic of indiscriminateness is one of the reasons why DDoS attacks can have such a significant impact, as they can affect a wide range of targets [1,29]. Although DDoS appears to be a relatively simple type of attack [30], its efficiency, objectivity, and aggressiveness make it one of the biggest threats to the Internet, causing significant damage across the world [25,31,32].

DDoS attacks are effective because (*i*) they use botnets that exploit compromised devices under the attacker's control; (*ii*) bot-generated traffic often mimics legitimate traffic, making it difficult to detect; (*iii*) they use spoofed IP addresses, which makes it challenging to identify the attack sources due to the variety of IPs used; (*iv*) attackers deploy a wide range of methods that combine different protocols and techniques, such as reflection and amplification attacks; (*v*) the impact caused by these attacks disrupts critical Internet services, resulting in substantial financial losses.

Fig. 2 shows a general comparison of attacks in different network environments: Internet of Things (IoT), Software Defined Networks (SDN), Traditional Networks, and Cloud Computing. These environments were selected because they are the most common in the Internet [33–36]. In addition, it shows the used criteria and the main vulnerabilities, as following described.

IoT: Devices are often easy targets for DDoS attacks due to poor security and lack of regular updates, making them ideal for botnets (e.g., Mirai) [37,38]. These devices are distributed widely, which increases the attack surface and makes defense more difficult [39,40]. The main difficulty is that many IoT devices were not designed with robust security, making them easily exploited by malware[1] [42–45].

[1] Malware poses a threat by being specifically designed to evade detection mechanisms, spreading over long periods, collecting sensitive information, or positioning itself for a high-impact attack [41].

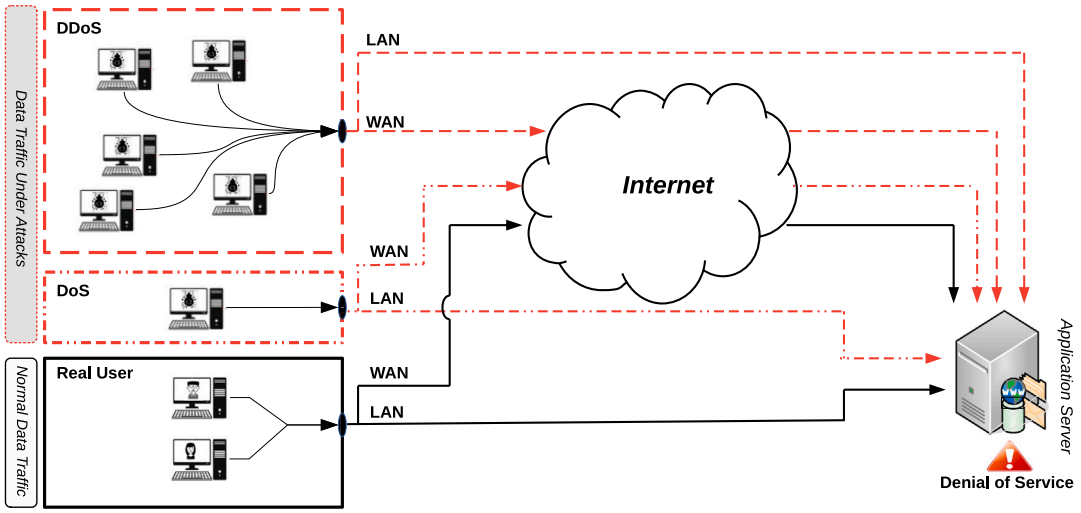


Fig. 1. DoS attack and its variant DDoS.

General comparison of DDoS attacks

Criterion	IoT	SDN	Traditional	Cloud
Main target	Connected devices with low security	Central network controller	Servers, Routers, Firewalls	Hosted services, cloud infrastructure
Common Attack Vectors	Limited capacity, lack of monitoring and large distribution of devices	Flow table flooding, Controller Attacks	SYN Flood, DNS Amplification	HTTP Flood, Amplification attacks, Resource Exhaustion and APIs
Vulnerabilities	Lack of security and processing capacity	Centralization in the controller	Unsafe protocols, volumetric traffic	Dependence on scalability and resource usage
Impact	Massive outage of devices and services	Complete network downtime	Service degradation or interruption	Widespread impact across multiple cloud customers
Mitigation Challenges	Difficulty updating and controlling devices	Controller protection and control plane	Firewalls, IDS, Load balance	Widespread impact across multiple cloud customers

Fig. 2. General comparison of DDoS attacks.

SDN: Architecture separates the control plane from the data plane. As a result, SDN controllers can be vulnerable points in a DDoS attack [36]. An attack targeting the controller has the potential to bring down the entire SDN network because the controller is responsible for managing traffic and routing policies. While SDNs offer dynamic mitigation of DDoS attacks, the centralization of control also creates a vulnerability. Therefore, protecting the SDN controller and implementing logical network segmentation is essential for defending against DDoS attacks [46–48].

Traditional Networks: In conventional networks, DDoS attacks often target servers and critical network devices, such as routers and switches, responsible for maintaining connectivity between systems and services. Volumetric attacks can overload these devices, resulting in significant network outages. The DNS (Domain Name System) amplification attack is an example that exploits DNS servers to amplify traffic directed to the victim. Volumetric attacks, such as User Datagram Protocol (UDP) Flood and DNS amplification, are characterized by sending massive amounts of traffic to the target to exhaust bandwidth or system resources. Protocols that lack security features are more susceptible to DDoS attacks. Although traditional networks have robust defense systems, such as firewalls, IDS (Intrusion Detection System), and IPS (Intrusion Prevention System), their limited physical infrastructure and rigid architecture make it difficult to adapt to large-scale attacks. Recovery can be slow depending on the size of the infrastructure, especially in environments with limited redundancy [49–51].

Cloud: In these environments, dealing with DDoS attacks can be more challenging due to the scalability of the infrastructure. However,

there are advanced solutions like Content Delivery Networks (CDN) and global load balancing that allow you to help distribute malicious traffic. While cloud infrastructure is able to scale to handle sudden increases in traffic, well-coordinated attacks exploit this scalability, leading to high costs and resource exhaustion. In order to mitigate these attacks, securing Application Programming Interface (API) and implementing Web Application Firewalls (WAF) are crucial [52–55].

Fig. 2 presents an overview of vulnerabilities and challenges that different types of networks (IoT, SDN, Traditional, and Cloud) face under DDoS attacks. Each network and its devices have their own specificities determining the most frequent type of attack, as well as the appropriate defense measures. Therefore, understanding these definitions, it is possible to improve the adoption security strategies with greater efficiency and effectiveness to ensure that protection solutions are aligned with the infrastructure of each network environment.

DDoS attacks have evolved over time, driven by increased Internet connectivity and usage. This evolution includes the exploitation of multiple compromised systems, which are used in a coordinated manner to overload a target, such as a server. One of the main mechanisms used in this strategy is the botnet, which consists of a network of devices infected by malware (known as "zombies") controlled remotely by a botmaster [56]. The botmaster commands these compromised devices, directing them to carry out malicious activities through specific commands. Botnets comprise a wide range of devices, including computers, servers, and various IoT devices such as IP cameras, DVRs, Android applications, routers, and printers [57–60]. This diversity makes botnets a powerful and versatile tool for carrying out DDoS attacks, complicating the detection and mitigation of these threats.

Bots are an automated software program that performs specific tasks without human intervention [50]. These programs are designed to operate independently, carrying out activities that range from benign, such as serving as virtual assistants, to malicious, such as those performed by bots in a botnet [61,62]. As a result, McDermott et al. [63] categorize benign bots as useful and necessary, performing tasks such as responding to commands or indexing web pages. On the other hand, malicious bots are used for illegal and fraudulent activities, often involved in DDoS attacks [43,64].

The investigation of botnets in all malicious activities has been gaining attention, given the potential that a distributed and coordinated attack has, which naturally makes the use of botnets increasingly explored for the practice of cybercrimes [48,65]. The increase in the number of criminal actions and technological advances have led to more efficient attacks, such as DDoS [63]. DDoS attacks use botnets, networks of compromised devices (bots) controlled by an attacker [37]. Through these botnets, attackers can coordinate massive attacks against specific targets, overloading them with malicious traffic. Thus, the DDoS attack process through botnets is divided into phases [37]:

- **Recruitment:** the attacker scans vulnerable hosts (referred to as agents or bots) that will be utilized to carry out the attack. Initially, this process was manual, but over time, it has become automated, and currently, several scanning tools can be used for this purpose;
- **Exploitation and Infection:** vulnerable devices are infected and become part of the botnet;
- **Communication:** the attacker communicates with the botnet to identify which bots are active and program attacks;
- **Attack Execution:** the bots begin sending malicious packets to the target, with parameters adjusted as necessary to succeed in the denial of service attack.

Botnets use different protocols to carry out DDoS attacks. This occurs because they aim to exploit diverse vulnerabilities and maximize the impact of their attacks [66,67]. By leveraging different protocols, botnets can overwhelm network resources in distinct ways, making it challenging for the target's defenses to mitigate the attacks [37]. This complexity makes the attacks harder to detect and block, as each protocol can be manipulated to generate different types of malicious traffic and exploit specific weaknesses in target systems, thereby increasing the likelihood of a successful attack [50]. The main protocols used for these attacks include:

- **Internet Relay Chat (IRC):** One of the main protocols used for communication between botnets due to its easy-to-implement structure, although it is rarely used in corporate networks and can be easily blocked by firewalls;
- **Hypertext Transfer Protocol (HTTP):** Used in a client–server model where the client makes a request and the server responds. It is simple to implement but has higher latency compared to IRC;
- **Server Message Block (SMB):** A protocol designed to share resources such as printers and files between computers, used by attackers for communication on local networks;
- **Peer-to-Peer protocols (P2P):** Used to maintain communication between bots, allowing devices to share different files and reduce download times;
- **Simple Mail Transfer Protocol (SMTP):** Used to send spam or phishing emails from compromised machines;
- **Proprietary protocols:** Attackers can create specific protocols for communicating with bots, typically based on TCP or UDP and in some cases using the Internet Control Message Protocol (ICMP) for command and control (C&C) communication.

2.1.1. Challenges in DDoS detection

The identification of DDoS attacks faces several challenges, as pointed out in the literature [68]. The diversity and complexity of the methods employed by attackers make identifying attack patterns a difficult task, as they often adapt their strategies to evade detection, complicating the accurate analysis of these events [1]. Furthermore, the real-time nature of attacks requires fast and accurate responses, as the speed at which attacks are launched and the traffic generated is significant [69,70]. This need requires the creation of detection systems capable of operating at high speed and precision to respond effectively [47,48].

Another relevant challenge is scalability in large network environments, where the volume of data to be analyzed makes the distinction between legitimate and malicious traffic complex [71]. The situation worsens when DDoS attacks imitate the behavior of legitimate users, especially in application layer attacks, making identification difficult by security systems [72]. Furthermore, the lack of real-time data generated from DDoS attacks provides a limitation in the analysis and development of new models, with the collection and analysis of data being extremely important to understand attack patterns and consequently the creation of defenses [55,73]. These difficulties reinforce the importance of continuous research and development to propose new approaches to improve DDoS attack detection.

Over the years, DDoS attacks have become increasingly complex and sophisticated; this includes the techniques used, artificial intelligence tools, and even the growth of the Internet, contributing to increasing the complexity of this type of attack [71] . In this way, several challenges have increased to detect attacks, such as diversity of attack vectors, packet flooding, protocol attacks, and application attacks, among other types of attacks that are part of the list of DDoS attacks [70,74]. Furthermore, the increase in connected IoT devices contributes to attackers providing large-scale actions, making detecting attacks even more challenging.

According to these complexities, the need for quick and effective responses to propose a safer network infrastructure for this type of attack is evident. Therefore, implementing traffic filtering techniques like those in Access Control Lists (ACL) and firewall implementations should help prevent attacks [75]. In addition to these tasks, other services, such as DDoS mitigation and Content Distribution Networks

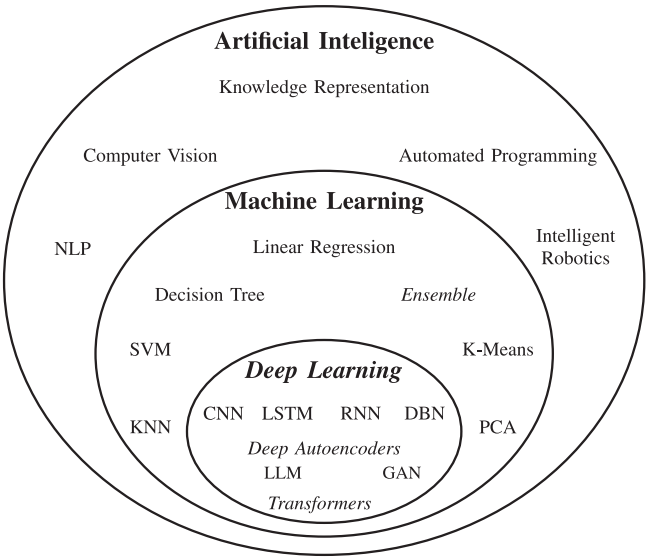


Fig. 3. Artificial intelligence and subfields.
Source: Adapted from [84].

(CDNs), help contain and reduce server attacks. Additionally, continuous monitoring strategies and automated real-time responses help quickly identify traffic anomalies, minimizing the impact of attacks and ensuring more effective defense. These approaches are key to addressing the growing threat of DDoS attacks, especially in IoT environments where protection against disruptions is critical.

2.2. Learning models

Artificial Intelligence (AI) encompasses Machine Learning (ML), with Deep Learning (DL) being one of its subfields, where in ML, algorithms learn to perform tasks based on the data provided [76]. In machine learning techniques, humans and models perform feature extraction [77,78]. Initially, humans select and configure the relevant features to train the model. However, many advanced machine learning algorithms autonomously extract features directly from the data, identifying patterns and relationships without requiring detailed manual configuration. While humans set the initial parameters, models take over feature extraction during training and analysis. The mathematical model begins by extracting features from the data and then classifies new data based on the identified patterns [79,80].

Various algorithms can build models based on training data through this dynamic, allowing decisions or decisions on new data without explicit instructions. Applications span multiple domains, particularly cybersecurity, which predict [67], detect [73], and discover new and automated attacks [81]. One of the classic definitions of AI is the ability of machines and computer systems to perform tasks that require human-like intelligence, such as reasoning, learning, and decision-making [82,83]. In this way, ML is a subarea that focuses on developing algorithms and models that allow machines to learn from data [84]. ML techniques are used instead of being programmed to perform a specific task [85]. As a result, ML can extract features, identify patterns, and make predictions based on prior information [9]. This approach has proven effective in several applications, from voice recognition, images, medical diagnoses, and even the detection of attacks such as DDoS [86].

Fig. 3 shows the hierarchy and interrelationships between the main fields of AI, emphasizing their specific subareas and techniques. AI is the broadest category and is divided into ML and DL. ML includes methods like KNN, SVM, and Decision Trees. DL covers advanced techniques such as CNN, LSTM, RNN, GAN, Transformer, and LLM models [21,87]. Each subfield represents a set of approaches and models applied to areas like CV, NLP, KR, AP, and IR. The figure demonstrates how these concepts interconnect, forming a comprehensive and interdependent structure in AI system development [88,89].

The paper proposed by [80] highlights the importance of ML techniques in detecting DDoS attacks. This paper discusses the need for security measures to deal with threats that contain increasingly sophisticated AI resources. The paper thoroughly analyzes ML applications in security, covering areas such as intrusion detection, malware identification, spam filtering, DDoS attacks, and types of connected devices from computers to mobile devices.

According to the paper proposed by [90], ML has different learning approaches, such as supervised, unsupervised, semi-supervised, and reinforcement learning. By default, supervised learning papers with labeled data to train your algorithms. This action allows the learning process to make predictions or classifications based on new training data. Unsupervised learning has the characteristic of working with unlabeled data to identify patterns or groupings without the need for external supervision. Semi-supervised learning can combine elements of the two types of previous learning (supervised and unsupervised) through a limited amount of labeled data and a limited amount of unlabeled data, especially when there is a computational cost in the data labeling time. Finally, reinforcement learning is the approach in which the agent learns to make decisions through interactions with the environment, through rewards or punishments based on the actions taken, always aiming for performance over time [91]. For this reason, these models are explored in several areas, including security for detecting DDoS attacks [68,89,92,93].

The paper proposed by [94] presents the Deep Reinforcement Learning (DRL) technique. This ML technique combines deep learning with reinforcement learning, enabling agents to make optimal decisions through interactions with the environment. In this way, agents in DRL learn when they receive rewards or penalties, so they must continually adjust their strategies. This approach efficiently applies complex problems in dynamic environments such as DDoS attacks. DRL-based systems have the ability to automatically detect and respond to threats such as DDoS, malware, and data injections, which are also applied in real-time. This ability to learn and adapt makes DRL an important tool capable of analyzing data in networks containing different types of attacks. Table 2 presents the datasets, algorithms, and various applications to detect DDoS attacks according to recent papers.

2.2.1. Machine learning applied to detect DDoS attacks

Researchers implement various ML and DL techniques to detect DDoS attacks. Analyzing recent papers [99] exploring these approaches is essential to highlight the challenges faced and proposed solutions. These papers [100–102] improve the understanding of current strategies and their limitations, driving the development of effective and adaptable methods to detect DDoS attacks across scenarios.

Detecting DDoS attacks is a significant challenge in modern networks, especially with the exponential growth of IoT devices and emerging networks like SDN networks. Researchers actively investigate ML and DL techniques in this scenario, highlighting advances and challenges in identifying malicious traffic patterns and DDoS attacks. IoT networks have particularities that increase their vulnerability to attacks, such as the heterogeneity of devices and resource limitations. Almaraz et al. [88] explore using the Bot-IoT dataset for DDoS attack detection, using an approach that combines ML models, such as DT and MLP, with DL techniques. The approach stands out for treating class imbalance and its high accuracy, which is more significant than 99%. Despite this, the exclusive dependence on Bot-IoT limits the application in more diverse IoT scenarios, highlighting the need for datasets representing many real situations.

In the domain of SDN networks, where flexibility and centralization of management bring benefits and risks, [103] propose an architecture to detect DDoS attacks based on statistical methods and classification algorithms. The Floodlight controller's collection of network flows,

Table 2
Features and datasets utilized in studies to detect DDoS attacks.

Ref.	Year	Dataset	Algorithm used	Application
[76]	2021	Leading India Project dataset (22 features: TCP/UDP/ICMP)	SAE-MLP, CNN, LSTM, SVC, SOUND	Traffic classification (malicious/normal) in SDN
[88]	2022	Bot-IoT dataset (36,340 samples)	RF, SVM, KNN, DT, LR	Detecting DDoS attacks in IoT using ML
[92]	2020	Custom dataset + NSL-KDD, CAIDA 2007, LongTail	SVM, KNN, ANN, NB	DDoS detection in SDN
[93]	2020	KDD'99 dataset (42 features, 494,021 instances)	LDA, CART, RF	Fog computing, IoT, 5G security
[95]	2021	CICDoS2017, CICDDoS2019 (SYN, UDP, DrDNS attacks)	SVM, RF, K-NN, MLP, CNN, GRU, LSTM, NN	SDN-based detection with ONOS controller and IDS
[96]	2022	CIC-IDS2017, USTC-TFC2016 (PCAP-based)	RF, CNN, ViT	ML-based network traffic classification
[97]	2023	WSN-DS (Blackhole, Grayhole, Flooding, TDMA), NSL-KDD	DNN, CNN, RNN	DoS attack detection in WSN using DL
[98]	2023	UNSW NB15, KDD Cup99, NSL-KDD, Bot-IoT, CCD-INID-V1, TON_IoT	RF, NB, SVM, LR, K-NN, LDA, CART, LSTM	IoT network attack detection with ML models

and the use of algorithms such as BayesNet and RandomTree enable the identification of massive and subtle attacks. This paper highlights the network topology's independence, but the model's computational complexity can make it challenging to scale in large environments. Another relevant study in the SDN context is that by [92], which uses feature selection techniques to optimize the performance of classification models such as KNN and SVM. Applying these techniques results in an accuracy of 98.3%, but the dependence on a topology based on a single controller represents a limitation for more complex and distributed scenarios.

One of the innovations in recent years is the integration of ML techniques into vehicular communication networks, which has gained significant visibility according to [104]. The authors proposed a misbehavior detection system (MDS) that uses ML techniques to identify various attacks, including DDoS attacks that compromise vehicle communication. This technique used a combination of supervised ML models, such as RF and SVM, distinguishing between normal and malicious activities. This technique improves accuracy and detection and addresses the challenges vehicular networks pose.

Furthermore, Guastalla et al. [105] introduce pre-trained language models in detecting DDoS attacks, addressing both aggressive and subtle attacks. Using the CICIDS 2017 and Urban IoT Dataset datasets, the authors highlight attribute reduction to increase training efficiency, achieving accuracy rates of up to 96%. However, addressing the impact of a limited context and the need for well-defined and customized pre-processing is vitally essential for the proposed approach to be successful. The papers provide an overview of efforts to combine ML and DL techniques to detect DDoS attacks, considering different scenarios and constraints. At the same time, they point out open issues, such as model generalization and computational requirements, and emphasize the need to improve security in detecting DDoS attacks.

2.2.2. Deep learning applied to detect DDoS attacks

Deep Learning (DL) is an AI technique that learns from computer data to perform tasks similarly to humans through complex and hierarchical data representations, extracting important patterns and features from a dense set of data [18,106]. Hence, DL addresses complex, dynamic, and high-dimensional problems, making it a powerful and relevant tool for cybersecurity [20,95]. Another contribution of these techniques is the integration of DL with Reinforcement Learning (DRL), making it highly capable of solving complex cyber-defense problems, dynamically adapting to constantly evolving threats [94]. Then, EAD plays a fundamental role in DRLs addressing complex cybersecurity challenges, enabling adaptive and practical systems to protect digital assets against malicious threats [63,94].

Nguyen et al. [19] conducted a study where researchers utilized Deep Learning to detect malicious DNS-over-HTTPS tunneling attacks in corporate networks. The proposed system employed the Transformer architecture as a classifier to identify malicious DoH traffic. Researchers chose the Transformer model for its capability to process sequential data in parallel and automatically select features during the training phase. In the paper proposed by [107], DL was applied to detect attacks on cyber–physical systems. A study reviewed by seniors from 2017 to 2021 proposed the development of a methodology built in six stages. Using this methodology, it was possible to obtain an overview of advanced solutions for detecting cyber attacks on cyber–physical systems (CPS). In addition, existing challenges were identified, and future directions for security were explored. These contributions were guided by the evident potential of DL for several areas, including CPS security. Finally, the paper proposed a set of interesting insights to address computational security on different fronts, including denial of service attacks.

2.2.3. CNNs model

Convolutional Neural Networks (CNNs) process input data through several layers, including convolutional, activation, pooling, and fully connected layers [108]. These networks begin with raw input, often an image, and apply filters that slide across the data to extract features such as edges, textures, and patterns. Activation functions like ReLU introduce non-linearity while pooling layers reduce dimensionality and computational load by summarizing local regions [109]. Finally, fully connected layers interpret the extracted features and produce output probabilities, typically using a softmax function for classification. During training, the network optimizes its filters and weights through backpropagation to minimize prediction error.

CNNs stand out in deep learning for their ability to automatically learn relevant features from raw data without manual extraction. Their use of shared weights reduces the number of parameters, making models more efficient and less prone to overfitting [44]. CNNs also provide translational invariance, allowing them to recognize objects regardless of slight changes in position. Their hierarchical structure enables deep layers to learn increasingly complex patterns, resulting in outstanding performance in visual tasks such as image classification, object detection, and image segmentation. These strengths make CNNs a leading choice for handling image-based and spatial data applications.

CNNs play a central role in deep learning because they efficiently process structured and visual data. As a result, researchers have applied CNNs in diverse domains, from real-time image classification and object detection using architectures such as YOLO and Faster R-CNN to network security applications such as DDoS attack detection [108]. When it comes to the application of CNNs in DDoS attack

detection, this technique significantly improves the performance of DDoS attack detection in SDN, for example. In this case, the paper proposed by [110] presented the proposal of employing an ensemble-based framework, where the proposed approach combines multiple CNN models to capture diverse network traffic features, leading to more reliable and consistent detection results. The system effectively identifies anomalous patterns associated with DDoS attacks, even under varying traffic conditions, and maintains strong detection capabilities across evaluation metrics. The CNN ensemble offers improved accuracy and robustness while maintaining computational efficiency compared to existing deep learning methods. These results highlight the model's potential to strengthen network security through accurate and scalable threat detection [111].

2.2.4. LSTM model

Long Short-Term Memory Networks (LSTMs) represent an advanced form of Recurrent Neural Networks (RNNs) designed to overcome the limitations of traditional RNNs in capturing long-term dependencies in sequential data [108]. Introduced in 1997, LSTMs have gained popularity in the deep learning community due to their ability to retain and manage information in extended sequences [112]. Its architecture relies on internal memory cells and three main gates (input, forget, and output) that regulate the flow of information and allow the network to preserve relevant context over time. This makes LSTMs particularly effective in tasks involving time-dependent data, such as natural language processing, speech recognition, and time series prediction. Thus, recent advances in LSTM implementations using memristor-based circuits are highlighted, which aim to accelerate computation and improve energy efficiency in sequence modeling [113].

Implementing DDoS attack detection using a hybrid model that combines Deep Belief Networks (DBNs) for feature extraction with an LSTM network optimized by Paper Swarm Optimization (PSO) [114]. DBN extracts key features from IP packet data, condensing the information and making it easier for LSTM to analyze patterns linked to DDoS activity. LSTM, known for retaining long-term dependencies, models network traffic flow over time. PSO adjusts LSTM weights to improve classification accuracy reducing prediction errors when identifying anomalous traffic. Evaluated on the NSL-KDD dataset, this approach outperformed traditional classifiers across several performance metrics [115].

LSTM contributes significantly to DDoS detection by learning and predicting network behavior based on temporal patterns. It captures abrupt or unusual changes in traffic, which are often early indicators of DDoS attacks [116]. Additionally, the system supports collaborative detection, allowing LSTM models across distributed locations to share results, improving detection accuracy even in isolated environments. It dynamically adjusts detection thresholds based on aggregated insights, reducing false positives as traffic patterns change. LSTM's ability to handle non-linear and irregular traffic (such as jitter or sudden spikes) makes it particularly effective for protecting real-time and cloud network systems against sophisticated DDoS threats.

2.2.5. Transformer model

Transformer is a Sequence to Sequence model based on Attention Mechanisms [11]. Recurrent layers commonly used in Encoder–Decoder architectures are remanded by multi-head self-attention. The Transformer model adopts a self-attention mechanism, completely replacing the traditional structures of RNN and CNN. This approach, widely used in NLP tasks, consistently achieves excellent results [52]. Applications of attention in the Transformer model, such as Encoder–Decoder attention and self-attention, are discussed, along with plans to apply attention-based models to other tasks and input/output modalities. Additionally, it is possible to revamp mathematics in parallel processing systems to achieve faster execution [11].

In [117], the authors introduced a hierarchical CNN-Transformer architecture that significantly enhances anomaly detection, including DDoS attacks, by leveraging both spatial and temporal features of network traffic. The approach highlights the growing importance of Transformer models in detecting complex attack patterns, as they capture the intricate dependencies within traffic data, making them invaluable for protecting networks against evolving cyber threats.

In [118], a Transformer-based model for DDoS attack detection is presented. The preprocessing applied in the proposed model transforms flow data into optimal sequences for analysis. In this way, the transformer's attention mechanism can identify confusing and diverse relationships between several flows, which facilitates the detection of anomalous patterns indicating DDoS attacks. Finally, the customized Transformer architecture reduced the dimensionality of the data, thus increasing efficiency and preserving accuracy in classifying anomalies for real-time detection.

The paper presented by Alrahmani et al. [74] evaluated two Transformer models, the Frequency Enhanced Decomposed Transformer (FEDformer) and the Patch Time Series Transformer (PatchTST), to predict DDoS attacks. Their analysis shows that PatchTST outperforms FEDformer in DDoS attack detection, emphasizing the effectiveness of adapting Transformer models for better detection and response.

The paper published by the Forbes website [8] highlights the Transformer as a significant trend in AI, along with techniques such as Self-Supervised Learning and Federated Learning, the latter of which was formally introduced by Google researchers in 2017. However, the Transformer model has gained recognition as one of the most influential innovations in artificial intelligence beyond these techniques. According to the paper, ‘‘Attention is all you need’’, this model could potentially drive a paradigm shift in the field [11]. The Transformer architecture was first applied to solve sequence-to-sequence problems. Researchers enhance these blocks by incorporating additional strategies such as positional encoders and attention units [119]. The architecture uses parallel attention queries to compute interrelations between data elements [120]. Designers introduced attention mechanisms as a key component to solving sequence-to-sequence problems [121]. Today, Transformer models support a variety of tasks [122]. Thus, this fundamental architecture serves as the basis for prominent models such as BERT [106,123], GPT-2 [124], and GPT-3 [125].

2.2.6. Transformer architecture

Fig. 4 presents the Transformer Architecture, and this section offers a detailed description of all the layers and components that make it up. Prior knowledge about the architecture and its structure, in addition to the layers, is essential to understand the purpose and efficiency with which Transformer can be applied and the customization possibilities for different areas of knowledge. As pointed out by [64], since the original Transformer proposal in 2017, several models have been developed to address a variety of tasks in different fields. While some models maintain the Vanilla architecture, others have created variants depending on the structure utilized.

This division organizes the functioning and scalability of the Transformer model. The Transformer Layer represents the internal layers of the model, which are responsible for processing input through self-attention and feedforward neural networks. These elements form the basis of the model, consisting of Self-Attention Mechanism, Feedforward Networks, Layer Normalization & Dropout, and Multi-Head Attention. Depending on complexity, these layers determine the model's depth, which can have 12, 24, 48, or more layers. The more layers, the better the learning capacity for complex patterns and the higher the computational cost. The Layers and Areas of a Transformer Architecture follow the terms below.

2.2.7. Transformer layers

The descriptions and identifications of the layers and areas of the transformer were taken from various authors, such as: [11,120,123,126] and thus, they are described below:

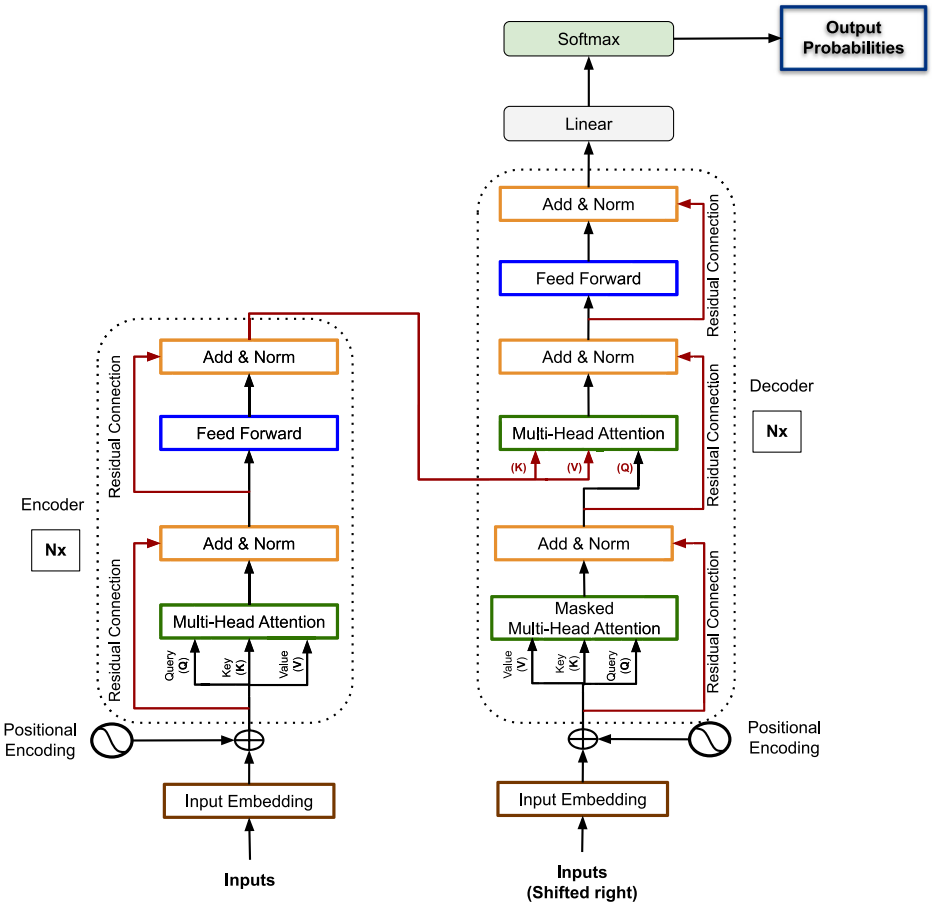


Fig. 4. Transformer architecture. *Source:* Image from the original paper [11].

(01) — Embedding Layer: In the Input Embedding, the input words or symbols are converted into a dense vector representation. These embeddings are learned during model training. The Transformer model proposed by Vaswani et al. [11] starts with Input embedding, a sentence divided into words mapped into numbers. These numbers represent the position of the words in the vocabulary of words. The so-called Tokens are words a word that are identified by their position in the vocabulary through so-called input IDs and that weigh the vector of fixed size. In this case, following the original Transformer values, it is 512, as it is a parameter for the model that will learn to change these numbers to represent the word's meaning. However, it is essential to emphasize that the entry IDs never change, as the vocabularies are fixed. Meanwhile, embeddings will change along with the model training process. This way, the numbers based on the embedding will change based on the requirements of the loss functions. Therefore, the model = 512 is the input embedding that maps the unique words into an embedding of 512 and is called mode. Positional encoding is a sinusoid (sine Eq. (1) and cosine Eq. (2)) used to encode the order, and then a new image of a defined size (in this case, 512) is added to the original image.

The equations for positional encoding in Transformer, according to Han et al. [127], is defined as:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10\,000^{\frac{2i}{d_{model}}}}\right) \tag{1}$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10\,000^{\frac{2i}{d_{model}}}}\right) \tag{2}$$

These equations enable the self-attention layer to capture positional information from words in a sentence. Since sentences in a language follow a sequential order, the model must incorporate this information into the encoding. The model adds positional encoding with dimension d_{model} to the original input using these equations to achieve this. In this process, "pos" represents the tokens of positions in the sequence; "i" indicates the dimension index, and d_{model} defines the size of the embedding vector. Each element of the positional encoding follows a pattern of sine and cosine functions, enabling the model to learn relative positional relationships. This approach enhances the model's ability to interpret ordered sequences and improves its efficiency in generalizing to longer sequences during inference [127]. The effect of this task is to embed in the vector that represents each word of information related to the order being represented with the change in frequency.

(02) — Positional Encoding: This layer represents positional encodings added to input embeddings to provide information about the position of each token in the sequence, helping the model capture word order. The function of positional encoding is to provide information about the relative or absolute position of tokens in the input sequence. As the Transformer does not have recurrence or convolution mechanisms to capture the sequential order of the tokens, positional encoding is introduced to consider the position of the tokens during processing. The positional encoding is added to the input embeddings before they are provided to the Transformer layers. It provides a numerical representation that encodes the position of each token in the sequence, allowing the model to distinguish tokens based on their relative position in the input. This is crucial to ensure that the

model captures information about token order and learns sequential dependencies during sequence processing.

(a) — Encoder: — The Encoder is responsible for processing the input sequence. It is formed by a stack of identical blocks (L) processing inputs and capturing context representations. The Encoder transforms the input sequence into an internal representation that understands the relationships between the sequence elements. It uses self-attention layers to model dependencies between words, regardless of their position in the sequence.

The roles of Query (Q), Key (K), and Value (V) are very important, as they are three copies of the same thing, which are made before entering the attention module. At this point, it helps parallel processing in the attention mechanism as input to the multi-head through these word vectors with the order information added. From this moment on, it enters the attention mechanism, in this case, Multi-head attention, where the idea is to represent a single word by several vectors. In the Transformer Encoder, the concepts of Query (Q), Key (K), and Value (V) are fundamental for the functioning of the attention mechanism. Therefore, the details of how this trident papers in the TA layers follow the following detailed steps:

- Generation of Q, K, and V: In Encoder, inputs are sequences of words (or tokens) converted into vectors through embeddings. Then, each of these input vectors is used to generate three different representations: the Query (Q), Key (K), and Value (V) vectors. This is done through multiplications by matrices of different weights. Thus, although Q, K, and V are derived from the same input (i.e., from the exact word representation), each vector has a specific function within the attention mechanism.

- Attention Mechanism: The Encoder's attention mechanism is base-Transformerd on scalar product attention, where the Queries (Q) of the inputs interact with the Keys (K). The operation involves calculating the similarity (generally in the form of a scalar product) between the Queries and the Keys. These products are normalized using a softmax function to create an attention distribution, which indicates the importance of each input about the current Query. This distribution is then used to weight the Value Vectors (V), producing an output vector that integrates contextual information from the whole input sequence.

- Multi-Head Attention: Multi-head attention enables the model to extract different representations of attention subspaces in parallel. This means that for each attention head, various sets of Q, K, and V are calculated, allowing the model to learn to focus on other parts of the input simultaneously. After processing in multiple heads, the results are concatenated and passed through a linear layer, resulting in a rich and diverse combination of input information.

- Propagation and Normalization: The attention output is then added to the original input of the Encoder block through a residual connection, and this sum is normalized using a normalization layer (Layer Norm). This normalization helps stabilize learning by preserving important information from the original input.

- Repetition in Blocks: The described process occurs in several Encoder blocks, where each block applies its own set of weights and, potentially, learns different aspects of the same input sequence. The result is a representation that captures long-range dependencies between words in the input sequence.

In this way, while the Encoder attention mechanism uses the same inputs to generate Q, K, and V, it has an internal working that allows it to model complex relationships within the sequence very efficiently, preparing these representations for the next phase of the architecture.

(b) — Decoder: — generates the output sequence based on the Encoder representation. Like the Encoder, it is also composed of a stack of identical blocks. Still, it includes additional mechanisms such as cross-attention, which allows the Decoder to focus on the representations generated by the Encoder when producing outputs. The Decoder generates output one word at a time, using the previous word and input information to predict the next word in the sequence.

In the Transformer Decoder, the concept of Query (Q), Key (K) and Value (V) is also applied, but there are some important differences compared to the Encoder. Here is how the trident occurs in Decoder:

- Generation of Q, K and V: Just like the Encoder, the Decoder generates the Query, Key, and Value vectors from the inputs. However, the inputs in the Decoder are different. The Decoder utilizes the words that were already generated in the output sequence as input, in addition to the representation of the input sequence that comes from the Encoder. For each word that is generated, the Decoder creates a Query vector, while the Key and Value vectors are derived from the words that were previously generated and from the Encoder output.

- Attention Mechanism: In the Decoder attention layer, in addition to auto-regressive attention, which allows the Decoder to focus on previously generated words (using Q, K and V coming from the Decoder's own outputs), there is also cross-attention. Cross-attention involves the Key and Value vectors that come from the Encoder, allowing the Decoder to use contextual information from the input sequence when generating the output. This is crucial for coherent text generation, as it allows the Decoder to "pay attention" to the most relevant parts of the input string.

- Masked self-attention: Decoder also incorporates a masked self-attention mechanism, which prevents the prediction of a word from being based on future information in the sequence. In other words, when generating the $i-$th word, the Decoder cannot "see" the (i+1)th word. This is done by masking the attention vectors, ensuring that each word can only be related to the previous words.

- Output Production: The output of the attention mechanism (both self-attention and cross-attention) is then passed through a feed-forward network before being combined with the original input from the Decoder layer through residual connections and layer normalization.

This process allows the Decoder to generate output sequences efficiently, using both the context of the previously generated words and the input sequence information processed by the Encoder.

(RC) — Residual Connections: The following action is the arrow that leaves before the trident and goes to the Add & Norm (Skip-type Residual Connection). This action of the Skip connection appears several times throughout all layers of the Transformer architecture. It means that at this moment, a sum of two matrices is made; after this sum occurs and a normalization, this is where Add & Norm comes in because it has the possibility of retro propagation, that is, the original vectors are combined with new vectors, which contain the previous representations with the new representations. The residual connections and layer normalization are essential components of the Transformer. Residual connections allow the model to obtain information directly from previous layers, facilitating the flow of the gradient and improving learning capacity. Layer normalization normalizes the activation values of each layer, helping with model stability and performance.

(03) — Multi-Head Self-Attention: The multi-head attention mechanism allows the model to learn diverse token representations by processing multiple attention heads in parallel. Each head captures distinct contextual relationships between tokens, allowing the model to understand global sequence dependencies. The outputs from these heads are concatenated and projected back into the original space, ensuring an efficient and comprehensive representation. In the Encoder, this layer allows each token to interact with all other tokens in the input sequence, providing a complete understanding of the global context. This ability is essential for accurately interpreting information. The Decoder, on the other hand, applies attention differently through masked self-attention and cross-attention. Masked self-attention prevents a token from accessing future tokens, essential for text generation.

Cross-attention connects the Decoder to the Encoder, allowing the Decoder to use the Encoder's output when generating responses. This interaction ensures that the model produces contextually accurate and relevant outputs. When used in this way, multi-head attention is also called self-attention. This mechanism generates multiple vectors to

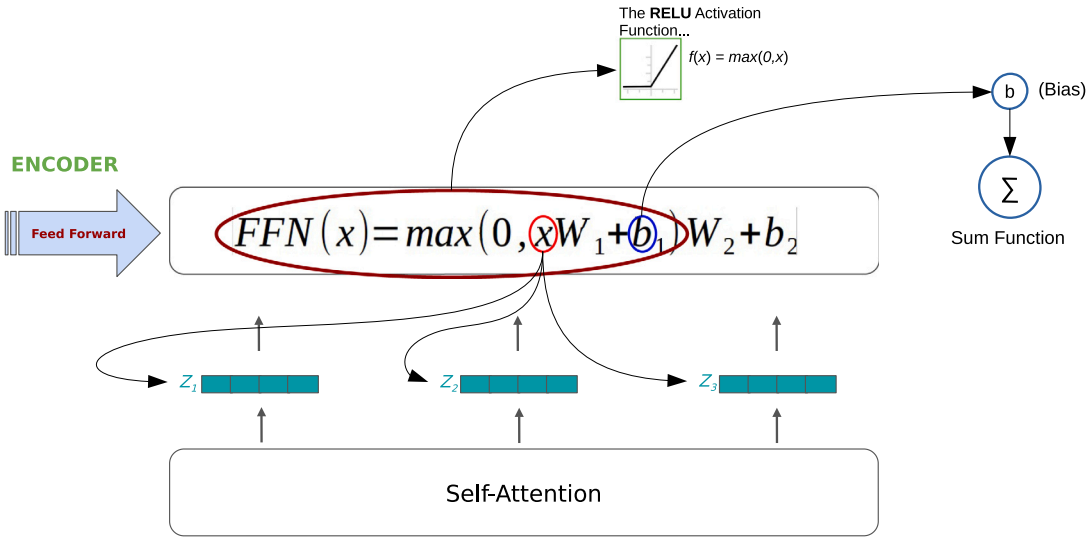


Fig. 5. Feed-Forward equation [11].

represent each word, with the original implementation using eight different representations per word. Since words have different meanings depending on the context, this multi-head approach allows the model to capture these nuances. Each attention head processes the input independently, and the model restores the original word dimensions by concatenation and projection. Although the attention mechanism transforms the word representations, the size of the input vectors remains unchanged, preserving the structure needed for further processing.

(04) — Add & Norm Residual Connections + Layer Normalization: This step is applied after each sub-layer (attention and fully connected network) to stabilize the training. **- In the Encoder:** Each Encoder layer comprises two main sub-layers, a multi-head self-attention layer, and a feed-forward network layer. After each of these sub-layers, an addition and normalization step occurs. This step consists of adding the output of the sub-layer (the attention layer or the fully connected network) to the original input of that sub-layer and then applying normalization to stabilize the training. Using multiple layers of addition and normalization in the Encoder allows the model to learn progressively more complex and abstract representations of the input sequence as it passes through each layer. **- In the Decoder:** Analogous to the Encoder, each Decoder layer also has two main sub-layers, a Multi-Head Self-Attention layer (in the first sub-layer) and a Multi-Head Attention layer over the Encoder output (in the second sub-layer), followed by a Feed-Forward Network. After each sub-layer, there is an addition and normalization step. The addition and normalization step in the Decoder plays a similar role to that in the Encoder, allowing the output of each sub-layer to be added to the original input of that sub-layer and normalized to facilitate training.

(05) — Feed-Forward Network (FFN): The Feed-Forward layer, shown in Fig. 5, follows a standard Deep Learning architecture. This layer takes an input "x" represented as "Z_i" and multiplies it by a weight matrix xW_1. After adding a bias term "b_1", the result passes through the ReLU activation function. The output is then multiplied by another weight matrix W_2, adding a second bias term b_2. This layer uses fully connected networks or linear layers to model the representations, applying them after the attention mechanism in each sub-layer. Non-linear activation functions (usually ReLU) intersperse these layers to enhance their expressiveness. These networks are responsible for transforming the intermediate representations of the Transformer.

This process forms a two-layer Feed-Forward representation. It is a neural network, that is, a network with two convolutional layers with a kernel of size 1. This means that the input size is the same as the output. After that, other Skip connections appear after Feed Forward, and the Encoder block ends. At this point, the so-called "Nx" comes into action, which represents the number of times (in the case of the original paper by Vaswani et al. [11]) the block is repeated six times. Because the first block output is the second block input, and so on, until the sixth block is called the Encoder, and the second part is the Encoder.

(06) — Masked Multi-Head Attention: This layer plays a crucial role because it ensures that, during the generation of each token (e.g., a word or subword in a sentence) in the sequential output, the model cannot access future information that has not yet been generated. This is essential to maintain the model's auto-regressive property, where the generation of each token depends only on the previously generated tokens. The mask applied in Masked Multi-Head Attention hides future connections during attention calculation, ensuring that each position in the Decoder can only pay attention to the positions before it is in the output sequence. This mask is implemented within the attention layer to prevent positions in the Decoder from accessing information that has not yet been generated, preventing future information leakage. Therefore, the Masked Multi-Head Attention layer in the Decoder plays a fundamental role in the auto-regressive generation of sequences, ensuring that the Transformer model produces correct and coherent outputs, considering only the information available until the moment of generation of each token in the output sequence. **(07) — Linear:** The Linear layer carries out linear transformations on input data. This layer is essential for projecting the input data into a different dimensional representation space, allowing the model to learn more complex and abstract representations. The function of the Linear layer in the Transformer architecture encompasses two main operations: **- Linear Projection:** The input vectors are projected into a space of different dimensions. This is critical for adjusting the dimensionality of the input data to match the desired dimensionality for subsequent operations on the model. **- Linear Transformation:** In addition to projection, linear transformations are performed on the input data. These transformations allow complex patterns to be learned and non-linear relationships between different sequence parts to be captured.

(08) — Softmax: The Softmax layer's function is to calculate the probability distribution over the output vocabulary. This step is crucial for generating output sequences, like in machine translation, where the next word is predicted based on available information. The Softmax operation is applied to the output of the last Decoder layer to convert the scores into probabilities. These probabilities indicate the likelihood of each token in the output vocabulary being the next token in the generated sequence. The Softmax function normalizes the output scores so that the sum of all probabilities equals 1, allowing the model to select the most likely token as the next prediction.

(Nx) — Number of Layers in the Encoder and Decoder: The "N" in "Nx" in the Encoder and Decoder represents the number of layers in the Encoder and Decoder stack of layers. In the Encoder, "Nx" refers to the number of Encoder layers, that is, the number of identical layers stacked to process the sequential input. In the Decoder, "Nx" refers to the number of Decoder layers, which consists of a stack of identical layers to generate the sequential output based on the information from the Encoder and the previous outputs from the Decoder. Therefore, "Nx" indicates the number of repetitions of the Encoder and Decoder layers in the Transformer model, allowing the model to capture information at different levels of abstraction and complexity when processing and generating sequences.

(N) is used after the Residual Connection in each sublayer (Multi-Head Attention and Feed-Forward) in the Encoder and Decoder. This stabilizes the gradients and enhances training.

(c) Attention Mechanism: — In this section, the Self-Attention mechanism, present in both the encoder and decoder, is discussed. Cross-attention is used in the decoder to focus on the encoder. The functionality of the self-attention layer is analyzed to clarify its role in the Transformer architecture since this mechanism is fundamental in the structure of this model. Fig. 6 illustrates how the layers and attention mechanism operate, showing layer = 0 and Attention = All. When processing a sentence, the model focuses on the weights associated with the word detection, considering the representations of all words in the sentence. When processing a sentence, the model focuses on the weights associated with the word detection, considering the representations of all words.

The mechanism relies on a weighted sum, where more significant words receive higher weights. In the example, words like detection, security, DDoS, and attacks hold the maximum weight of 14, while less critical words receive lower weights or lose significance in the process. This weighting embodies the essence of Self-Attention: the model generates a unique representation for each word in the sentence, calculated and weighted according to its contextual importance. After passing through a self-attention sublayer, each word acquires a distinct representation. This representation captures information contextually, prioritizing words that contribute significantly to the meaning of each specific word in the sentence.

Transformer architectures incorporate the self-attention mechanism (Fig. 7), also known as scaled dot-product attention [120]. During training, the model employs three weight matrices as parameters: W_q (Query), W_w (Key), and W_v (Value). These matrices project the input embeddings x into the query (Q), key (K), and value (V) components of the sequence. To generate these components, the model multiplies Q, K, and V by their respective weight matrices.

Eqs. (3), (4) and (5) proposed by [128] show that the vectors $\mathcal{Q}^{(i)}$ and $\mathcal{K}^{(i)}$ are of dimension d_K. Note the index i as it refers to the index position of the token in the input sequence that has length T. Consequently, the projection matrices W_Q and W_K have the format d_K x d, and W_V has the format d_V x d. It is important to emphasize that d specifies the size of each word vector, x. When calculating the dot product between the query and key vectors, both vectors must have the same number of elements ($(d_Q = d_K)$). In contrast, the value vector $V^{(i)}$ can contain an arbitrary number of elements, as this number defines the size of the resulting context vector. The Softmax function (Eq. (6)) plays a crucial role in the Self-Attention mechanism. This function generates the probability scores combining the weights of the three matrices, as in Fig. 4. The process uses this probability distribution to compute the scaled dot-product attention function, as in Eq. (6).

$$\textbf{Query sequence}: \mathcal{Q}^{(i)} = \{W_q \mathbf{x}^{(i)} \text{ for } i \in [1, T]\} \tag{3}$$

$$\textbf{Key sequence}: \mathcal{K}^{(i)} = \{W_k \mathbf{x}^{(i)} \text{ for } i \in [1, T]\} \tag{4}$$

$$\textbf{Value sequence}: \mathcal{V}^{(i)} = \{W_v \mathbf{x}^{(i)} \text{ for } i \in [1, T]\} \tag{5}$$

$$\textbf{Attention}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = Softmax\left(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{d_K}}\right)\mathcal{V} = \mathcal{Z} \tag{6}$$

Word order is fundamental for understanding context in Natural Language [11,52,126]. However, in specific applications, such as graphs, this order is not relevant. Therefore, the model must be able to represent this order. For this reason, the Positional Embedding function proposed by Vaswani et al. [11] should be utilized so that the first block of the transformer, represented by the Encoder mechanisms, can receive the input Embeddings. Fig. 8 illustrates this mechanism, where, for each embedding of the original word, an embedding generated by Positional Encoding is added (represented by $(x^3_{n=1} + t^3_{n=1} = X^3_{n=1})$, to position the word in the sentence.

The Encoder generates the word representation (denoted as X_1, X_2, and X_3) by adding the embeddings of the words. This process uses vector position equations derived from sine and cosine functions to capture the relative positions of words in a sentence. This mechanism identifies the position of each word and calculates how many steps each word is ahead of or behind another. By incorporating this information, the model generalizes effectively, enabling the generation of novel sentences or leveraging other models to learn these positional relationships based on network training.

In order to calculate the Self-Attention for each word, the model generates query (Q), key (K), and value (V) vectors for each word. For example, the words "detection" and "attacks" produce Q_1, K_1, and V_1 for the first word, and Q_2, K_2, and V_2 for the second word. The model computes the inner product of Q_1 with K_1 (P_1) and with K_2 (P_2), iterating over all key vectors (K_n) to determine the relationships between the first word and the others.

The model calculates the inner product of the query vector Q_1 with itself and with each key vector (K_n), producing a number called the "score". For every word in the sentence, the model divides this score by the square root of 2 (d_k), a network hyperparameter stabilizing the gradient during training. The model uses the Softmax function to convert normalized scores into values ranging from 0 to 1, ensuring their sum equals 1.

This process distributes attention across the words in the sentence. In this example, words closely related to or more important for the target word, such as "Detection", receive higher Softmax scores, indicating their greater relevance, as shown in Fig. 6. During training, the neural network automatically learns to focus on the words most significant to the target word. Once the model determines these scores, it multiplies each score by the corresponding value vector (V), resulting in weighted representations for each word: $(softmax \times V_1)$, $(softmax \times V_2)$, $\ldots(softmax \times V_n)$. This step generates the final representation of the sentence.

The model calculates the sum of these weighted products for all vectors in the sentence, producing the Weighted Sum, which determines the value of the vector Z_1. This value represents the final contextualized representation of the first word, "Detection", in this example. The model repeats these calculations for each word in the input sequence, generating the corresponding vectors Z_2 through Z_n. Transformer excel at this task because they perform these calculations in parallel rather than sequentially, unlike LSTM or RNN models. This parallelism allows Transformer to handle large volumes of data efficiently during training, significantly improving performance. As detailed in Eq. (6), the model produces a matrix Z, where each row corresponds to a word and incorporates information from all other words in the sequence.

At this point, the Transformer architecture stands out, as these calculations do not need to be structured sequentially and can be performed in parallel, proving to be very efficient compared to LSTM or RNN techniques due to the large volume of data on which this model undergoes training.

The understanding of the equation is as follows: (6) obtains a matrix Z, where each row represents a word based on the other words.

$$Z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V) \tag{7}$$

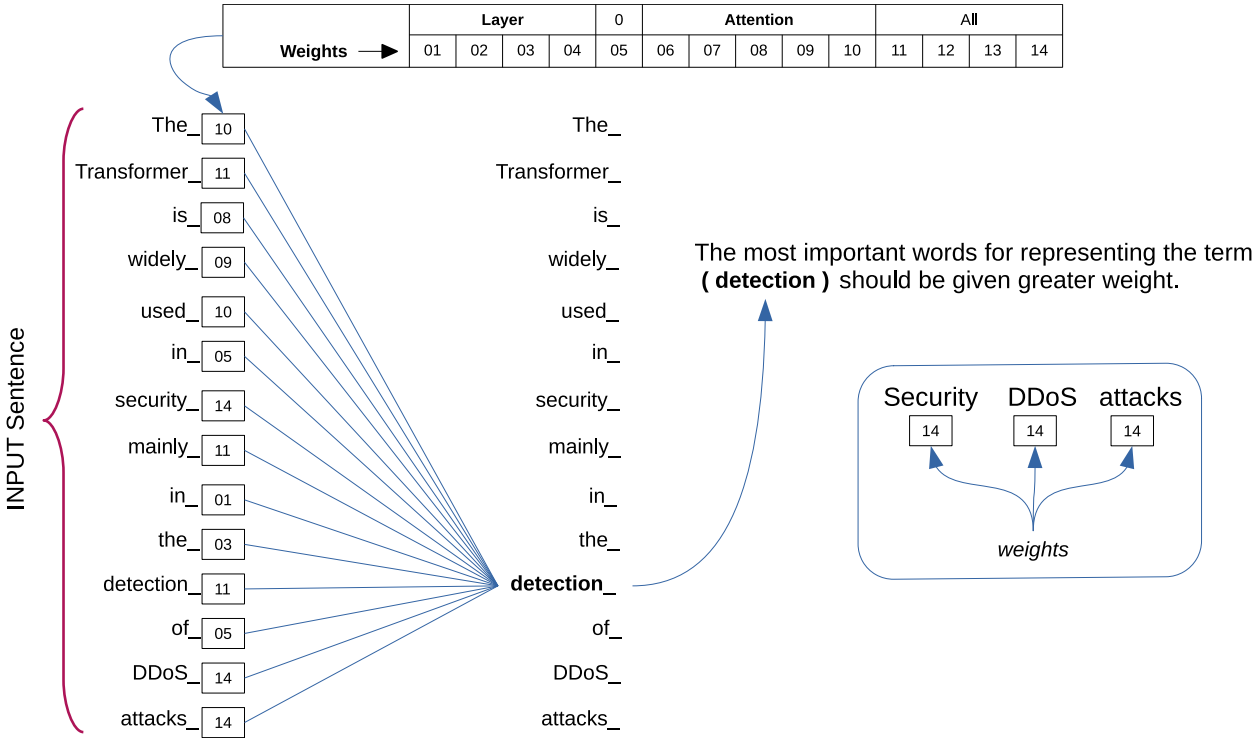


Fig. 6. Representation self-Attention weights on a sequence of tokens (based on [126]).

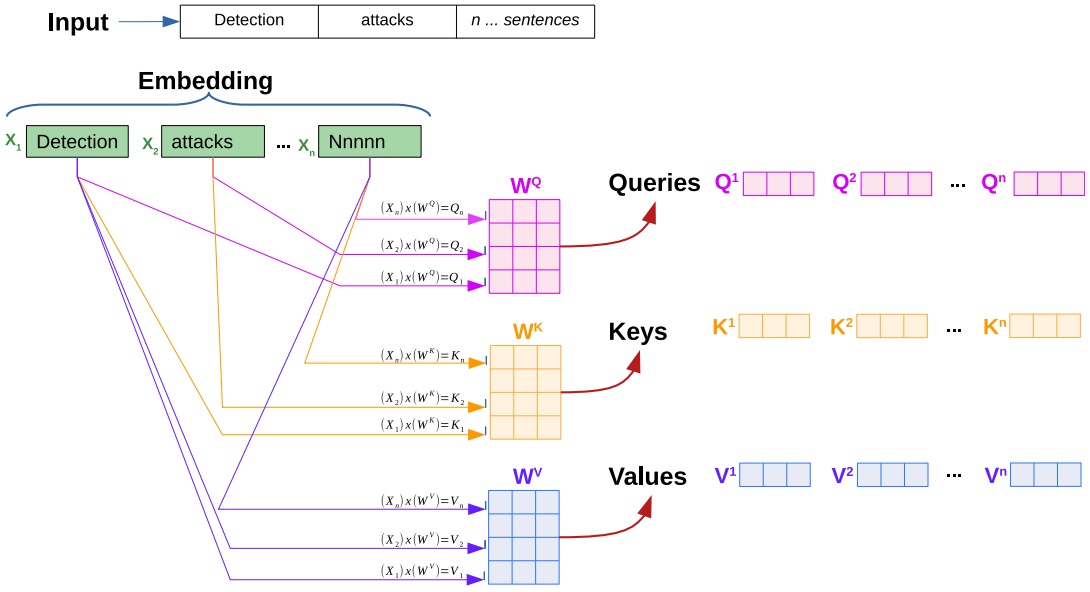


Fig. 7. Self-Attention in details [126].

The Transformer uses the Attention Mechanism called Multi-Head-Attention. The description presented in this survey was inspired and adapted from [11,129,130], and [126,131]. This mechanism carries out a series of matrix manipulations until a final Z matrix is generated, as presented here in phases (Fig. 9 [120]), which are:

- Phase I, where each row is an embedding or representation of a word, represented here by X_{ij};
- Phase II, the matrix X_{ij} must be multiplied by each of the matrices for each heading, i.e. $[W_0^Q, W_0^K, W_0^V, \ldots, W_n^Q, W_n^K, W_n^V]$;
- Phase III, the vectors generated by these matrix multiplications are subjected to Self-Attention calculations for $[Q_0K_0V_0, \ldots, Q_7K_7V_7)$ to generate the matrices for the next phase using the values of $[Z_0, \ldots, Z_7]$ for one of the heads;
- Phase IV, using the values obtained from $[Z_1, \ldots, Z_7]$, the Softmax function is applied to capture all the information in order to concatenate it, thus generating the W_0 matrix;
- Phase V, using the values obtained from the $Z = (Z_1, \ldots, Z_7)$ matrices, the information is aggregated into the W_0 matrix, which will be responsible for generating the final response using the Z matrix.

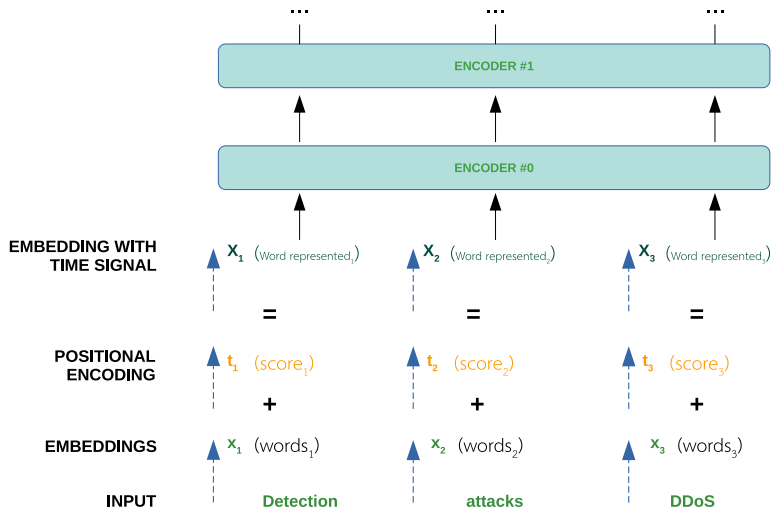


Fig. 8. Embedding the position [126].

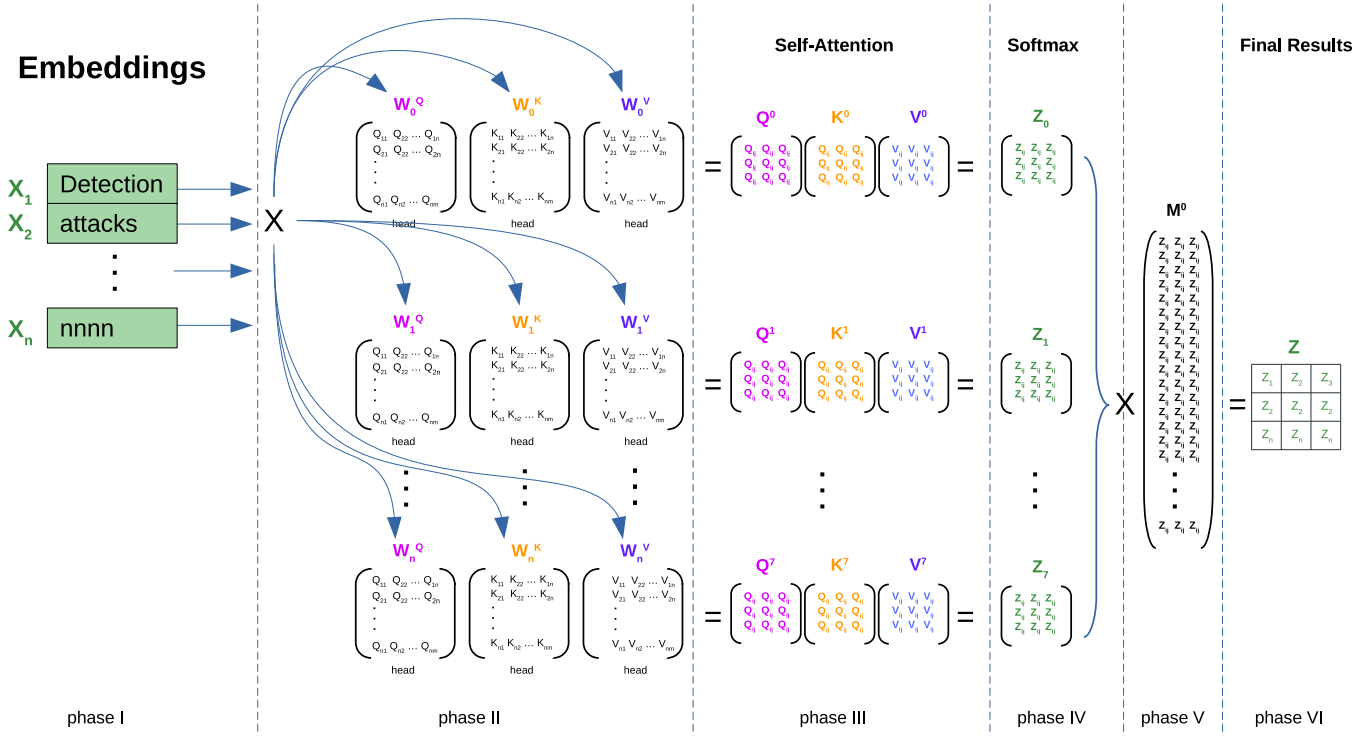


Fig. 9. Manipulating matrices with multi-head-attention.

- Finally, Phase VI produces the Z matrix, which represents the final result of all the information from the embeddings inserted into the attention mechanism. The final Z matrix shows that each row is the representation of a word ($X_1 \rightarrow Z_1$, $X_2 \rightarrow Z_2$, $X_n \rightarrow Z_n$), respectively.

This section provided an overview of the key components necessary for understanding the Transformer Architecture and its operation. Building on the paper of several authors, particularly Vaswani et al. [11], the Transformer model is recognized as a set of tasks that execute in parallel. This parallelism makes the Transformer more efficient and faster than conventional Deep Learning models, such as LSTM or RNN. To achieve this efficiency, it is essential to follow the phases outlined below:

- Data Entry: This phase involves converting raw input data into vector representations called embeddings. These embeddings are numerical encodings that allow the model to process and interpret the data effectively.
- Encoding: In this phase, the model uses stacked layers of Encoders. Each layer includes self-attention mechanisms and fully connected neural networks. The self-attention mechanism enables each token in the input sequence to attend to all other tokens, capturing long-range dependencies and contextual relationships. The result is a contextualized representation of the input sequence.
- Decoding: Similar to the encoding phase, the decoding phase adapts the process to generate output sequences. This phase integrates both self-attention and Encoder–Decoder attention mechanisms. Self-attention captures relationships within the output

sequence, while Encoder–Decoder attention extracts relevant information from the encoded input. The model generates tokens sequentially, with the output of one step feeding into the next.

- Output Generation: After decoding, the model's final layer produces a probability distribution over the vocabulary for each position in the output sequence. This distribution ranks potential tokens, enabling the model to select the next token in the sequence. This phase supports tasks like translation, summarization, and text generation by producing coherent output sequences.
- Training: During training, the model adjusts its parameters to minimize a loss function that compares its predictions with the target sequences. Using the backpropagation algorithm, the model computes gradients and updates its weights, improving its accuracy and performance.

Initially associated with NLP, the Transformer model has evolved into a robust machine-learning architecture that can be applied in several areas of knowledge, mainly cybersecurity. The attention mechanism is one of the main customizable functions of the Transformer, allowing the developed model to be capable of solving a wide variety of computational problems. This flexibility positions Transformer as one of the most effective machine learning and deep learning architectures, used in areas such as IoT, cloud computing, fog computing, and even in traditional networks, simple or complex, becoming one of the main tools for detecting DDoS attacks [132–135]. This dynamism of Transformer highlights its growing role, demonstrating its potential to redefine how this work secure digital systems across different domains.

The literature demonstrates that the Transformer model can outperform traditional methods such as seq2seq, particularly regarding learning efficiency [136]. The Transformer allows the simultaneous processing of entire data sequences, allowing the model to capture intra-sequence dependencies and effectively structure complex time series data. This capability is essential for modeling the diversity and dynamics of IoT systems, especially in environments where robots analyze applied papers. Furthermore, combining Transformer with other models, such as GCNN, has improved IoT network performance, highlighting the potential for integrating multiple techniques for detecting DDoS attacks.

Similarly, the paper proposed by [137] emphasizes the Transformer's superior accuracy and performance over traditional network architectures like RNNs. The attention mechanism, which assigns different weights to features during processing, significantly enhances the model's learning ability and allows it to focus on critical data. The advantage of this technique is particularly evident when dealing with long strings and simplifying data initialization. According to the authors, these properties make the Transformer model ideal for detecting intrusions in integrated power systems, which generate complex and large-scale data sets.

The work [138] confirms the Transformer model capacity to handle complex sequential data in IoT network environments by using it to classify traffic data for anomaly detection. Their method aims explicitly to enhance generalization for previously unseen malware scenarios, showing the ability of the model to adjust to new and unrecognized threats. Wang et al. [139] introduced a hybrid model that merges multi-scale convolutional neural networks with Transformer (DDoS-MSCT) to identify DDoS attacks in network traffic. The authors evaluated the model using the CIC-DDoS2019 and CIC-IDS2017 datasets and found that it accurately detects traffic patterns and anomalies. The model improves detection precision by combining local and global feature extraction, mainly for the complex nature of DDoS attacks. Its real-time processing of data enables it to detect attacks, making it well-suited for rapidly changing network environments.

They already [132] propose the RTIDS system, a robust Transformer-based approach for intrusion detection. Using self-attention techniques, RTIDS transforms high-dimensional data into low-dimensional representations, capturing contextual relationships between traffic features. Tested on CICIDS2017 and CIC-DDoS2019 datasets, RTIDS outperforms traditional methods, achieving higher precision and F1 scores. The system also addresses class imbalance issues through SMOTE, which helps improve the detection of less frequent types of attacks. RTIDS focuses on real-time operation, making it effective in dynamic networking environments.

2.2.8. Transformer architecture comparison

The Table 3 presents a comprehensive comparison between Transformers and other techniques, based on findings from recent papers that analyze their performance. These papers [140,141] highlight the advantages of the Transformer architecture and underscore its growing importance across various deep learning applications. While CNNs excel at processing structured image data and LSTMs are effective for time-dependent sequences, Transformers offer a more versatile and scalable architecture that efficiently handles visual and sequential data. Their attention-based mechanism allows them to capture global dependencies within the data, unlike CNNs, which are limited by local receptive fields, or LSTMs, which rely on stepwise memory and are prone to gradient issues. This global attention enables Transformers to process entire sequences in parallel, significantly accelerating training while improving contextual understanding.

Transformers stand out due to their adaptability across diverse domains, such as NLP, image classification, hyperspectral image (HSI) analysis, and multimodal tasks. They handle long-range dependencies more effectively than LSTMs and outperform CNNs in modern vision applications with less rigid data structures. Their scalability, training efficiency, and capacity to generalize across variable-length inputs make them the preferred architecture in many state-of-the-art systems. As deep learning tasks become more complex and data-driven, Transformers offers a flexible and robust solution that aligns with current demands in both research and real-world applications.

This table was structured in this way because comparing all 45 selected papers would not provide the same level of clarity and accuracy when analyzing the Transformer architecture in relation to other techniques. Each paper has its own particularities such as the type of dataset, preprocessing methods, and specific techniques implemented within the Transformer architecture which would make a fair and consistent evaluation difficult. These differences could compromise the credibility of the comparison. Therefore, to present a more appropriate and reliable analysis, this paper focused on three papers that were published with trustworthy data, clearly highlighting the advantages and prominence of the Transformer compared to other approaches.

3. The survey methodology

This section outlines the methodology organized to structure this survey, detailing the techniques and strategies for collecting relevant literature and developing evaluation criteria for paper analysis. The quality of the selected papers was assessed through ten key questions focused on detecting DDoS attacks using transformer architectures. The primary goal is to examine scientific papers using this architecture, their requirements, and their adaptations for addressing different types of DDoS attacks. Additionally, this survey highlights the gap in current research, as no prior paper specifically addresses DDoS detection through transformer models, leading to an in-depth investigation of DDoS attacks and ML and DP techniques utilized in this context. This section comprises two subsections: the first addresses the entire methodology for collecting bibliographic references for the construction of this survey, and the second subsection explains how the methods create variants that the transformer model can provide, leveraging the customizability of TA.

This research began with a preliminary survey based on bibliographic references, which guided the study and contributed to its preparation. The process involved analyzing several sources whose exploration provided the theoretical and methodological basis necessary

Table 3
Comparison between Transformers, CNNs, and LSTMs — Highlighting key advantages of Transformers.

Criterion	CNNs	Transformers vs. CNNs	LSTMs	Transformers vs. LSTMs	Transformer advantage
Architecture	Uses 2D/3D convolutions for spatial/temporal patterns.	Uses attention to dynamically weigh spatial regions.	Uses memory cells with sequential processing.	Uses parallel attention for full-sequence modeling.	Global, parallel attention
Ideal data type	Structured image/video data.	Effective for both visual and spectral–sequential data.	Time-series and variable-length sequences.	Better generalization for variable and long sequences.	Versatile across domains
Performance	Strong in image-rich and class-imbalanced datasets.	Matches CNNs in visual tasks, even using only spectral data.	Performs well in translation and temporal tasks.	Outperforms LSTMs in complex sequence modeling and HSI tasks.	Broad, cross-task performance
Training efficiency	May require more memory/time in high dimensions.	Lower memory use in spectral tasks; better training speed with parallelism.	Slower due to sequential updates.	Faster, parallel training even for long sequences.	Faster, scalable training
Memory Handling	Limited to local receptive fields.	Attends globally to all features in input.	Maintains long-term memory through gated cells.	Captures long-range dependencies more efficiently.	Stronger long-range memory
Flexibility and scalability	Less flexible with variable input formats.	Easily adapts to different input types (text, image, spectrum).	Limited by stepwise input handling.	Supports dynamic input lengths and complex dependencies.	Highly flexible and scalable
Applications	Computer vision (e.g., classification, detection).	NLP, HSI classification, modern vision, multimodal analysis.	Speech, time-series forecasting, translation.	Broader use across text, HSI, and vision.	Broadest application scope

for the construction of the survey. Also, the references have provided a theoretical scientific foundation for the topic addressed in this survey.

In order to achieve relevant research, the systematic literature review (SRL) and specific research questions defined the focus to identify primary studies, extract data from the papers, and analyze them. Therefore, the research questions developed as a basis to guide this Survey are:

- **RQ1:** What are the Transformer approaches applied to types of denial of service attacks, and how can these approaches be categorized?
- **RQ2:** What are the methodologies, strengths, and weaknesses of existing Transformer model approaches for detecting DoS and DDoS attacks in real-time or not?
- **RQ3:** What types of ML and DL algorithms excel in handling DoS and DDoS attacks?
- **RQ4:** How does the choice of tokenization or segmentation techniques affect the quality of learning in the Transformer model applied to detect DDoS Attacks?
- **RQ5:** What is the impact of data normalization strategies on the overall performance of the Transformer when dealing with malicious traffic patterns?
- **RQ6:** How have Transformer-based approaches been applied to DDoS attack detection in three distinct contexts, real-time settings, production environments, and testbed scenarios, and what are the challenges and advancements associated with each of these implementations?
- **RQ7:** What are the research gaps in the literature?

In order to ensure the credibility of this survey, a structured procedure was developed, as illustrated in Fig. 10. The figure describes the main phases, stages, and tasks involved in selecting relevant papers for the survey. It was necessary to include search methods using specific sequences, data storage protocols, acceptance and exclusion criteria, and the final selection of the collected papers. Thus, it was necessary to follow two major phases:

- Phase I — planning: At this moment, Stage 1 is applied, which contains Task 1, Task 2 and Task 3;
- Phase II — execution: containing Stage 2 and Stage 3. Stage 2 is composed of Task 4, and Stage 3 is composed of Tasks 5 and 6, respectively.

The methodology for this survey has followed two phases, consisting of three stages and six tasks. Phase I focuses on planning, while Phase II is about execution. In Phase I, Stage 1 entails organizing and creating Tasks 1, 2, and 3. Thus, Task 1 (Stage 1) involves developing questions that effectively measure the quality of the papers. To accomplish this, Task 1 defines a set of Evaluation Questions (EQ), resulting in a total of 10 quality assessment questions, which are:

- **EQ1 -** Is there innovation in the application of the Transformer Model to detect DDoS attacks?
- **EQ2 -** Is there a description of how the pre-processing was built to apply the Transformer model?
- **EQ3 -** Does the study present well-defined experiments?
- **EQ4 -** What are the main challenges in detecting DDoS attacks through the application of the Transformer model?
- **EQ5 -** Does the study present test data on the solution proving its effectiveness?
- **EQ6 -** To what extent do approaches beyond labeled data support the detection of DDoS attacks through Transformer Models?
- **EQ7 -** Is the evaluation of the results well described and clear for reproduction?
- **EQ8 -** Are the Transformer model application criteria well described?
- **EQ9 -** Is the Study a Paper or survey? (for SURVEY = Partial)?
- **EQ10 -** Does the paper present information relevant to the current research?

In Task 2 (Step 1), the search process combines the most significant possible number of relevant texts that meet specific criteria for inclusion in selecting scientific papers evaluated in this research. As a result of the process outlined in Task 2, this paper created nine search strings, which are presented in Table 4. Task 3 (Stage 1) was developed to select research sources and extract relevant literature for this survey. Through this Task, 11 search environments were selected: Scholar Google, IEEE Explorer, Springer, ACM, MDPI, Scopus, Elsevier, USENIX Security, Wiley Research, ArXiv Research, and others.

Phase II (Execution) includes Stages 2 and 3. In Stage 2, Task 4 consists of applying nine search queries across ten repositories, as shown in Fig. 11. The researchers submitted each of the nine queries to all eleven repositories, continuing the process until all searches were completed. They defined the search period as 2018 to 2024, based on the widespread adoption of Transformer Models since 2017, as

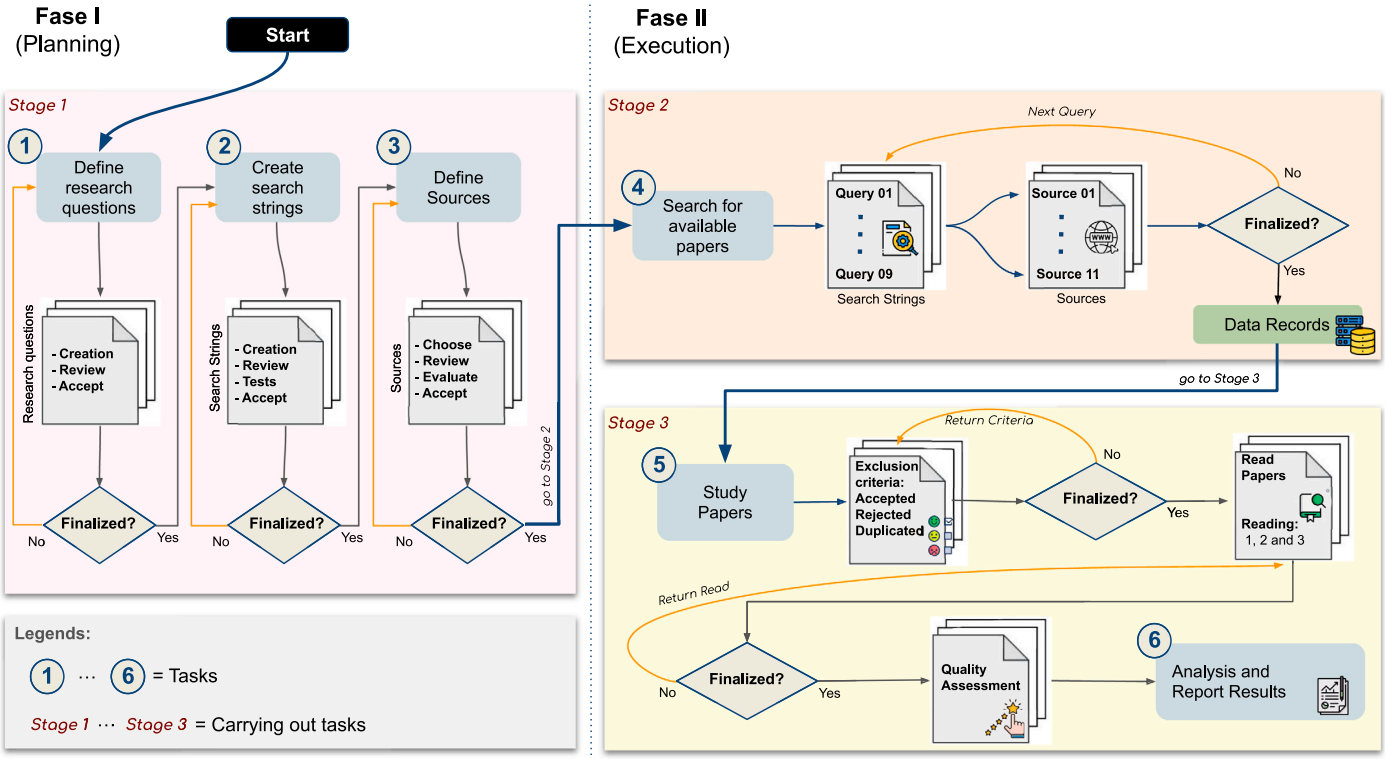


Fig. 10. Survey method.

Table 4
Keywords search methods.

Number	Query
01	(("SURVEY") AND ("Generative I.A." OR "GAN") AND ("Intrusion detection" OR "Intrusion prevention") OR ("Attacks DoS" OR "DDoS")))
02	(("Generative I.A." OR "GAN") AND ("Intrusion detection" OR "Intrusion Prevention") OR ("DDoS" OR "Transformers"))
03	(("Attacks DDoS" OR "Detection") AND ("Cybersecurity" OR "Generative Adversarial Network") OR ("Prediction"))
04	(("Attacks DDoS" OR "Prediction") AND ("Cybersecurity" OR "Generative Adversarial Network") OR ("Detection"))
05	(("Attacks DoS" OR "DDoS") AND ("Intrusion detection" OR "Intrusion Prevention") OR ("Generative I.A." OR "GAN")))
06	(("Transformers") AND ("Network DDoS" OR "DDoS Attacks" OR "DDoS Detection" OR "DDoS Attack Prediction"))
07	("transformers") AND ("Detection attacks DDoS" OR "detection attacks DoS" OR "Intrusion detection" OR "prevention attacks DDoS" OR "prevention attacks DoS")
08	("transformer networks DDoS")
09	(("transformers" OR "transformers network" OR "transformers neural network") AND ("Detection attacks DDoS" OR "Detection attacks dos" OR "Intrusion detection" OR "Prevention attacks DDoS" OR "Prevention attacks DoS"))

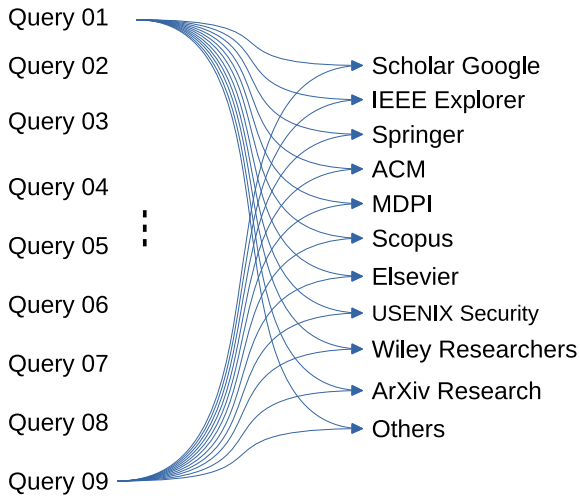


Fig. 11. Keywords search methods.

highlighted in [11]. These searches yielded a total of 603 papers, which the team then forwarded to Step 3 of the methodology.

Exclusion criteria were established to maintain a precise and methodologically sound scope, with a specific focus on the use of Transformer-based models for DDoS attack detection. Given that the first known research applying Transformers to this domain was observed from 2018 onwards, this led to limiting the selection to papers published since then. Although the Transformer architecture was initially proposed in 2017, its application to cybersecurity challenges (particularly for DDoS detection) only began to take shape in the following year. This time limit ensures that the reviewed papers are contextually relevant and reflect the current stage of technological evolution. Additionally, secondary analyzes were excluded to focus on original research that directly contributes to the technical discourse. This approach avoids the layers of interpretation introduced by reviews and allows for a more authentic assessment of the Transformer's performance. By adopting this selective framework, the proposal is to highlight the strategic relevance of Transformer models in detecting DDoS attacks as through this modern technique, emphasizing its initial but growing influence and its potential to reshape cybersecurity defenses.

Stage 3 begins with Task 5, the document analysis phase, where exclusion criteria are applied to each paper, as described in Table 5 titled "Inclusion Criteria and exclusion". They are questions about prediction,

Table 5
Inclusion and exclusion criteria used.

Inclusion criteria	Exclusion criteria
Detection attacks DoS and DDoS	Duplicate document
Intrusion detection	Secondary or tertiary studies
Prevention attacks DoS and DDoS	Papers before January 1, 2024
Cybersecurity	Out of scope
Prediction	Outside the scope of the Transformer theme
Transformer	Gray literature (textbooks, reports, theses, dissertations)
Transformer approach attacks DDoS	Short paper
DoS	Non-academic paper
DDoS	Do not have access to the paper
Transformer model in detecting attacks	Papers after October 1, 2024

Transformer, approach to DDoS attacks, DoS and transformer model in attack detection. These are the key questions for a paper to be accepted as the main part of the composition of this Research.

The exclusion criteria [142] aim to exclude papers that do not enter as main papers. They are papers that do not meet the requirements, i.e., they may be in at least one of the following situations:

- duplicate documents: can be found when searching for a string, a paper in more than one repository;
- secondary studies: preliminary studies of a primary study, i.e., initial study of a paper;
- tertiary studies: this is the review of the secondary study, i.e., these researches are not useful at this time, as they are still in the analysis stage of the main studies;
- paper prior to 2018: TA was created in 2017. TA application to attack detection dates from 2018;
- Out-of-scope papers: These papers contain similar information but are irrelevant to this research. For example, an electrical power transformer may appear in search results. It is not related to AI, but to electrical power equipment;
- Gray literature: gray readings are textbooks, reports, theses, and dissertations, as they are texts outside of published scientific papers;
- Short paper: papers that have less than four pages;
- Non-academic paper: ex. a company white paper;
- Papers without access: these are papers only available through subscription or payment, limiting access to information potentially relevant to research;
- This survey considers papers up to October 1, 2024, as this was the date on which the survey was conducted the searches.

After initial assessment based on established criteria, 603 papers were pre-selected for further review. These papers were prepared for the next step, Step 3, Task 5 (Reading Papers). The first reading phase followed the "Skimming" technique, as described, which consisted of a superficial analysis of the abstracts and figures of the papers, with the aim of evaluating their relevance to the topic of security against denial of service attacks using the Transformer Model. Then, a second, more detailed reading was carried out, applying the "Scanning" technique, which focused on reading the summary and the main body of the texts to extract specific information related to the topic. Skimming and Scanning [143,144] techniques, combined with the exclusion criteria, were essential in screening the 603 papers, resulting in the selection of 45 for complete reading and the rejection of 558 papers.

This research conducted a comprehensive literature review, using tables, graphs, and figures to detail the results. Variables such as year of publication, dataset, advantages, results, research gaps, type of publication, journal and ML, DL, and TA techniques applied with tools were analyzed. The quantitative and qualitative data survey provided an overview of the current research scenario in detecting DDoS attacks through the application of TA, highlighting trends, identifying gaps, suggesting opportunities for future studies, and presenting a structured perspective on the area's evolution.

4. Findings and insights: DDoS attack detection using transformer

This section provides a comprehensive literature review of selected papers related to applying Transformer models to detect DDoS attacks. The objective is to summarize the literature's leading trends, innovations, and patterns. Therefore, the analysis includes a discussion of frequently used datasets and an exploration of the effectiveness of various approaches that use TA. Another critical point to highlight is the focus on evaluation metrics, such as precision, recall, and F1 Score, among other metrics considered for performance analysis. Furthermore, the section also presents a complete analysis of the strengths and limitations of the approaches contained in this study. It compares the advantages of applying Transformer models with traditional ML and DL methods. Finally, the review aims to highlight the current state of the art, as well as identify existing gaps and suggestions for future studies for the application of TA in the detection of DDoS attacks.

4.1. Classification variants — method

Variants are different ways of customizing the Transformer Architecture to build a model applied to detect DDoS attacks. The idea of creating variants for the types of Transformers applied is because there are different forms of customization in all layers that make up the Transformer architecture. Therefore, to simplify understanding of the other customizations built in the various models presented, the variants guide and classify such modifications in the Transformer Architecture. Therefore, to define and classify the variations, it was necessary to establish a set of questions aimed at naturally evaluating each paper reviewed. These questions try to identify common properties that distinguish the variations, as follows.

- **VC-1:** Does this paper use Encoder, Decoder, or Encoder and Decoder to solve the proposed problem?
- **VC-2:** Which ML algorithms did this paper use to solve the proposed problem?
- **VC-3:** Which Machine Learning technology was used in this paper? Supervised, Unsupervised, Semi-supervised, or Self-supervised?
- **VC-4:** Which layers were used or changed in the Transformer architecture: Encoder Input Embedding, Multi-head Attention, Add & Norm, and Feed Forward layers. In the Decoder layer: Input Embedding, Masked Multi-Head Attention, Add & Norm, Multi-head Attention, Feed Forward, Linear, Softmax?
- **VC-5:** According to the architecture, which of these variants is suitable to classify the paper?

The Transformer architecture is entirely modular, as detailed by Vaswani et al. [11]. Based on this principle, Section 2.2.6 shows a reproduced TA with all layers enumerated according to Fig. 4. This enumeration helps identify the full range of environments within a complete TA and highlights the architecture's flexibility. This approach provides a framework for evaluating and categorizing TA variants based on their effectiveness in addressing DDoS attacks, offering precise insights into their relative strengths and applications. The entire description of these enumerated layers is demonstrated in Section 2.2.7 according to [11,52,145–147].

4.1.1. Layers and transformer description architecture

In Section 2.2.7 is highlighted by modular categorization and TA customization. The purpose is to allow researchers to adapt its components to specific requirements, such as detecting DDoS attacks. Therefore, a list of standard approaches is presented that shows how these features are used to address various challenges and improve performance in multiple applications:

- **A variation in the model's components:** For this type of variant, it is possible to evaluate the importance of different components of the Transformer architecture in various ways in the models and thus measure the performance impacted by such variations. One of the most common actions is to vary the number of attention heads, as well as the dimensions and values of attention, as demonstrated in the paper [11] (section 6.2) with the application of different components;
- **Incorporation of more elaborate attention techniques:** The discussion of all the papers researched in this paper reflects the importance of the attention mechanism in the Transformer and the different approaches to this mechanism that can be applied and explored. In this case, you can, for example, use weighted attention techniques, multi-head internal attention, or directly customized ones to provide a significant improvement in the model performance;
- **Integration in the reinforcement mechanism:** This reinforcement learning technique is the part that can be combined with the Transformer to improve the training process and, thus, the optimization of the system. In these cases, the evaluation between deliberations and rewards during training has the purpose of helping to improve the model, thus improving the quality of the expected result, specifically in cases of translations;
- **The exploration of regularization techniques:** In this case, techniques such as dropout, layer normalization, or specific regularization techniques for neural networks can also be incorporated into the Transformer architecture in order to avoid overfitting and improve the generalization of the model.

These are some ways in which, as described in the paper by [11], it is possible to make customizations, modifications, or even incorporate techniques not only from Machine Learning (ML) and Deep Learning (DL) but also from other approaches that contribute to helping Transformer Architectures (TA) achieve their proposed objectives. This explanation clarifies that the examples mentioned earlier refer specifically to this paper. Several papers have since been proposed, including the use of different types of neural networks, such as Long Short-Term Memory (LSTM), and sets of ML and DL algorithms to solve problems in diverse areas of knowledge [102,148,149]. The classification of the proposed variants is presented in Table 6, which provides each variant's name, acronym, and description. These variants are based on studies of personalization types identified in various papers, with acronyms suggested to categorize the TA variants. This table, derived from selected papers, can be further expanded as the literature evolves to include additional acronyms and variants applicable to various fields, providing a foundation for continued exploration and refinement.

Thus, the names of the variants were chosen according to the classification performed by the level of customization found in the references, whether in a layer, in the block of layers or in the entire Transformer architecture. As support in the construction of the classification of the variants, the Sub Section 2.2.6 presents the Transformer Architecture where all the essential points that can be customized were identified. In turn, the Sub Section 2.2.7 presents the description of each of the enumerated layers. Thus, this information aims to guide on the possibilities that the Transformer Architecture can provide modularity and scalability in the customization, alteration or creation of new acceptable solutions within the layers of this architecture.

Table 6
Transformer model variants classification.

Name	Acronym	Description
Adaptive Transformer	ADT	Variant that dynamically adjusts its layers or parameters during training or inference within the Transformer architecture
Attention-Enhanced Transformer	AET	This variant is identified through some improvement or change in attention mechanisms
Custom Transformer	CST	This variant is identified when there are customizations, changes, adaptations or alterations in the architecture or in any layer of the Transformer architecture
Hybrid Transformer	HBT	This variant is identified when in its structure, there are several ML, DL or other techniques that are added to the Transformer architecture infrastructure
Real-Time Processing Transformer	RPT	This technique is particularly identified because it deals with real-time data and time series. In this case, there is normally a different task, mainly due to the data processing type used for training. This technique was defined by the fact that data generated by computer networks needs to be analyzed in real-time
Standard Transformer	STD	This variant refers to the Transformer architecture standard, i.e., it will only be used when there is a solution that makes use of the Transformer without any changes to the architecture nor to the layers that make the transformer model paper properly [11].

4.2. Classification of variants

A review of several studies identified the Variants Classification Method described in Section 4.1, which is grounded in the current literature. This method clarifies and organizes how the Transformer Architecture (TA) has been adapted for DDoS attack detection. Rather than focusing on detecting specific types of attacks, the proposed classification highlights the versatility of the Transformer, which is made possible by its modular design and high capacity for customization. These qualities allow the integration of diverse techniques from machine learning, deep learning, and other innovative domains, reinforcing the Transformer's growing relevance and adaptability across different application contexts [11,39].

Despite this potential, a significant challenge in the literature is the absence of a standardized classification for these architectural adaptations. Researchers often introduce modifications and enhancements without a consistent naming or structural framework, leading to ambiguity about the nature and scope of their contributions [11]. Table 6 addresses this gap by providing a structured classification of six Transformer variant types used in DDoS detection. Importantly, these variants were not created to specialize in identifying specific types of attacks, such as low-rate or high-rate traffic. Instead, each variant reflects a distinct architectural strategy for customizing the Transformer model. For instance, Custom Transformers (CST) and Hybrid Transformers (HBT) involve structural changes and the incorporation of external techniques. At the same time, Real-Time Processing Transformers (RPT) are designed to process time-sensitive data. These classifications help guide researchers in selecting appropriate model configurations based on their design needs, not based on attack types.

Consequently, all variants demonstrated the capability to detect diverse DDoS attacks, highlighting the inherent flexibility of the Transformer architecture. The proposed classification thus captures the primary distinctions necessary for model application and removes the need for additional categorization based on attack type. Therefore, it is necessary to determine the important characteristics for each paper to be evaluated, which are:

- **Model Components: Ec** — Encoder, **De** — Decoder, **ED** — Encoder and Decoder;
- **Algorithms: ML, DL**;
- **Technique: Su** — Supervised Learning, **Un** — Unsupervised; **Sm** — Semi-Supervised; **SS** — Self-Supervised; **OD** — Others;
- **Changed Layers:** Only layers that have been changed (Section 2.2.6);
- **Brief Description of**: presents a short description of what the paper uses and the justification for why it is classified according to the variant given to the cited paper.

4.2.1. ADT

The paper [150] presents the TransIDS model, which employs the Transformer architecture and focuses on layers (01), (02), (03), (04), (05), (a), (c), and (N) to detect intrusions. The model selects these layers to increase efficiency by using Encoder's self-attention mechanism to extract complex patterns from network traffic. The model reduces the computational complexity by using only **Ec** and omitting the Decoder, making it suitable for classification tasks. The technique exploits **Su** learning like Label Smoothing to improve generalization and deal with class imbalance. The paper integrates several ML and DL algorithms to strengthen DDoS attack detection, including E-GraphSAGE, LSTM, RNN, GMS-IDS, Extra Trees, TCN, and XCM. These algorithms offer a variety of intrusion detection approaches, focusing on DDoS attacks. Therefore, the **ADT** variant is suitable for TransIDS due to its adaptability to changes in network traffic. Combining these techniques, the model improves accuracy, making it ideal for high-risk environments, such as IoT networks prone to DDoS threats.

4.2.2. AET

The paper [54] integrates both the TA **ED** block in its methodology. In addition, the proposal uses several algorithms, such as LSTM-IDS, CTIDS, CNN-LSTM-IDS, and CNN. In the presented architecture, **Su** techniques were used but presented modifications in some layers, which were used (01), (02), (03), (04), (05), (a), (b), (c), (Nx), (N) — It is essential to highlight that the number of layers is not mentioned, (RC) and (N). In this way, the **AET** variant is suitable, as the authors apply improved attention mechanisms to capture long-range relationships in data sequences, which is essential for detecting distributed intrusions. To achieve this, the model implements three self-attention mechanisms (SA1, SA2, and SA3) instead of a single layer, capturing different levels of relevance between input data and improving global analysis capacity. This variant prioritizes relevant information and captures complex patterns, offering flexibility and adaptability to varying data types and scenarios with evasive attack patterns.

4.2.3. CST

Based on the work [146], it was possible to detect that it uses the **Ec** component of the Transformer architecture, in addition to employing several algorithms, including SVM, RF, KNN, CNN-TWD, SSL, AB, SRDLM, SNADE, and LDA-ELM, along with **Su** learning techniques. Several modifications were implemented in layers (01), (02), (03), (04), (05), (08), (a), (RC), (N) of the architecture. Therefore, the **CST** variant is ideal for this paper, as it incorporates the segmentation of network traffic into packet windows, which is then processed using the Transformer model. This approach demonstrates customization of the standard Transformer to detect network traffic anomalies. The model implements Matmul in the architecture, a key component in the attention mechanism computation, allowing it to determine the relationship between traffic packets based on their characteristics. Through this paper, it is possible to enable the model to capture contextual and relational dependencies between packets, facilitating the efficient identification of anomalies in network traffic.

4.2.4. HBT

In [139], only **Ec** is utilized. This paper applies the ML and DL algorithms RF, SVM, LR, CNN, LSTM, and 1D-Inception, following a pattern based on **Su** learning. Modifications span layers (01), (02), (03), (04), (05), (a), (c), (Nx). In the context of this paper, the most effective variant for detecting DDoS attacks is **HBT**. The proposed DDoS-MSCT model combines a multi-scalar convolutional neural network with a Transformer, integrating ML and DL techniques. Additionally, it leverages the ability of the Transformer to model long-range dependencies. The paper combines the different techniques that Transformer employs in an MHSA engine: Global Feature Extraction Module (GFEM), Local Feature Extraction Module (LFEM), dropout rate, and batch normalization (BN). These techniques and changes to the Transformer architecture were implemented to improve the detection of DDoS attacks by leveraging the combined advantages of CNNs in feature extraction and Transformer networks in modeling long-term dependencies.

4.2.5. RPT

Based on the outlined characteristics, the papers highlighted key components and methodologies in applying Transformer architectures for detecting DDoS attacks. The paper by [52] focuses on the **Ec** component of the Transformer, utilizing algorithms such as RNN, Seq2Seq, and CNN-LSTM and employing **Su** techniques. Modifications were made to layers (01), (02), (03), (04), (05), (a), (c), (Nx), (RC), and (N) of the Transformer model. The authors proposed the **RPT** variant since the attention mechanism enables the model to capture the temporal dependencies and correlations between data inputs over time. This is fundamental for intrusion detection in time series, where network behavior is followed continuously. The model also uses multiple encoder layers that benefit from the advantages of the attention mechanism, allowing better processing of long data sequences, which is relevant for real-time network traffic analysis. Furthermore, the paper mentions that the detection procedure involves rigorous data preprocessing to ensure that the model correctly represents and analyzes the temporal characteristics.

4.2.6. STD

The paper by [109] also employs an **Ec** architecture, using a variety of algorithms such as NB, DT, Bagging, RF, SVM, KNN, Adaboost, and LR under **Su** learning. Modified layers include (01), (02), (03), (04), (05), (a), (c), (Nx), (RC), and (N). The **STD** variant is considered appropriate for this context because it captures long-term dependencies in IoT traffic. The multi-head attention mechanism allows the model to focus on different parts of traffic even when interference from anomalous traffic occurs, while automated feature extraction increases accuracy in identifying IoT devices. Furthermore, the implementations shown in Figs. 3 and 4 demonstrate how the proposed method uses the capabilities of the Transformer architecture to improve the identification of IoT devices, dealing with routine and anomalous traffic effectively. The method flow and feature processing module indicate an advancement over traditional approaches, leveraging self-attention and positional coding to accurately capture temporal features without altering the Transformer architecture.

Table 7 significantly by detailing the type of model and the form of learning used to detect DDoS attacks, in addition to specifying whether the paper uses labeled data and whether the techniques are applied to time systems real during attacks. The table aims to guide researchers in their papers, serving as a reference when choosing experiments according to the dataset type, the nature of the data in real-time or not, and the form of machine learning. Researchers can use this table to help select solutions suited to their experimental context, whether driven by labeled data, real-time learning, or a combination of both.

In the "Model" field, the table identifies the type of Transformer architecture applied, such as "Encoder", "Decoder", "Encoder/Decoder" or other variations. The table also specifies whether the data used is labeled, thus indicating solutions for varying scenarios. The table

Table 7
Models and machine learning types in Transformer architecture.

Authors	Year	Model	Labeled data (Yes/No)	Real-time (Yes/No)	Learning form
[19]	2022	Encoder	Yes	Yes	Supervised
[12]	2023	Encoder	Yes	Yes	Supervised
[14]	2023	Encoder	Yes	No	Supervised
[52]	2024	Encoder	Yes	No	Supervised
[69]	2019	Encoder	Yes	Yes	Supervised
[74]	2023	Encoder	Yes	No	Supervised
[146]	2022	Encoder	Yes	No	Supervised
[149]	2023	Encoder	Yes	No	Supervised
[151]	2023	Encoder	Yes	No	Supervised
[152]	2023	Encoder	Yes	No	Supervised
[153]	2022	Encoder	Yes	No	Supervised
[134]	2023	Encoder	Yes	Yes	Supervised
[154]	2024	Encoder	Yes	No	Supervised
[133]	2023	Encoder	Yes	No	Supervised
[150]	2023	Encoder	Yes	No	Supervised
[155]	2023	Encoder	Yes	No	Supervised
[132]	2022	Encoder	Yes	Yes	Supervised
[156]	2022	Encoder	Yes	No	Supervised
[157]	2022	Encoder	Yes	No	Supervised
[139]	2024	Encoder	Yes	Yes	Supervised
–	–	–	–	–	–
[13]	2022	Encoder	No	No	Supervised and Semi-Supervised
[158]	2023	Encoder	Yes	No	Supervised and Unsupervised
[148]	2022	Encoder	Both	No	Semi-supervised
[17]	2022	Encoder	No	No	Unsupervised
[159]	2021	Encoder	No	No	Unsupervised
–	–	–	–	–	–
[16]	2021	Encoder/Decoder	Yes	No	Supervised
[70]	2023	Encoder/Decoder	Yes	No	Supervised
[75]	2023	Encoder/Decoder	Yes	No	Supervised
[147]	2023	Encoder/Decoder	Yes	Yes	Supervised
[118]	2024	Encoder/Decoder	Yes	No	Supervised
[160]	2023	Encoder/Decoder	Yes	No	Supervised
[109]	2022	Encoder/Decoder	Yes	Yes	Supervised
[161]	2023	Encoder/Decoder	Yes	No	Supervised
–	–	–	–	–	–
[162]	2022	BERT	Yes	No	Supervised
[163]	2024	BERT	Yes	No	Supervised
[164]	2024	SecurityBERT	Yes	Yes	Supervised
[165]	2022	CAN-BERT	Yes	Yes	Unsupervised
–	–	–	–	–	–
[54]	2022	CTIDS	Yes	No	Supervised
[166]	2022	CIDS-Net	Yes	No	Supervised
[117]	2022	CNN-Transf. NIDS	Yes	Yes	Supervised
[167]	2023	Denseformer	Yes	No	Supervised
[51]	2023	GSF-TNN	Yes	No	Supervised
[145]	2022	MFVT	Yes	No	Supervised
[168]	2022	TAB	Yes	No	Supervised
[53]	2022	VSI-TN	Yes	No	Supervised

indicates the data type processed in the "Real Time" field, while the "Learning Form" field describes the machine learning algorithms used in each study. This informative structure is essential for evaluating the complexity and scope of models applied to specific problems in computer security, especially in the areas of detection, prediction, vulnerability analysis, and monitoring.

Table 7 organizes and categorizes 45 scientific papers that in some way used machine learning models based on Transformer architectures, showing the efficiency of the approaches to detect DDoS attacks. The methodology requires each paper to be classified according to the Transformer model used. Hence, the information analyzed is labeled data (yes, no, or both), real-time applicability, and the type of learning employed (supervised, semi-supervised, or unsupervised). Finally, it groups the papers into five main categories: Encoder (20 papers), Encoder/Decoder (8 papers), BERT and variants (4 papers), in addition to other specific models, such as CTIDS, Denseformer, and VSI-TN (8 documents in total). This organization facilitates comparative analysis between models, identifying patterns and trends in different implementations. The Table presents the characteristics of each paper analyzed, such as the year of publication, the authors, and the specificities of the models. This information is important to understand the main trends and gaps in the area of cybersecurity. It is possible to analyze that the Encoder models and supervised learning are predominantly used in the papers. The papers that use the BERT model explore supervised and unsupervised learning. Applications aimed at real-time detection vary considerably between groups, as the solutions presented are diverse. In this way, the table presented becomes a good tool for evaluating the state of the art and identifying possible directions for developing new models through varied implementations for detecting DDoS attacks.

An important point is that all papers that address real-time data use labeled data, which reinforces the importance of labeling for real-time detection solutions. However, it is observed that only two of these papers [17,165] adopt unsupervised learning methods. These findings highlight the need to assess the status of data labeling to meet real-time detection demands, highlighting a possible dependence on previously

labeled data and indicating opportunities to explore new approaches, such as unsupervised methods, to increase the flexibility and efficiency of the proposed solutions.

Table 8 presents a comprehensive comparative overview of different Transformer model-based approaches applied to network intrusion detection and cybersecurity. This comparison details, for each study analyzed, the used datasets, the proposed (and compared) models, and the most relevant metrics for performance evaluation: F1-Score, Accuracy, Precision, and Recall. The primary purpose of this table is to provide a unique and exclusive panorama of the best metrics achieved by each solution in a standardized and comparable way, helping to identify the most promising models within the state of the art.

Each row in the table corresponds to a scientific paper, in which the highlighted model (marked with **(*)**) represents the author's main proposal, generally a variation or improvement based on the Transformer architecture. The models are tested on different datasets widely recognized in the cybersecurity domain, such as CIC-IDS2017, NSL-KDD, UNSW-NB15, and ToN-IoT. Specific datasets are also present, such as those focused on DoH traffic, vehicular networks, SCADA/ICS, and industrial or IoT environments, and proprietary datasets, such as those made available by the University of New Brunswick.

The information about the metrics were extracted directly from the papers and standardized for a transparent and objective comparison. In cases where a metric was not available in the paper, the corresponding field is filled in with an indication of absence ("**N/A**"). Including the four leading performance indicators allows not only the evaluation of the general effectiveness of the models but also the analysis of their specific strengths, for example, models with high precision but lower recall or vice-versa. This table not only aggregates data in an organized way but also highlights the best solutions based on the reported metrics, allowing the reader to identify which proposals stand out in specific contexts. Therefore, this compilation offers a solid basis for decision-making and directing future research in DDoS attack detection based on deep learning and Transformer architectures.

4.2.7. Computational efficiency analysis of transformer-based models

Assessing the computational efficiency of Transformer-based models used for DDoS attack detection is fundamental for their practical deployment in network security, especially in real-time network environments. As these scenarios often demand rapid decision-making under limited computational resources, understanding the algorithmic cost becomes indispensable in determining feasibility. The "Efficiency" column in the comparative table provides the asymptotic computational complexity, expressed in Big-O notation, for each model. It offers a detailed view of the resource requirements associated with their execution. This analysis draws upon foundational papers such as those by [169–172], which provide a theoretical grounding for the complexity profiles of Transformer, BERT, LSTM, CNN, and other model architectures implemented across the reviewed papers.

The majority of Transformer-based models listed in the table follow a computational complexity of $O(n^2 \cdot d)$, where "n" denotes the input sequence length (typically the number of tokens or network packets) and "d" represents the dimensionality of the embedding vectors. This quadratic complexity arises from the standard self-attention mechanism, which computes pairwise interactions between all input tokens. While this structure enables strong contextual representations and typically results in high accuracy, F1-score, and precision, it poses serious scalability issues in high-throughput environments. In real-time traffic analysis systems such as IDS, where packet rates can be extremely high, this level of computational demand can lead to increased latency and energy consumption, making such models unsuitable for resource-constrained or time-sensitive applications.

Several studies propose more efficient architectures that reduce complexity to linear levels to address these limitations, specifically $O(n \cdot d)$. Models such as those in [69,74,146], and [118] illustrate this shift toward computationally leaner solutions. These models often adopt approximations of attention mechanisms or introduce temporal filtering techniques that reduce the volume of internal operations. Although they may experience a marginal drop in predictive performance, they gain significantly regarding inference time and hardware efficiency. Hence, they are particularly suitable for deployment in edge computing scenarios, embedded systems, and other distributed infrastructures where computational resources are reduced.

In contrast, some models integrate more complex structures and additional components, increasing overall computational costs. For example, the model proposed by [158] has a complexity of $O(n \cdot d^2 + n^2 + m^2 \cdot p)$, reflecting a hybrid framework with multiple layers and parameterized modules designed to extract deeper semantic features. Similarly, [157] reports a complexity of $O(n^2 \cdot d + m)$, which combines quadratic attention costs with additional operations dependent on external model parameters. These models often aim to balance semantic richness and detection granularity, but their higher computational overhead may limit their practicality for deployment at scale.

Other papers report unconventional complexity forms tailored to specific architectures or application domains. For instance, [162] defines the model complexity as $O(N^2 D_k + N^2 D_v)$, and [164] reports $O(N_M^2 + N_E \cdot M + N \log N)$, incorporating multiple input variables tied to attention matrices, feature dimensions, or specialized processing stages. These custom formulations often reflect optimizations for IoT security or industrial environments with lightweight processing and context-aware detection.

BERT-based models and their variants, such as CAN-BERT [165], BERT-MLP [163], and SecurityBERT [164], also commonly operate under a complexity of $O(N^2 D)$ or similar expressions, due to their use of full attention and deep contextual embeddings. These models are known for their high classification accuracy and robustness across multiple datasets. However, their computational burden remains a significant constraint for real-time or large-scale deployment, especially in streaming-based DDoS detection systems where rapid throughput and low latency are essential.

This comparative analysis shows that models with linear complexity $O(n \cdot d)$ offer the best prospects for real-time DDoS detection in constrained environments, balancing acceptable predictive performance with efficient computation. Conversely, models with quadratic or hybrid complexity, such as $O(n^2 \cdot d)$ or more elaborate variants, although effective in capturing complex patterns, require careful consideration regarding their scalability and energy efficiency. The trade-off between detection accuracy and computational feasibility must be carefully managed when selecting models for deployment in production-environment cybersecurity infrastructures.

In order to analyze and reinforce the papers demonstrating the complexity and efficiency of these algorithms, this study reviewed 45 papers addressing hyperparameter adjustment in Transformer-based models for intrusion detection. The majority, 25 papers (approximately 55%) employed manual processes to define hyperparameters, relying on extensive empirical experimentation to maximize model performance [12,13,16,17,19,52–54,69,75,109,132–134,139,145,149–151,155,157,160,164,165,167]. This predominance of manual tuning reflects both the technical complexity and the high computational costs often associated with automated optimization in certain contexts. Additionally, only four papers (about 9%) adopted hybrid approaches that combined manual and automated methods, such as grid search and Bayesian optimization, aiming to balance tuning efficiency with model accuracy [14,152,154,162]. Conversely, 15 papers (around 33%) did not clearly describe their hyperparameter selection processes, limiting methodological transparency and exposing a gap in the documentation practices within a significant portion of the literature [51,70,74,117,146–148,153,156,158,159,161,163,166,168]. Finally, only one paper (2% of all papers) employed a fully automated method, systematically applying grid search techniques [118]. These findings suggest that, although automated techniques are gradually gaining traction, manual

Table 8
Transformer-based approaches and evaluation metrics.

Authors	Datasets	Model	F1-Score	Accuracy	Precision	Recall	Efficiency
[19]	Dataset 1: DoH and Non-DoH traffic classification, Dataset 2: DoH traffic and malicious DoH traffic classification.	DoH Tunneling (*), Malicious DoH, DoH Traffic, DoH Resolver, DoH Proxy	99.00%	99.40%	99.00%	99.00%	$O(n^2 \cdot d)$
[12]	NSL-KDD, CIC-IDS2017, MQTTset	Res-TranBiLSTM (*), LeNet5, VGG16, ResNet18, ResNet18 + BiLSTM	99.88%	N/A	99.07%	98.96%	$O(n^2 \cdot d)$
[14]	ToN_IoT dataset	MTNN, LSTM (*), RNN	63.94%	57.15%	33.09%	70.35%	$O(n^2 \cdot d)$
[52]	CIC-IDS 2018	Transformer-based (*), CNN-LSTM	99.53%	99.98%	99.71%	99.35%	$O(n^2 \cdot d)$
[69]	CICIDS2017	Bi-LSTM, CRF, ANID (*)	95.28%	N/A	95.28%	94.40%	$O(n \cdot d)$
[74]	Dataset made available by the University of New Brunswick	Frequency Enhanced Decomposed Transformer (FEDformer), Patch Time Series Transformer (PatchTST) (*)	N/A	N/A	N/A	N/A	$O(n \cdot d)$
[146]	CIC-IDS2017	AdaBoost, SRDLM, CNN-TWD, LDA-ELM, SNADE, SSL, HAST-IDS, PWT-30 (*)	95.60%	96.10%	94.30%	97.10%	$O(n \cdot d)$
[149]	Car Hacking dataset, In-Vehicle Network, Intrusion detection dataset, Survival analysis dataset	Transformer-Based Attention Network (*), RNN, LSTM, CNN-LSTM	99.93%	N/A	99.93%	99.93%	$O(n^2 \cdot d)$
[151]	SDN Network Flow Data of RLFA	RLFAT (*), LSTM, GRU, CNN, TopoGuard+, MLLG	95.20%	95.10%	95.50%	95.00%	$O(n^2 \cdot d)$
[152]	KDDCup99, NSL-KDD, CICIDS-2017	CNN, Transformer, CNN-LSTM, CNN-Transformer (*), LSTM, GRU, NB, RF, KNN	100.00%	99.90%	99.00%	100.00%	$O(n^2 \cdot d)$
[153]	CIC-IDS 2017, UNSW-NB15, Kitsune	Decision Tree, Tab-Transformer, FT-Transformer, Cascade (Tab-DT), Cascade (FT-DT) (*), MLP-Embeds, MLP-Embeds-Attn	83.20%	N/A	84.60%	83.60%	$O(n^2 \cdot d)$
[134]	ToN IoT dataset	FT-Transformer1, FT-Transformer2, CNN-IDS, FED-IDS, ResNet-50, P-ResNet	96.94%	97.06%	97.23%	96.67%	$O(n^2 \cdot d)$
[154]	UNSW-NB15, CIC-IDS2017, NSL-KDD	CNN-LSTM (*), CNN-RNN, CNN-GRU	99.00%	99.21%	99.00%	100.00%	$O(n^2 \cdot d)$
[133]	UNSW-NB15	EVM, IDS-GAN, CNN-BiLSTM, CNN-WDLSTM, DUA-IDS (*), OCN	88.54%	88.47%	87.94%	N/A	$O(m \cdot d)$
[150]	TON-IoT	E-GraphSAGE, LSTM, RNN, GMS-IDS, ExtraTrees-IDS, TCN, XCM, TransIDS (sem Label Smoothing), TransIDS (*)	99.44%	99.46%	99.46%	99.43%	$O(n^2 \cdot d)$
[155]	ISCX2012 Dataset, CICIDS2017 Dataset	KNN, LR, DT, GBDT, SVM, CNN, LSTM, Transformer, CNN+LSTM, GID (*)	99.37%	99.42%	99.41%	99.72%	$O(n^2 \cdot d)$
[132]	CICIDS2017, CIC-DDoS2019	SVM, RNN,FNN, LSTM, RTIDS (*)	99.97%	99.98%	99.99%	99.98%	$O(n^2 \cdot d)$
[156]	Temperature Residuals Dataset (owner)	RF, LR, GBDT, GN, LDA, SVM, Transformer, Encoder-Transformer (*)	99.00%	N/A	100.00%	99.00%	$O(n^2 \cdot d)$
[157]	NetML, CICIDS2017, CICDDoS2019	R1DIT (*), Self-Attention Based Model, Siamese Architecture	97.50%	99.99%	99.91%	99.99%	$O(n^2 \cdot d + m)$
[139]	CIC-IDS2017, CIC-DDoS2019	CNN, DDoSNet, DDoSLSTM, GRU, DDoSTC, DDoS-MSCT (*)	99.95%	99.97%	99.94%	99.95%	$O(n^2 \cdot d)$
[13]	HDFS dataset, Firewall dataset	PCA, DeepLog (LSTM model), LogAnomaly (LSTM model), LogRobust (Bi-LSTM model), HitAnomaly (BERT-based log and parameter embeddings), ROBERTa (transformer variant), Single AAs	98.00%	N/A	99.00%	98.00%	$O(n^2 \cdot d)$
[158]	NSL-KDD dataset	CNN, CNN-LSTM, CBA-CLSVE, SSC-OCSVM, CNN-GRU, FCNN-SE, RTIDS, VIT, CNN-Transformer, Ours (*)	88.20%	88.70%	N/A	N/A	$O(n \cdot d^2 + n^2 + m^2 \cdot p)$
[148]	CIC-IDS2017, CSE-CIC-IDS2018	KNN, ID3, Adaboost, MLP, PBCNN, Semi-SVM, Semi-RF, ESeT (*)	99.60%	99.80%	99.50%	99.50%	$O(n^2 \cdot d)$

(continued on next page)

Table 8 (*continued*).

[17]	NSL-KDD	DNN, CNN+BiLSTM, ResNet-18, ResNet-50, IG-PCA-Ensemble, DT-EnSVM2, ViT, Improved ViT (*)	99.65%	99.68%	99.74%	99.57%	$O(n^2 \cdot d)$
[159]	SWaT, WADI, SMAP, MSL	PCA, AE, KitNet, DAGMM, GAN-Li, OmniAnomaly, LSTM-VAE, MAD-GAN, KNN, FB, MTAD-GAT, GDN, LSTM-NDT, GTA (*)	91.00%	N/A	89.00%	92.00%	$O(n^2 \cdot d)$
[16]	CICDDoS2019	Transformer, Transformer + LSTM + CNN (TLC), Transformer + LSTM (TL), DDosTC (*)	99.92%	99.82%	99.88%	99.96%	$O(n^2 \cdot d)$
[70]	SWaT (Secure Water Treatment)	STA-Tran (*), TranAD	91.09%	91.42%	91.91%	91.72%	$O(n^2 \cdot d)$
[75]	KDD Cup 1999, CICIDS2017	CNNs, RNNs, Transformer Models (*)	94.00%	96.00%	94.00%	94.00%	$O(n^2 + n \cdot d)$
[147]	UNSW-NB15	Decision Transformer (DT) (*), Behavior Cloning (BC), Conservative Q-Learning (CQL), Deep Neural Network (DNN)	99.00%	99.33%	99.00%	99.00%	$O(n^2 \cdot d)$
[118]	NSL-KDD, UNSW-NB15, CSE-CIC-IDS2018, MQTT-IoT-IDS2020, CSE-CIC-IDS, TON-IOT , UNSW-NB15, CSE-CIC-IDS2018	Flowtransformer (*), RTIDS, VIT, CNN-Transformer	99.90%	N/A	N/A	N/A	$O(n \cdot d)$
[160]	WUSTL-IIoT-2021	Transformer (*), ECOD, DeepSVDD, RNN-AE	94.31%	97.44%	N/A	N/A	$O(n^2 \cdot d)$
[109]	N-BaIoT, Training set 1, Training set 2, Test set 1, Test set 2	NB, SVM, LR, kNN, DT, RF, CNN, MLP, Transformer Based Device-Type (*)	100.00%	100.00%	N/A	N/A	$O(n^2 \cdot d)$
[161]	ISCXIDS2012 dataset, CSE-CIC-IDS2018 data	Non-image Algorithm, Channel Algorithm, Proposed Algorithm (*)	93.86%	95.60%	95.41%	92.55%	$O(n^2)$
[162]	ECU-IoHT, ICE, ToN-IoT, Edge_IIoTset, EMBER	LightGBM, BERT-based Transformer (*), BiLSTM	100.00%	100.00%	100.00%	100.00%	$O(N^2 D_k + N^2 D_v)$
[163]	CIC-IDS 2017, UNSW-NB 2015, NSL-KDD 2009	BERT-MLP (*), DT, SOM, DNN, NDAE	99.99%	99.39%	99.34%	99.33%	$O(max(N^2, m \cdot n \log(n)))$
[164]	Edge-IIoTset	DT, RF, SVM, KNN, CNN, RNN, LSTM, DNN, Transformer model w/o Tokenization and Embedding, SecurityBERT with PPFLE (*)	98.00%	98.20%	98.00%	98.00%	$O(N_M^2 + N_E \cdot M + N \log N)$
[165]	Car hacking: Attack and defense challenge 2020 — collected from three different cars: Chevrolet Spark, Hyundai Sonata, Kia Soul	CAN-BERT (*), iForest, PCA, BiLSTM-AE-4, LSTM-AE-4, LSTM-AE-8	99.00%	N/A	98.00%	99.00%	$O(N^2 D)$
[54]	CICIDS2017	LSTM-IDS, CNN-LSTM-IDS, CTIDS (*)	96.60%	96.80%	96.40%	96.80%	$O(n^2 \cdot d)$
[166]	SCVIC-CIDS-2021	CIDS-Net (*), LR, AB, NB, GB, DT, RF, XGBoost (XGB), TabNet	99.89%	N/A	N/A	N/A	$O(n^2 \cdot d)$
[117]	KDDCUP 99, NSL-KDD, UNSW-NB15	Transformer NIDS, CNN NIDS, CNN-Transformer NIDS (-Attn), CNN-Transformer NIDS (+Attn) (*)	— (O*)	— (O*)	— (O*)	— (O*)	$O(n^2 \cdot d)$
[167]	NSL-KDD, KDD-CUP99	KNN, SVC, XGB, DT, LGBM, MLP, Conv+BiLSTM, Conv+LSTM, CNN-IDS, HLSTM, Denseformer (*)	84.89%	85.64%	85.97%	85.64%	$O(n^2 \cdot d)$
[51]	WUSTL-IIOT-2018 ICS SCADA cyber security dataset.	GSFTNN (*), ResNet, RNN, LSTM	99.20%	99.26%	98.85%	98.85%	$O(max(n^2 \cdot d, g \cdot n^2))$
[145]	IDS 2012, IDS 2017	MFVT, MFVT (CPR) (*)	99.99%	N/A	99.99%	99.99%	$O(n^2 \cdot (d + m))$
[168]	UNSW-NB15, NSLKDD, KDD98, KDDCUP 99, CIDDS-001, DARPA, ADFA	RF, DT,KNN, LSVM, Logistic Regress, MLP, 1D CNN, Tab Transformer (*)	96.68%	98.35%	95.79%	97.60%	$O(n \cdot l \cdot d)$
[53]	UVSI-DDoS-I, UVSI-DDoS-II, CIC-DDoS2019, UNSW-NB15	VSI-TN (*) , VSI-RTN, DeROL(Deep Reinforcement Learning model), Bi-LSTM, LSTM	N/A	98.26%	97.99%	98.87%	$O(K \cdot n^2 \cdot d)$

Note:
- **(*)** Main dataset for evaluation.
- **(N/A)** In F1-Score, Accuracy, Precision and Recall value was Not Available by the authors.
- **(O*)** Others Metrics DR (Detection Rate), FDR (False Detection Rate), MDR (Missing Detection Rate) e FAR (False Alarm Rate).

experimentation remains the dominant approach in the field, due to its flexibility and lower computational demands in research environments.

Table 9 presents a comparative summary of recent Transformer-based approaches for network intrusion detection. It synthesizes key information from selected papers, including the models employed, the datasets used for evaluation, the experimental protocols adopted (such as the number of evaluation repetitions or folds), and the main performance metrics reported (accuracy, precision, recall, F1-score). Additionally, the table indicates whether the papers employed measures to assess the robustness and statistical significance of their results, such as cross-validation, repeated experiments, or variance analyses. This synthesis aims to provide a clear overview of the current state of research in this area, highlighting both methodological strengths and common gaps related to experimental rigor and reporting of results.

By analyzing the methods and metrics used in Transformer-based intrusion detection papers presented in the Table 9, it is evident that although most research adopts traditional metrics such as precision, recall, and F1-score to evaluate model performance, few papers conduct in-depth analyses on the stability and reliability of the results. Among the most relevant cases, those that applied cross-validation or multiple repetitions stand out, although they did not always detail variability measures such as the standard deviation. For example, [152] applied 10-fold cross-validation to improve the reliability of the results, while [153] applied a stratified 5-fold split. Similarly, [148] conducted robustness analyses considering average repetitions for each level of perturbation, and [75] reported multiple repetitions with different initializations, although without providing details on statistical variability. Other papers, such as [149,151], present a descriptive analysis of performance variation with changes in decision thresholds, regarding the F1-score, which raises concerns about model stability.

On the other hand, most of the analyzed papers limit themselves to reporting point metrics obtained from single repetitions without formally exploring variability measures or conducting statistical tests to validate model robustness. Examples include [12,19,69], and [146], which, despite presenting relevant performance metrics, do not specify the number of experimental repetitions nor apply statistical analyses to ensure the reliability of the results. In addition, many papers, such as [150,154], and [157], merely report aggregated F1-score and precision values, without mentioning practices such as calculating the standard deviation or conducting significance tests. This trend reveals a methodological gap in the current literature, where the predominance of classical metrics contrasts with a widespread absence of robust statistical evaluation practices to ensure the stability and reliability of Transformer-based models applied to intrusion detection.

5. Discussion

This section presents a broad discussion of the content covered in this survey. Different options and formats for the Transformer Model have been explored using the standard Transformer Architecture. The dynamism offered by this architecture has allowed several applications to be developed in various areas of knowledge, including security against DDoS attacks, which is the focus of this study. Artificial intelligence, primarily through its subareas of Machine Learning and Deep Learning, plays an essential role in the Transformer Architecture, particularly in the customizations applied to Transformer models aimed at solving different problems, with emphasis on the detection of DDoS attacks [74,139,161].

In this context, relevant research questions were formulated, which are answered based on a detailed analysis of several papers aimed at detecting DDoS attacks, both directly and indirectly. In this context, relevant research questions were formulated, which are answered based on a detailed analysis of several papers aimed at detecting DDoS attacks, both directly and indirectly, using different anomaly detection techniques or customized IDS. These systems, in turn, have specific characteristics for detecting DDoS attacks. The questions were organized into RQ1, RQ2, RQ3, RQ4, RQ5, RQ6 and RQ7, and are widely discussed in Section 6. These questions not only deepen the understanding of the state of the art but also seek to guide future research for developing robust and real-time solutions to combat DoS and DDoS attacks using different anomaly detection techniques or customized IDS. These systems, in turn, have specific characteristics for detecting DDoS attacks. These questions not only deepen the understanding of the state of the art but also seek to guide future research for developing robust and real-time solutions to combat DoS and DDoS attacks.

RQ1: What are the Transformer approaches applied to types of DDoS attacks, and how can these approaches be categorized? This survey is worth highlighting the importance of Tables 6 and 7 built in this survey. Table 6 arises from the need for an organized understanding of different Transformer forms and models. Customizations are implemented and made without distinction by utilizing different ML and DL techniques and proprietary solutions integrated into the Transformer architecture. The purpose of this table is to standardize how applications are searched according to the customized form of the transformer model through acronyms classified as TA variants. The suggested variants are ADT, AET, CST, HBT, RPT, and STD.

The Table 7 stands out. It was designed to evaluate the use of applications in detecting DDoS attacks, whether directly or indirectly, considering that many papers do not exclusively deal with this type of attack. From this table, relevant statistics can be extracted to guide researchers in using the Transformer Architecture to detect DDoS attacks. Based on the groupings presented in the table, 44.44% of the 45 papers analyzed used only the Transformer Architecture Encoder. Papers that used both the Encoder and the Decoder totaled 17.78%. Those exploring the BERT approach instead of the standard Transformer corresponded to 8.89%.

This survey analyzed three different categories of features of papers that deal with DDoS attacks. His papers that deal exclusively with DDoS attacks focus only on this type of threat or DoS attacks without mentioning or addressing other types of attacks, totaling 7 papers [12,16,52,74,132,139,154]. Additionally, 22 papers explored DDoS attacks in conjunction with different types of threats, such as slow-rate attacks, SQL injection, Cross-Site Scripting (XSS), Man-in-the-Middle (MITM), backdoor attacks, password attacks, ransomware, scanning attacks, injection attacks, flooding attacks, fuzzy attacks, malfunction attacks, port scans, vulnerability exploitation, tunneling DoH, spam, command and control (C2), radio frequency interference (RF Jamming), Sybil attacks, among others [14,51,53,75,117,133,134,145,146,148–150,153,155,157,158,161–163,167,168]. The paper proposed by [164] stands out for presenting the SecurityBERT model, designed to detect cyber attacks in real-time. The model addressed fourteen distinct types of attacks, primarily focusing on DoS and DDoS attacks. Although the paper does not detail all kinds of attacks mentioned, the main objective was to explore the application of the model in IoT environments, considering this diversity of threats.

Finally, 16 papers were found that indirectly addressed DDoS attacks. These papers focused on anomaly detection, IDS systems for monitoring network flows, models such as the Packet Window Transformer (PWT), and solutions aimed at detecting code injections, such as the DUA-IDS system (Dynamic Unknown Attack Intrusion Detection System). Other papers analyzed IDS systems for vehicle CAN buses, addressing attacks such as flooding attacks (Flood Attack/DoS), fuzzy attacks, spoofing attacks (Spoofing Attack), and replay attacks (Replay Attack). These solutions, in large part, were implemented using models based on Transformer [13,17,19,54,69,70,109,118,147,151,152,156,159,160,165,166].

RQ2: What are existing Transformer model approaches' methodologies, strengths, and weaknesses for detecting DoS and DDoS attacks in real-time or not? Question RQ2 stands out due to the relevance of DDoS attack detection time, whether in real-time or not. According to the papers analyzed in this survey, approximately 24.44% of the papers

Table 9
Overview of the stability of models transformers applied in DDoS detection.

Authors	Datasets	F1-Score Range	Trials	Observations
[19]	Dataset 1: DoH and Non-DoH traffic classification, Dataset 2: DoH traffic and malicious DoH traffic classification.	91.00%–99.00%	N/A	This paper does not provide further details (particularly regarding the number of experiments). This paper reports only precision metrics without addressing stability and reliability through statistical analysis. Recall and F1-score were also used to evaluate the performance, demonstrating the effectiveness of the proposed solution.
[12]	NSL-KDD, CIC-IDS2017, MQTTset	99.01%–99.88%	N/A	Although the number of traials is not specified, early stopping was used to halt Res-TranBiLSTM training when validation accuracy decreased, indicating potential overfitting.
[14]	ToN_IoT dataset	56.72%–63.95%	N/A	The paper provides precision and recall but does not explicitly report the F-score, leaving it for the reader to derive. The results of the paper show that MTNN improves the F-score by up to 63% over baseline models, highlighting the importance of this metric.
[52]	CIC-IDS 2018	12.10%–99.53%	N/A	The paper employs traditional metrics to evaluate model quality and presents performance analyses across various configurations. However, it lacks robust statistical methods to assess result stability and reliability, such as multiple repetitions with variance calculations or significance testing
[69]	CICIDS2017	23.34%–95.28%	N/A	The evaluation relies on measuring the F-score from a single experiment, trained to convergence with early stopping, without reporting multiple repetitions, variability measures, or statistical significance analysis.
[74]	Dataset made available by the University of New Brunswick	N/A	N/A	In the paper, only "*loos*" values were utilized to validate the technique.
[146]	CIC-IDS2017	77.00%–95.60%	N/A	Although the paper employs traditional classification metrics for Transformer-based Intrusion detection (Accuracy, Precision, Recall, F1), it does not adopt rigorous procedures to assess the stability of the model and robustness through multiple repetitions or statistical analyses of the experimental results.
[149]	Car hacking dataset, In-Vehicle Network, Intrusion detection dataset, Survival analysis dataset	92.00%–100.00%	N/A	It presents F1 scores for Intrusion detection using 16 CAN IDs in the car hacking dataset, illustrating how different thresholds affect performance. The best result achieved 100%, showing how F1 varies with threshold adjustments and providing insights into model stability and reliability.
[151]	SDN network flow data of RLFA	85.00%–95.20%	N/A	The results presented indicate that proposed model outperforms the other models, showing a significant improvement in the F1-score.
[152]	KDDCup99, NSL-KDD, CICIDS-2017	89.00%–100.00%	10 fold	The performance of the proposed model is generally superior (higher bar for CNN-Trans). Although various results present precision (P) and detection rate (DR), the F1-score is derived from these two by: $F = \frac{2\times P\times DR}{P+DR}$.
[153]	CIC-IDS 2017, UNSW-NB15, Kitsune	55.90%–83.20%	5-fold	For the evaluation, the dataset was divided into five stratified folds, with 20% of the data allocated to each assessment fold and 80% for training, thereby characterizing the 5-fold cross-validation technique.
[134]	ToN IoT dataset	86.00%–96.94%	N/A	This paper compares the multi-class classification performance of the FT-Transformer model with other methods using key metrics, including Accuracy, F1-score, Recall, and Precision, and specifies the data used ("Data Usage"). While it reports F1-score values from selected experiments, it lacks details on the number of repetitions and does not provide additional analyses beyond F1-score, loss, accuracy, and precision.
[154]	UNSW-NB15, CIC-IDS2017, NSL-KDD	94.00%–99.00%	N/A	The paper uses standard metrics to evaluate performance but does not clearly report experimental repetitions, variability measures, or formal statistical analyses. In order to enhance reliability, it utilizes Explainable AI to interpret model decisions and highlight key features that influence classifications.
[133]	UNSW-NB15	83.79%–88.54%	06	The paper covered six unknown categories, indicating the training required to reach the observed values.
[150]	TON-IoT	82.30%–99.44%	N/A	Transformer (TransIDS), especially when combined with Label Smoothing, provides high reliability for IoT Intrusion detection, outperforming traditional methods and ensuring good performance even in complex and unbalanced real-world scenarios.

(*continued on next page*)

Table 9 (*continued*).

[155]	ISCX2012 Dataset, CICIDS2017 Dataset	39.20%–99.37%	N/A	The Transformer-based studies in the paper utilize standard classification metrics to evaluate performance, Including Accuracy, Precision, Recall, F1 score, true positive rate (TPR), and false positive rate (FPR). However, they do not apply formal statistical procedures to measure stability (e.g., multiple repetitions with standard deviation reporting) or conduct statistical tests to ensure the reliability of the results.
[132]	CICIDS2017, CIC-DDoS2019	81.86%–99.97%	N/A	Transformer-based Intrusion detection studies, as exemplified by this paper, typically rely on traditional evaluation metrics (Accuracy, Precision, Recall, F1-score). However, practices such as multiple experimental repetitions, reporting variability measures (e.g., standard deviation), and conducting formal statistical analyses are uncommon and were not adopted in this study.
[156]	Temperature residuals dataset (owner)	59.00%–99.00%	N/A	The study employs standard metrics to assess stability and reliability. Still, it lacks detailed statistical analysis, variability measures, and information on experimental repetitions, relying mainly on point metric values.
[157]	NetML, CICIDS2017, CICDDoS2019	95.20%–99.99%	N/A	The paper uses standard metrics (macro F1-score, TPR, FAR) to evaluate the performance of the Transformer for intrusion/malware detection, although it does not report the number of repetitions.
[13]	HDFS dataset, Firewall dataset	79.00%–98.00%	N/A	The papers in this work use classic metrics (F1-score, precision, and recall) to evaluate performance but do not report the number of experimental repetitions, statistical variability measures, or formal analyses. However, they complement the evaluation with explainability techniques to examine the model's robustness and reliability.
[158]	NSL-KDD dataset	81.60%–88.20%	N/A	In the paper, the assessment of model stability and reliability is based on single-point performance metrics (accuracy, F1-score) evaluated on fixed test sets without formal statistical analyses or explicit variability measures to quantify the stability of the experimental results.
[148]	CIC-IDS2017, CSE-CIC-IDS2018	47.40%–99.60%	10	Presents a robustness analysis that considers averages over repetitions for each level of disturbance, distinguishing between the scale of disturbance and the number of iterations.
[17]	NSL-KDD	98.82%–99.65%	N/A	N/A
[159]	SWaT, WADI, SMAP, MSL	71.00%–91.00%	N/A	The paper employs cross-validation to enhance the reliability of the evaluation. Still, it does not formally explore variability metrics or statistical analyses to assess the stability of Transformer-based models in Intrusion detection.
[16]	CICDDoS2019	99.84%–99.92%	N/A	The results are presented based on the best parameters identified (e.g., optimal epoch, learning rate, and architecture). However, there is no mention of evaluating the variability of the results or conducting multiple repetitions to validate the model's robustness.
[70]	SWaT (Secure Water Treatment)	71.71%–91.09%	N/A	Although the basic metrics were applied, the analysis of the model's stability and reliability in statistical terms may not have been thoroughly explored in the presented paper.
[75]	KDD Cup 1999, CICIDS2017	85.00%–94.00%	N/A	The paper reports multiple repetitions with different initializations to enhance the robustness of the results, providing average values for standard metrics (accuracy, precision, recall, and F1-score). However, it lacks details on the exact number of repetitions, variability measures, or formal significance analyses regarding the stability of Transformer-based Intrusion detection models.
[147]	UNSW-NB15	94.00%–99.00%	N/A	The models are evaluated using traditional classification metrics and combined measures of detection accuracy and speed. While the evaluation considers scenarios with varying data quality to assess robustness indirectly, there is no explicit mention of multiple repetitions, variability analysis (e.g., standard deviation), or statistical significance testing.
[118]	NSL-KDD, UNSW-NB15, CSE-CIC-IDS2018, MQTT-IoT-IDS2020, CSE-CIC-IDS, TON-IOT , UNSW-NB15, CSE-CIC-IDS2018	90.45%–99.90%	03	The paper mentions carrying out experimental repetitions to control the stability of the models.

(*continued on next page*)

Table 9 (*continued*).

[160]	WUSTL-IIoT-2021	81.04%–94.31%	N/A	The paper employs standard evaluation metrics (AUC, F-score, ROC, and PR curves) and analyzes training convergence to ensure the stability of the Transformer model in Intrusion detection. However, it does not include experimental procedures to assess result variability or detailed statistical analyses to confirm the reliability of the reported performance.
[161]	ISCXIDS2012 dataset, CSE-CIC-IDS2018 data	88.73%–93.86%	N/A	In this study, the performance of Intrusion detection models based on multiple vision Transformers is evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score, across specific datasets. However, the analysis does not include robust statistical evaluations or quantitative measures of variability across different runs.
[162]	ECU-IoHT, ICE, ToN-IoT, Edge_IIoTset, EMBER	76.00%–100.00%	N/A	The paper employs cross-validation to enhance the reliability of the evaluation. Still, it does not formally explore variability metrics or statistical analyses to assess the stability of Transformer-based models in Intrusion detection.
[163]	CIC-IDS 2017, UNSW-NB 2015, NSL-KDD 2009	80.00%–99.99%	N/A	While the proposed model evaluation lacks details on the number of repetitions (trials), it is notable for its quantitative precision, strong generalization, and stability, making it a promising approach for practical applications in network security systems.
[164]	Edge-IIoTset	57.00%–98.00%	N/A	The paper does not specify the number of experimental repetitions (trials), but it provides detailed performance metrics, especially F1-scores for various attack classes, demonstrating SecurityBERT's ability to accurately and robustly detect multiple attack categories. This detailed presentation helps compensate for the lack of information on repetitions and reinforces the reliability of the results.
[165]	Car hacking: Attack and defense challenge 2020 — collected from three different cars: Chevrolet spark, Hyundai sonata, Kia soul	81.00%–99.00%	N/A	The paper utilizes the F1-score as the primary metric to evaluate the effectiveness of CAN-BERT compared to standard baselines. However, it neither applies nor reports formal methods to assess model stability, such as experimental repetitions with variability measures (e.g., standard deviation) or statistical analyses to support the reliability of the results.
[54]	CICIDS2017	93.60%–96.60%	N/A	Although appropriate performance metrics are utilized, the study does not explicitly assess model reliability or stability through the analysis of result variability or quantitative statistical significance.
[166]	SCVIC-CIDS-2021	92.50%–99.89%	N/A	The combination of host and network data (CIDS) with appropriate architectures and customized loss led to exceptional performance (99.89% macro F1), highlighting the importance of integrating multiple data sources for IDS.
[117]	KDDCUP 99, NSL-KDD, UNSW-NB15	N/A	N/A	N/A
[167]	NSL-KDD, KDD-CUP99	56.60%–84.89%	N/A	Transformer-based Intrusion detection papers utilize traditional metrics, such as Accuracy, F1-score, Recall, and False Positive Rate, to assess performance. Stability is evaluated through experiments and comparisons across different architectures and depths.
[51]	WUSTL-IIOT-2018 ICS SCADA cyber security dataset.	98.97%–99.20%	N/A	The paper adopts standard performance evaluation metrics but does not explicitly describe or implement multiple experimental repetitions to ensure statistical robustness.
[145]	IDS 2012, IDS 2017	99.64%–100.00%	N/A	This paper presents experiments whose F1-score results are summarized in Table 5, showing the performance of the MFVT model combined with the CPR algorithm on the more complex and recent IDS 2017 dataset. The precision, recall, and F1-score metrics for each attack category are all near 1 (i.e., nearly 100%).
[168]	UNSW-NB15, NSLKDD, KDD98, KDDCUP 99, CIDDS-001, DARPA, ADFA	56.05%–96.68%	N/A	N/A
[53]	UVSI-DDoS-I, UVSI-DDoS-II, CIC-DDoS2019, UNSW-NB15	N/A	N/A	The model evaluation primarily relies on Area Under the Receiver Operating Characteristic Curve (AUC-ROC), as highlighted in Tables 5, 7, and 8 and discussed in the corresponding section. Additionally, Receiver Operating Characteristic (ROC) curves and analyses of True Positive Rate (TPR) and False Positive Rate (FPR) are presented in the results section, effectively replacing conventional metrics such as the F1-score.

cover real-time attacks. Furthermore, around 8.89% of papers do not use labeled data to detect DDoS attacks, highlighting a gap that still needs to be explored due to the scarcity of work in this field.

The papers in this survey identified different approaches to detecting attacks. Most have focused on utilizing the traditional Transformer model, which focuses on DDoS attacks and other types of threats. Among the strengths of these approaches is the efficient application of ML and DL techniques, which allow automatic feature extraction without relying on manual engineering-still a significant advantage. Also, Transformer captures long-term temporal dependencies, being beneficial for identifying traffic patterns.

However, researchers have identified weaknesses. Data preprocessing and the need for labeled data still prevail and require specialized knowledge, limiting the practical application of models. In particular, detecting attacks in real-time still presents challenges, as only 11.11% of the papers studied use unlabeled data. Furthermore, many papers do not detail the need for large amounts of labeled data for training nor discuss the complexity of the models compared to more straightforward approaches, which can impact the decision to use Transformer to detect DDoS attacks.

RQ3: Machine Learning (ML) and Deep Learning (DL) models, combined with Transformer, an advanced DL technique, are widely used to detect DDoS attacks. Among the most used methods, 80% of the papers analyzed in this survey mention the use of CNNs, CNN-LSTM, LSTM, RNNs, MLP, RF, NB, LR, RL, KNN, DT, and GRU [69,109]. In addition to these conventional approaches, autoEncoders have been highlighted. According to [158], the "stacked autoEncoder" was employed as a pre-processing technique that reduces the dimensionality of the data while preserving its essential characteristics. Another relevant approach is the Synthetic Sampling Technique (ADASYN), presented by [152]. Although not a classic ML or DL algorithm, ADASYN effectively deals with data imbalance, especially when intrusion samples are significantly smaller. Furthermore, other less common techniques have also been applied, such as Deep AutoEncoders (DAs), Deep Belief Networks (DBNs), Ensemble Classifier for Outliers Detection (ECOD), DeepSVDD, Recurrent Neural Network AutoEncoder (RNN-AE), Artificial Immune System (AIS), Conditional Variational AutoEncoders (CVAE), Bidirectional Long Short-Term Memory (BLSTM), eXtreme Gradient Boosting (XGBoost) and the PSO-GBoost hybrid model [14,147,160]. These approaches show that ML and DL algorithms are essential in DDoS attack detection. However, Transformer, according to most authors, stands out as the most promising and well-defined technique for dealing with attacks of this nature.

RQ4: How does the choice of tokenization or segmentation techniques affect the quality of learning in the Transformer model applied to detect DDoS Attacks? To understand the impact of the choice of tokenization and segmentation techniques on the effectiveness of Transformer models applied to the detection of DDoS attacks in network traffic, a comprehensive study of the currently available literature was necessary [14,17,51–53,70,74,117,118,132,139,152,154,157,158,160,163–165,168]. In this way, it was possible to understand that the way techniques are chosen directly influences the models' ability to process and learn relevant patterns from traffic data, affecting accuracy and efficiency. Through this analysis, it is possible to highlight the main aspects related to tokenization and segmentation in the context of network traffic to analyze DDoS attacks.

Determining factors when choosing tokenization and segmentation:

- Data Representation and Temporal Context: Well-planned tokenization captures temporal and sequential relationships between data packets, which is essential for identifying complex DDoS attack patterns. Models like FlowTransformer show the importance of representing traffic in a way that preserves temporal context, allowing anomaly detection based on traffic dynamics. In addition, efficient segmentation maintains crucial information about standard and anomalous behaviors. This includes dividing data into temporal windows or packets that enable capturing recurring patterns and temporal dependencies, such as sudden changes in traffic volume or sequences of anomalous packets;
- Noise Reduction and Data Quality: Tokenization techniques that effectively aggregate information can help reduce noise in data. As a result, the model can focus on the most relevant features for anomaly detection, improving learning accuracy and effectiveness;
- Computational Efficiency and Complexity of the Model: Choosing a tokenization technique that results in a compact network traffic representation can significantly reduce the model size and inference time. This is relevant and important in high-demand scenarios, such as detecting DDoS attacks, where speed of response is critical. Models such as BERT and Tab Transformer demonstrate that optimized representations allow you to learn from less data and achieve better results, especially when detecting anomalies and DDoS attacks;
- Generalization and Flexibility: The way data is tokenized affects the model's ability to generalize from training data, making it more effective at detecting anomalies in real-world scenarios. Adaptive tokenization techniques, such as those used in BERT and STA-Tran, adjust segmentation to the specific characteristics of traffic data and different types of attacks, improving accuracy even in adverse conditions and with unbalanced data;
- Hyperparameter Tuning: Different tokenization techniques require adjustments to the model's hyperparameters, such as the learning rate and number of layers. Inadequate segmentation can lead to suboptimal learning, in which the model cannot generalize well to new data, resulting in high rates of false positives or negatives in attack detection;
- Performance Assessment: Systematically evaluating tokenization techniques is essential to identify the most appropriate approach. Models like FlowTransformer highlight the importance of precision, recall, and F1-score in performance analysis. For example, in [168], Tab Transformer achieved 98.35% in classifying regular traffic and 97.22% in detecting multiple classes of attacks, highlighting the impact of choices in data preparation.

Several papers contribute valuable insights to enhance these concepts. Ahmed et al. [14] propose an approach that represents traffic data as integer numeric values, eliminating the need for word embeddings and positional Encoders commonly used in NLP tasks. Removing these components allows the model to focus directly on critical traffic characteristics without the complexity of textual representations. By concentrating exclusively on essential data sequences, the model improves learning efficiency and enhances its ability to detect anomalies or DDoS attacks. Relying solely on raw numeric representations sets this method apart from approaches that depend on labeled data for training. In [17], examine the limitations of traditional Vision Transformer (ViT) tokenization, which segments input into fixed-length tokens. Disregarding the local data structure can hinder intrusion detection, where preserving contextual relationships is crucial. To address this issue, the authors introduce a sliding window mechanism for data segmentation. Maintaining edge information more effectively results in a richer and more informative representation. Their findings highlight how precise segmentation strengthens the model's ability to capture subtle patterns associated with DDoS attacks, improving anomaly detection accuracy. Integrating a sliding window with position encoding, based on relationships between anchor points and neighboring elements, further enhances ViT's adaptability. Greater flexibility in handling varying input sizes proves essential in scenarios where attack patterns continuously evolve.

The papers conducted in this survey demonstrate that tokenization and segmentation play a crucial role in the success of Transformer models in detecting DDoS attacks and anomalies in network traffic. Well-structured representations preserve critical information, optimize

model learning, and increase model adaptability to different scenarios. A judicious choice of these techniques improves attack detection and contributes to developing efficient and robust solutions against cyber threats.

RQ5: What is the impact of data normalization strategies on the overall performance of the Transformer when dealing with malicious traffic patterns? Data normalization plays a key role in optimizing the performance of Transformer models in detecting malicious traffic [51,52,151,164]. Researchers implement techniques such as z-score and min–max normalization to keep input variables uniformly, ensuring that the model consistently evaluates different network traffic characteristics [52]. This approach prevents any single metric from dominating the analysis, reducing bias in data interpretation. In [74] showed that normalization minimizes discrepancies between training and testing data distributions, strengthening the model's ability to generalize and detect anomalies accurately. In the paper proposed by [14] emphasized that this process also speeds up model convergence, as normalized data help Transformer recognize patterns more efficiently. In this way, the model reduces variance and improves its generalization capacity, preventing minor variations in benign traffic from making it difficult to identify cyber-attacks.

Normalization also minimizes noise and outliers, favoring extracting relevant features for classifying anomalous traffic. In [154] showed that this technique helps the model to focus on the most significant patterns, which becomes crucial when there is an overlap between benign and malicious traffic. Data normalization plays a crucial role in improving the performance of the Transformer model by speeding up convergence and minimizing the impact of outliers, which aids in detecting malicious traffic patterns like DDoS attacks. Additionally, it boosts the model's ability to generalize, enabling it to recognize new attack patterns while preserving the consistency of contextual relationships in the input data. Finally, by integrating with other techniques such as feature selection, normalization facilitates a cleaner and more efficient feature space, optimizing intrusion detection [152].

Enhancing detection accuracy requires effective preprocessing techniques, and normalization plays a crucial role in this process. It ensures the model can reliably differentiate between normal network behavior and potential attacks. Additionally, normalization strengthens the stability of the model, making it more adaptable to fluctuations in network traffic. In [166] found that this technique helps address class imbalance, allowing less frequent attacks to be properly detected. Integrating normalization with methods like SMOTE further refines data distribution, reducing biases in intrusion detection and improving overall system dependability. Transformers also benefit from normalization through more efficient attention mechanisms. In [12] emphasized that applying this method enhances how results are interpreted and improves model transparency, leading to more precise identification of malicious patterns. Despite its importance, some papers, such as those by [19,155], do not always directly focus on its influence in Transformer-based models. However, their research indicates that effective data preprocessing, particularly normalization, significantly boosts the accuracy of cyberattack detection. Incorporating normalization into the preprocessing pipeline is a key factor in improving the ability of Transformer models to detect malicious traffic efficiently.

RQ6: This RQ provides a comprehensive analysis of the papers with practical characteristics, aiming to highlight current trends and advances in DDoS detection research. The discussion is organized into three main categories: papers focused on real-time detection capabilities, reporting implementation in production environments, and papers that present innovative and relevant solutions through transformers for detecting DDoS attacks. By examining these contributions, the section seeks to contextualize the state of the art and identify promising directions for future research and practical applications. Thus, among the 45 selected papers analyzed in this research, 06 present real-time applications, 03 present implementation in real environments, and the remaining 36 were implemented in simulated environments.

Papers with real-time implementation: Tan et al. [69] propose ANID, a neural attention-based model for real-time network intrusion detection. Using time-slot-based traffic features, the model addresses a critical shortcoming of traditional approaches that depend on flow-level analysis and introduces delays by detecting threats only after a session ends. ANID, inspired by the Transformer architecture, simplifies its design by removing the encoder–decoder structure while retaining attention mechanisms to dynamically and efficiently capture traffic patterns. The model demonstrates superior performance to Bi-LSTM and Conditional Random Fields in terms of accuracy, recall, and false favorable rates. Furthermore, the absence of recurrent layers results in greater computational efficiency. Despite these advantages, the authors conduct their evaluations solely in simulated environments, leaving the model's effectiveness in real-world or production settings untested.

Building upon the importance of real-time detection highlighted by Tan et al. [69], Wu et al. [132] introduce RTIDS, a Transformer-based model designed to handle the increasing complexity and scale of modern network traffic. Unlike ANID, which simplifies the Transformer structure, RTIDS employs a complete encoder–decoder architecture with positional embeddings to extract temporal relationships and high-dimensional features. Through self-attention mechanisms, the model enhances its capacity to classify sophisticated and previously unseen attacks. RTIDS achieves impressive F1 scores of 99.17% and 98.48% on the CICIDS2017 and CIC-DDoS2019 datasets, outperforming other models like RNNs, LSTMs, and fuzzy neural networks. However, like ANID, RTIDS remains untested in live production environments, and its evaluation is limited to public datasets in controlled settings. The authors acknowledge the need for additional work to integrate the model into real-world systems and adapt it to dynamic network conditions.

Extending the discussion on adaptability and real-time responsiveness, Chen et al. [147] explore a reinforcement learning-based approach that overcomes the limitations of the Markov property typically associated with traditional methods. Their model dynamically adjusts detection accuracy and latency through a reward function, allowing for more responsive and context-aware intrusion detection. The system is validated on realistic public datasets such as UNSW-NB15 and shows superior detection precision and faster response times than conventional baselines. Although Chen et al. go a step further by simulating operational conditions, their model (like those of Tan et al. [69], Wu et al. [132]) has not been deployed in actual production networks. They recognize this limitation and propose that future deployments may require hardware acceleration to ensure scalability and performance. These papers emphasize significant advancements in real-time and intelligent DDoS detection and underscore the pressing need for validation in real-world environments.

Casajús et al. [160] propose a transformer-based intrusion detection system tailored for Industrial Internet of Things (IIoT) networks, leveraging the sequential learning strengths of transformers to detect anomalies with high precision. Their model achieves strong results on the WUSTL-IIoT-2021 dataset, approximating real-world IIoT traffic and enhancing interpretability, an essential attribute for protecting critical infrastructure. Although their approach is designed for real-time traffic analysis and shows promise for practical deployment, it has not yet been tested in a live industrial or production environment. The authors acknowledge this limitation and recommend future deployment in active IIoT infrastructures to validate performance under real operational constraints.

In a similar direction, Alkhatib et al. [165] present CAN-BERT, a BERT-based intrusion detection model developed for Controller Area Networks (CAN) in modern vehicles. Like [160], they focus on domain-specific network environments and emphasize real-time detection capabilities. CAN-BERT applies masked language modeling to identify anomalies in CAN ID sequences without supervision, achieving F1 scores between 0.81 and 0.99 and maintaining low inference latency. However, despite the model's suitability for embedded deployment, it remains untested in actual automotive systems. The authors suggest

future integration into Electronic Control Units (ECUs), indicating a shared challenge with Casajús et al. in bridging the gap between simulation and real-world deployment.

Extending this discussion to a broader network context, Luo et al. [117] propose a hierarchical intrusion detection system that combines CNNs with Transformers to enhance anomaly detection accuracy in network traffic. Their model incorporates attention-based soft feature selection to improve precision and reduce false positives, achieving competitive results on the UNSW-NB15 dataset. While Luo et al. [117] align with the previous papers in targeting real-time detection and demonstrating promising experimental outcomes, their validation is likewise confined to controlled settings. They outline plans to test the model on more diverse datasets but, like Casajús and Alkhatib, have yet to implement or evaluate the system in live production environments.

These papers demonstrate the growing application of Transformer-based models in specialized network contexts ranging from industrial systems to automotive and general network environments. Although each approach contributes valuable advancements in accuracy, interpretability, and real-time performance, they share a standard limitation: the absence of real-world deployment. Future research must prioritize transitioning these models from experimental environments to operational infrastructures to validate their practical impact.

Papers with production application:

Wang et al. [150] introduced TransIDS, a Transformer-based intrusion detection system designed to meet the growing security demands of IoT environments. To overcome the limitations of traditional methods in detecting novel and sophisticated attacks, the authors eliminated the need for complex feature engineering. Instead, they leveraged multi-head self-attention to extract high-dimensional temporal features directly from raw traffic. They also incorporated label smoothing to address data imbalance, enhancing the model's generalization capability. Evaluated on the TON-IoT dataset (crafted to mimic realistic IoT conditions), TransIDS outperformed state-of-the-art baselines in accuracy and efficiency. Using a dataset that reflects real-world traffic patterns, the study positioned the system closer to production readiness, highlighting its suitability for real-time deployment in dynamic and resource-constrained IoT networks.

Building on the theme of practical applicability, Wang et al. [156] proposed a novel intrusion detection approach for Industrial IoT (IIoT) systems based on microcontroller unit (MCU) temperature fingerprints. They observed that increased program complexity raises the MCU temperature, which can act as a behavioral signal of anomalies. The authors designed a method for collecting temperature sequences, calculating residuals, and training an autoencoder-based model for anomaly detection. They implemented this on a Raspberry Pi 4B and achieved 89% detection accuracy. Although they conducted experiments on physical hardware, they did not test the model in production-level IIoT settings. Nonetheless, like TransIDS, this paper laid a strong foundation for practical deployment by validating the model under realistic testbed conditions and suggesting directions for scalability, such as edge computing integration.

Expanding the scope of Transformer applications, Luo et al. [109] developed a two-stage Transformer-based framework for identifying IoT device types in heterogeneous network environments. In the first stage, the model filters out malicious traffic, ensuring that only benign data is passed to the second Transformer, which performs precise device classification. The authors improved classification stability and accuracy by using an ensemble algorithm to aggregate predictions. Their evaluation of the N-BaIoT dataset (comprising real-world traffic and botnet attacks such as Mirai and BASHLITE) demonstrated superior performance compared to traditional models. However, the model's reliance on labeled data limits its scalability, as real-world environments rarely offer the ground truth necessary for supervised training. Like the previous papers, this study showcases promising results in controlled environments but highlights the challenges of deploying such solutions at scale in real-world networks.

These papers illustrate the growing momentum behind transformer-based approaches to secure IoT and IIoT systems. While each model introduces unique innovations (from raw traffic analysis to behavioral fingerprinting and device classification), they all underscore the need for further validation in production environments. Bridging this gap between research and deployment remains a key challenge and a critical focus for future paper in real-world cybersecurity applications.

Papers with testbed scenarios: A growing body of research has explored Transformer-based architectures for intrusion detection in IoT and related environments, with promising but largely simulation-bound results. Wang et al. [12] proposed the Res-TranBiLSTM model, which integrates ResNet for spatial feature extraction and combines Transformer modules with BiLSTM to capture temporal dependencies in IoT traffic. By applying SMOTE and ENN to address data imbalance, the authors achieved high accuracy rates (up to 99.56%) on public datasets such as NSL-KDD, CIC-IDS2017, and MQTTset. However, the experiments were conducted in controlled, simulated environments, and the authors emphasized the need for future validation using real IoT traffic to ensure the model's robustness under realistic conditions.

Echoing these concerns, Ahmed et al. [14] introduced the Modified Transformer Neural Network (MTNN), which employs an attention-based architecture to improve adaptability in IoT intrusion detection. Despite outperforming baseline models like RNN and LSTM (with significant gains in accuracy and recall), the study also remained within simulated environments, relying on the ToN_IoT dataset to approximate real-world scenarios. Although MTNN showed strong performance with fewer parameters, the authors acknowledged the absence of real-world deployment, underscoring the broader challenge of translating high-performing models from lab to production.

While both Res-TranBiLSTM and MTNN address generic IoT threats, Alrahmani et al. [74] targeted a specific attack vector (DDoS prediction) using FEDformer and PatchTST. Their results confirmed the superiority of PatchTST in capturing time-series patterns, yielding lower validation loss in controlled experiments. However, the authors did not implement the models in live network environments; instead, they evaluated them using publicly available datasets. This mirrors the limitations noted by Wang et al. and Ahmed et al. suggesting that while Transformer-based models hold considerable potential for cyber defense, practical validation remains a common gap across papers.

Expanding the application domain, Nguyen et al. [149] applied Transformer Attention Networks (TAN) to secure the Controller Area Network (CAN) in vehicles in a uniquely constrained and security-sensitive environment. Their model achieved high detection accuracy and demonstrated the ability to detect complex attacks like replays without labeled data. Yet, like the previously mentioned papers, the experiments were confined to pre-collected datasets. The authors acknowledged the challenge of applying TAN in real-time vehicular contexts, pointing to transfer learning as a future direction to enhance adaptability.

Zhang et al. [151] extended the use of Transformers to Software-Defined Networking (SDN), proposing RLFAT to detect relay link forged attacks. Their method leverages the self-attention mechanism to extract high-dimensional relationships in traffic data received by SDN controllers. Although RLFAT outperformed traditional deep learning models in experimental settings, it, too, was evaluated in a simulated environment. The authors recognized the risks of overfitting and the need for future deployment in operational SDN infrastructures.

Finally, Lan et al. [153] addressed the issue of data imbalance in intrusion detection by combining a decision tree classifier with a Transformer-based model (FT-Transformer). This cascaded approach yielded strong performance, particularly in detecting rare attack classes on the CIC-IDS2017 dataset. Nevertheless, like other papers, the model was not evaluated in production networks. The authors highlighted that while public datasets are valuable for benchmarking, they fail to capture live environments' full complexity and unpredictability.

Recent papers have demonstrated significant progress in leveraging Transformer-based models for intrusion detection, each proposing distinct strategies to enhance the learning of temporal and spatial features in network environments. Han et al. [155] introduced GTID, a novel model that combines n-gram frequency analysis with a time-aware Transformer architecture. This approach addresses the limitations of traditional detection techniques that often lose valuable information during feature extraction. GTID captures contextual and temporal dependencies more effectively by hierarchically learning traffic features at both the packet and session levels and separately analyzing headers and payloads. Although the authors achieved strong performance on benchmark datasets such as ISCX2012 and CICIDS2017, they only evaluated the model in controlled settings. While they acknowledged the importance of real-world applicability and simulated benign traffic conditions to estimate GTID's robustness, they left full deployment in operational networks as future work.

This gap between experimental success and real-world deployment also appears in the models proposed by [16,139], namely DDoS-MSCT and DDosTC. Both models integrate Transformer architectures with convolutional neural networks to capture both local and global traffic features, targeting the complex characteristics of DDoS attacks. They reported outstanding results (exceeding 99.9% in accuracy, precision, recall, and F1-score) when tested on benchmark datasets such as CIC-DDoS2019 and CIC-IDS2017. However, these papers also restricted their evaluations to offline, simulated environments. They did not provide evidence of integration into real-time monitoring systems or validation under real network conditions.

Similarly, Salam et al. [75] investigated deep learning techniques (including CNNs, RNNs, and Transformer-based models) for detecting web-based attacks in Industry 5.0. Their experiments showed that Transformer models consistently delivered the best performance across benchmark datasets like KDD Cup 1999 and CICIDS2017. Yet, the authors acknowledged the limitations of relying solely on static datasets, which fail to capture the complexity and unpredictability of live industrial systems. They highlighted the need for future research to bridge the gap between theoretical performance and operational reliability. These papers reflect a growing trend toward adopting attention-based models to address the challenges of detecting increasingly complex and dynamic DDoS attacks.

When reviewing the literature for this research on DDoS detection using Transformer-based models, although this paper found papers that address low-rate attacks (such as: [17,53,118,152,153]), this paper did not find papers that addressed more stealthy and modern threats, such as Shrew attacks [173] or dictionary emulation attacks [174]. At the time of the research, datasets that included these scenarios were also not found, limiting comparative analyses in this context. The absence of Transformer-based approaches targeting these threats reveals a critical gap in the current literature and highlights a promising direction for future research. By integrating anomaly detection techniques capable of identifying subtle contextual deviations in network traffic, researchers can increase the effectiveness of Transformer models against these sophisticated attacks.

RQ7: What are the research gaps in the existing literature? The Open Issues in 6 section presents a broad discussion of the possible open questions that this study raises for the reader to answer this question.

6. Open issues and research directions

Transformer models for DDoS attack detection represent a novel and effective approach for identifying malicious activities in intricate network systems [4,64]. These models are particularly notable for its capacity to process large volumes of traffic and swiftly identify unusual patterns linked to DDoS attacks, enabling an accurate distinction between legitimate and malicious traffic [18,120,175]. This section explores opportunities and key challenges associated with employing Transformer in the field of cybersecurity. Analyzing these challenges offers valuable insights into the existing gaps and obstacles, guiding future research to enhance the detection of DDoS attacks in complex and dynamic environments.

6.1. Issues in transformer models

The customization of attention layers is essential to capture complex patterns and long-range relationships in network data. Models such as Attention-Enhanced Transformer (AET) for example, show improvements in enhanced attention, but adaptation to different types of traffic and network contexts still requires investigation, mainly in environments with highly varied and evasive data traffic. Researchers have not yet found an efficient solution for detecting actions from multiple traffic sources that may behave similarly to legitimate traffic [54]. In addition, the integration of hybrid techniques, such as the combination of CNN and Transformer in models such as Hybrid Transformer (HBT), brings to light the challenge of balancing efficiency and accuracy in the extraction of spatial and temporal features, which can improve the detection of DDoS attacks.

Another relevant point is the need for approaches that guarantee scalability and real-time applicability, as seen in the use of Real-Time Processing Transformer (RPT) for industrial networks. Continuous analysis of temporal data is critical for intrusion detection, but there are limitations in real-time response capabilities, mainly in large-scale, high-traffic networks such as the IoT. Furthermore, the use of semi-supervised and unsupervised methods, as in some hybrid variants, highlights the importance of model adaptability to scenarios where data labeling is sparse or inconsistent, highlighting the need for methods that adapt to data streams, heterogeneous data and invalid values.

Areas such as NLP, computer vision, general cybersecurity, real-time network traffic analysis, IoT, adaptive intrusion detection systems, computational biology, drug discovery, economics and finance, robotics, and automation control, among others, present opportunities to standardize variants of the Transformer model [69,148,152]. This standardization and adjustment of the Transformer architecture variants benefits researchers and practitioners by simplifying the understanding of the specificities and potential of the Transformer model in different domains. By organizing and classifying knowledge about this technology, it is possible to create more accurate solutions and reduce false positives, exploiting specific layers of the Transformer to optimize the detection of DDoS attacks.

A research opportunity arises from exploring and developing unsupervised or semi-supervised learning methods for detecting DDoS attacks in real-time. Currently, the majority of papers follow supervised learning, which implies a significant dependence on labeled data. However, labeled data is not always readily available in cybersecurity scenarios, mainly in real-time attacks where speed and adaptability are essential. Therefore, a promising line of research would involve investigating unsupervised and semi-supervised learning techniques and exploring the use of Transformer models (such as Encoder–Decoder and other custom models) for detecting DDoS attacks in unlabeled data scenarios or partially labeled. This would pave the way for more autonomous and agile defense systems that are able to identify attack patterns in real-time without the need for human intervention for labeling.

6.2. Issues in transformer architectures

The open questions related to Transformer architectures are organized into four categories: encoder/decoder, real-time, unlabeled data, and others.

Encoder/Decoder — Includes transformer models that use only the Encoder or the Encoder and the Decoder, processing labeled data without real-time application. Detecting DDoS attacks with Transformer using continuous and autonomous learning without previously labeled data is a significant challenge, mainly for novel attacks such as zero-day attacks. Because it is necessary to address the cost reduction associated with maintaining data labeling manually, which requires specialists for this task. Therefore, ensuring the efficiency and long-term resilience

of systems, considering the lowest downtime and evolution of attack tactics, is a challenge.

Detecting multi-class DDoS attacks in complex scenarios using the Transformer Model is an open question in security. Since it is essential to explore models to identify multiple categories of DDoS attacks accurately, this approach should offer specific responses to each type of threat other than DDoS attacks. Therefore, the efficiency and effectiveness of the detection strategy increase. Across these open opportunities, others stand out as critical network infrastructure in AS, SDN, and industrial environments. In addition to reducing false positives, increasing confidence in detection systems can also reduce complications considered in network management.

Addressing resource constraints such as bandwidth, computing power, and bandwidth consumption are recent challenges. Opportunities include adapting transformer models to IoT devices and edge computing environments and dealing with resource constraints such as bandwidth, computational power, and energy consumption. The quest to protect IoT devices in critical systems, such as connected healthcare networks, smart home devices, and industrial automation environments, requires this type of security guarantee against DDoS attacks. Furthermore, the generalization of the Transformer model allows greater scalability and operates in distributed systems. Therefore, detecting attacks directly on devices is possible, reducing server dependence. With this, there is the possibility of increasing resilience in critical infrastructure and minimizing DDoS attacks on remote and distributed systems.

Another open issue in detecting Transformer-based attacks centers on incorporating multiple data sources, such as encrypted traffic, metadata, and system logs. Integrating these data sources can provide a holistic view of attacks, increasing accuracy by combining complementary information from different sources. The process should become relevant in detecting sophisticated attacks, such as encrypted DDoS, since packet traffic analysis alone may be insufficient. Furthermore, the need to incorporate multiple data sources strengthens layered defense, allowing the identification of more complex attack vectors that exploit different flaws in network components. Thus, this represents an ongoing challenge for management in detecting DDoS attacks.

Exploring scenarios with new threats that arise is unknown; the system needs to be able to identify them without prior classification or mistakenly as regular traffic and in real-time, which is an open question in the literature. In this context, it is relevant because cyberattacks constantly evolve, and new attack vectors often challenge traditional systems. DDoS attack detection systems utilizing the Transformer model must be capable of generalizing to identify strange behavior in the network. Hence, Transformer models must be able to protect complex networks, such as 5G networks, IoT, and edge computing environments, where the dynamics and diversity of data increase the challenge of distinguishing legitimate from malicious traffic.

Significant advancements can be achieved by integrating Transformer with Federated Learning, enabling decentralized training without compromising data privacy. This approach addresses critical barriers in cybersecurity while facilitating more reliable detection capabilities. This integration provides a more comprehensive and accurate analysis, overcoming the limitations of traditional traffic-only methods by offering a deeper understanding of network behavior.

The paper proposed by [155] notes that the nature of DDoS attacks, which typically involve multiple hosts and generate multiple sessions, makes it more challenging to distinguish between regular traffic and DDoS traffic in a single session. This limitation affects all session-based intrusion detection methods. The possibility of a Transformer model to provide clear and understandable information with the aim of correct interpretation when using different models to detect DDoS attacks is a topic that requires attention when using TA. In this way, the possibility of decision-making through a system with clear and well-defined purposes increases confidence in using the Transformer model, facilitating auditing and validations in various actions to detect DDoS attacks. These challenges highlight promising areas for further research.

Real-Time: This category includes models for processing labeled data in real-time, regardless of the type of Transformer model. Real-time traffic and processing networks are critical in detecting DDoS attacks, as detection at runtime minimizes downtime for both applications and the network itself. Therefore, TA must optimizer to achieve fast inferences, especially in DDoS attacks. Another opportunity is integrating unsupervised methods like clustering into TA to detect and identify anomalous behavior patterns like DDoS without relying on labeled data. Detecting attacks in real-time is essential, and it is an important open issue.

Model generalization is one of the challenges identified in detecting DDoS attacks through TA. The importance is due to making real-time attack detections without labeled data and integrating unsupervised ML methods. This approach can reduce computational costs, improving the performance of transformer models and the efficiency in detecting and reducing downtime in DDoS attack detection. One option for detecting low-rate DDoS attacks could be to train Transformer models on datasets with different patterns of low-rate malicious traffic. This way, the model can learn to identify subtle variations in packet sending rates or request intervals, as detection systems usually miss and fail to detect these behaviors. Furthermore, incorporating attention mechanisms focused on long temporal sequences could increase the effectiveness of identifying these behaviors.

Analyzing encrypted traffic without deep packet inspection (DPI-Free) is crucial in today's context, where encryption predominates. The transformer can detect anomalies using metadata alone, balancing security and privacy requirements. The dynamic construction of attention mechanisms is one of the innovations in customizing the Transformer model. An open task is to Dynamically adjust the transformer model based on traffic changes or variations in DDoS attack patterns in real-time. Systems incorporating multi-level temporal and hierarchical attention in DDoS attack detection can correlate suspicious events at different network levels, improving accuracy and reducing false positives in attack detection.

Another open issue suggests better combining federated learning to detect anomalies in encrypted traffic without network packet inspection, especially in detecting DDoS attacks in real-time. By exploring these approaches, the chances of efficiently detecting DDoS attacks in any network topology and in real-time increase. Another challenge when detecting DDoS attacks in real-time is optimizing and reducing processing latency. Integrating models with smart network devices such as Smart NICs and automated response systems is an opportunity to take advantage of these new devices' capabilities and contribute to a new systems model for intelligent detection of DDoS attacks.

However, building environments that work with data in real time and/or in real environments is a challenge. Therefore, according to several papers conducted by different authors (such as [176–179]), they highlight that implementing a real-time solution based on Transformer models for DDoS attack detection requires a careful analysis of specific computational and network resources. Adequate computational capacity, such as servers equipped with GPUs or TPUs, is essential to accelerate model inference, especially in high-speed network environments. Sufficient memory resources are also required to store model parameters and process large volumes of network traffic data without delays. In addition, a network infrastructure that supports low-latency operations is essential to ensure that data collection, processing, and event responses occur within acceptable timeframes. To further optimize performance and reduce the load on central servers, it is suggested to deploy edge computing solutions, including smart network cards, smart switches, and routers with integrated computational resources.

Despite the potential benefits, deploying such solutions can introduce a slight processing overhead and a small increase in latency, especially when inference is performed centrally. However, a distributed architecture, in which lightweight detection agents are deployed on

edge devices, can significantly minimize this impact, enabling real-time detection without degrading overall network performance. Finding a balance between model accuracy and computational cost remains essential; lightweight Transformer models or optimized variants offer a viable path to efficient real-time detection. Therefore, a well-designed architecture that incorporates real-time inference optimizations and effective resource management strategies enable accurate detection of DDoS attacks while keeping minimal impact on network performance.

Unlabeled Data — includes models that do not rely on labeled data and do not process information in real-time. Researchers are optimizing Transformer models to increase their effectiveness and adaptability in detecting DDoS attacks. Then, there is a need to understand how and why the Transformer model classifies a traffic pattern as malicious. There must be a detailed explanation of the application of the model in detecting these attacks. It will undoubtedly allow analysts to understand the factors influencing each diagnosis, facilitating decision-making in an incident. An open issue is integrating multimodal data to increase the effectiveness of Transformer models in detecting DDoS attacks. Combining different data sources, such as network traffic logs and metrics, gives the applied models a global view of network activity. However, the challenge lies in using multimodal data when the labeled data set is not available or does not exist. Then, how to train Transformer models to identify relevant patterns in detecting DDoS attacks with these restrictions is the main challenge.

Others — Includes applications that do not fit into the previous categories and classified as generic. Continuously adapting Transformer models to detect new types of DDoS attacks, using the identification of unknown patterns and the ability to update dynamically without the need to restart full training, is a promising approach in ML and DL. Organizations face significant challenges in building systems capable of sensing across different network environments, such as 5G, industrial networks, and networks utilizing emerging devices, including IoT and Smart NICs in servers. These applications are still underexplored in real networks. Furthermore, companies generally do not adopt a testing culture and do not provide resources or infrastructure to implement and test these systems in real environments, which makes experiments and tests in these scenarios particularly difficult to carry out.

7. Conclusion

This survey focused on exploring scientific approaches from the literature that employ Transformer models for Distributed Denial of Service (DDoS) attack detection. Transformer models are neural network architectures based on deep learning. Transformer models are incredibly versatile. When applied to cybersecurity, they can assist in DDoS attack detection. This work classified the existing approach into different variants of the Transformer model. Each variant indicates which approach can make the application accurate and reduce the trial and error effort in implementation. Each variant indicates which approach can make the application accurate and reduce trial-and-error effort in implementation. Furthermore, the research discusses various ML and DL techniques applied to attack detection, as they are employed in different variants of the Transformer Architecture.

Depending on the context, models can be applied to just the Encoder area, just the Decoder, or both, or in new implementations that change the original structure of the Transformer architecture. The survey also covers data preprocessing techniques to improve deep learning with Transformer. Approximately 88.89% of the examined papers rely on labeled data, which requires several processing tasks and significant computational costs. These factors directly impact the performance, efficiency, and accuracy of DDoS attack detection, mainly in real-time applications. These data demonstrate that the choice of techniques directly influences the models' ability to process and learn relevant network traffic patterns, affecting their accuracy and efficiency.

It is worth emphasizing the factor determining the choice of tokenization is the process of dividing the text into smaller units called tokens. Depending on the type of tokenization used, these tokens can be words, subwords, characters, or even sentences. In the context of Transformer models, tokens are the minimum unit the model can process, and data segmentation techniques include representation and temporal context, noise reduction, data quality, computational efficiency, generalization, flexibility, hyperparameter adjustment, and performance evaluation. Despite the advances, this survey also highlights open issues. Among them, the customization of attention layers to identify complex patterns in network traffic is essential. The detection of DDoS attacks can occur directly through classifiers, by analyzing traffic behavior, or by other models that assess the possibility of an attack. Finally, this survey reinforces the potential of the Transformer architecture and its versatility, as it was possible to highlight the ability to integrate different Machine Learning and Deep Learning techniques by answering RQ3. All layers of the architecture are modular and highly customizable. This customization allows you to modify components such as attention mechanisms, feed-forward networks, and normalization strategies to optimize performance. Further, Transformer areas, such as the Encoder, Decoder, and attention, are flexible to adjust, remove, or be combined, ensuring the model adapts to different contexts, from natural language processing to the detection of cyberattacks.

CRediT authorship contribution statement

Euclides Peres Farias: Writing – review & editing, Writing – original draft, Visualization, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Anderson Bergamini de Neira:** Writing – review & editing, Visualization, Validation, Resources, Investigation, Conceptualization. **Ligia Fracielle Borges:** Writing – review & editing, Validation, Supervision, Methodology, Investigation. **Michele Nogueira:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Euclides Peres Farias Junior reports was provided by Federal University of Parana. Anderson Bergamini de Neira reports financial support was provided by State of Sao Paulo Research Foundation. Ligia Francielle Borges reports financial support was provided by State of Sao Paulo Research Foundation. Euclides Peres Farias Junior reports a relationship with Federal Technological University of Paraná that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and by São Paulo Research Foundation (FAPESP), grants #2018/23098-0, #2022/06840-0 and #2025/00612-3. We also thank the Federal Technological University of Paran/'a (UTFPR–SH) – Santa Helena Campus for its institutional support.

Data availability

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

References

[1] S. Bhatia, S. Behal, I. Ahmed, Distributed denial of service attacks and defense mechanisms: current landscape and future directions, Versatile Cybersecur. (2018) 55–97.

[2] F. Duarte, EXPLODING TOPICS: amount of data created daily (2024), 2024, https://explodingtopics.com/blog/data-generated-per-day.

[3] J. Mirkovic, P. Reiher, A taxonomy of DDoS attack and DDoS defense mechanisms, ACM SIGCOMM Comput. Commun. Rev. 34 (2) (2004) 39–53.

[4] L. Mohammadpour, T.C. Ling, C.S. Liew, A. Aryanfar, A survey of CNN-based network intrusion detection, Appl. Sci. 12 (16) (2022) 8162.

[5] W. Jia, Y. Liu, Y. Liu, J. Wang, Detection mechanism against DDoS attacks based on convolutional neural network in Sinet, in: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, ITNEC, vol. 1, IEEE, 2020, pp. 1144–1148.

[6] M. Najafimehr, S. Zarifzadeh, S. Mostafavi, DDoS attacks and machine-learning-based detection methods: A survey and taxonomy, Eng. Rep. (2023) e12697.

[7] A. Singh, B.B. Gupta, Distributed denial-of-service (DDoS) attacks and defense mechanisms in various web-enabled computing platforms: Issues, challenges, and future research directions, Int. J. Semant. Web Inf. Syst. IJSWIS 18 (1) (2022) 1–43.

[8] Y. LeCun, The next generation of artificial intelligence, 2020, https://www.forbes.com/sites/robtoews/2020/10/12/the-next-generation-of-artificial-intelligence/?sh=4c85531b59eb.

[9] N. Aslam, S. Srivastava, M. Gore, A comprehensive analysis of machine learning- and deep dearning-based solutions for DDoS attack detection in SDN, Arab. J. Sci. Eng. 49 (3) (2024) 3533–3573.

[10] R. Toews, The next generation of artificial intelligence, 2020, https://www.forbes.com/sites/robtoews/2020/10/12/the-next-generation-of-artificial-intelligence/?sh=434c76da59eb.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[12] S. Wang, W. Xu, Y. Liu, Res-TranBiLSTM: An intelligent approach for intrusion detection in the internet of things, Comput. Netw. 235 (2023) 109982.

[13] U. Ünal, H. Dağ, Anomalyadapters: Parameter-efficient multi-anomaly task detection, IEEE Access 10 (2022) 5635–5646.

[14] S.W. Ahmed, F. Kientz, R. Kashef, A modified transformer neural network (MTNN) for robust intrusion detection in IoT networks, in: 2023 International Telecommunications Conference, ITC-Egypt, IEEE, 2023, pp. 663–668.

[15] S. Ruiz-Villafranca, J. Roldán-Gómez, J.M.C. Gómez, J. Carrillo-Mondéjar, J.L. Martinez, A TabPFN-based intrusion detection system for the industrial internet of things, J. Supercomput. 80 (14) (2024) 20080–20117.

[16] H. Wang, W. Li, DDosTC: A transformer-based network attack detection hybrid mechanism in SDN, Sensors 21 (15) (2021) 5047.

[17] Y.-G. Yang, H.-M. Fu, S. Gao, Y.-H. Zhou, W.-M. Shi, Intrusion detection: A model based on the improved vision transformer, Trans. Emerg. Telecommun. Technol. 33 (9) (2022) e4522.

[18] M. Landauer, S. Onder, F. Skopik, M. Wurzenberger, Deep learning for anomaly detection in log data: A survey, Mach. Learn. Appl. 12 (2023) 100470, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666827023000233.

[19] T.A. Nguyen, M. Park, Doh tunneling detection system for enterprise network using deep learning technique, Appl. Sci. 12 (5) (2022) 2416.

[20] N. Abdi, A. Albaseer, M. Abdallah, The role of deep learning in advancing proactive cybersecurity measures for smart grid networks: A survey, IEEE Internet Things J. (2024).

[21] M. Almehdhar, A. Albaseer, M.A. Khan, M. Abdallah, H. Menouar, S. Al-Kuwari, A. Al-Fuqaha, Deep learning in the fast lane: A survey on advanced intrusion detection systems for intelligent vehicle networks, IEEE Open J. Veh. Technol. (2024).

[22] S. AboulEla, N. Ibrahim, S. Shehmir, A. Yadav, R. Kashef, Navigating the cyber threat landscape: An in-depth analysis of attack detection within IoT ecosystems, AI 5 (2) (2024) 704–732.

[23] D. Attique, W. Hao, W. Ping, D. Javeed, P. Kumar, Explainable and data-efficient deep learning for enhanced attack detection in IIoT ecosystem, IEEE Internet Things J. 11 (24) (2024) 38976–38986.

[24] S.M. Specht, R.B. Lee, Distributed denial of service: Taxonomies of attacks, tools, and countermeasures, in: PDCS, 2004, pp. 543–550.

[25] S. Specht, R. Lee, Taxonomies of distributed denial of service networks, attacks, tools and countermeasures, CEL2003- 03, Princet. Univ. Princet. NJ, USA (2003).

[26] A. Gupta, S. Rajvanshi, M. Kaur, Impact of cybercrime on E-governance: confidentiality of government data affected by the cybercrime, in: Artificial Intelligence and Information Technologies, CRC Press, 2025, pp. 353–359.

[27] S. Lysenko, K. Bobrovnikova, S. Matiukh, I. Hurman, O. Savenko, Detection of the botnets' low-rate DDoS attacks based on self-similarity, Int. J. Electr. Comput. Eng. 10 (4) (2020) 3651–3659.

[28] X. Yang, F. Zhu, X. Yang, J. Luo, X. Yi, J. Ning, X. Huang, Secure reputation-based authentication with malicious detection in VANETs, IEEE Trans. Dependable Secur. Comput. (2024).

[29] A.V. Vu, D.R. Thomas, B. Collier, A. Hutchings, R. Clayton, R. Anderson, Getting bored of cyberwar: exploring the role of low-level cybercrime actors in the Russia–Ukraine conflict, in: Proceedings of the ACM on Web Conference 2024, 2024, pp. 1596–1607.

[30] M. Kim, H. Na, K. Chae, H. Bang, J. Na, A combined data mining approach for DDoS attack detection, in: Information Networking. Networking Technologies for Broadband and Mobile Networks: International Conference ICOIN 2004, Busan, Korea, February 18-20, 2004. Revised Selected Papers, Springer, 2004, pp. 943–950.

[31] R. Vishwakarma, A.K. Jain, A survey of DDoS attacking techniques and defence mechanisms in the IoT network, Telecommun. Syst. 73 (1) (2020) 3–25.

[32] P. Szynkiewicz, Signature-based detection of botnet DDoS attacks, in: Cybersecurity of Digital Service Chains: Challenges, Methodologies, and Tools, Springer, 2022, pp. 120–135.

[33] M. Alenezi, K. Almustafa, K.A. Meerja, Cloud based SDN and NFV architectures for IoT infrastructure, Egypt. Inform. J. 20 (1) (2019) 1–10.

[34] R. Muñoz, R. Vilalta, N. Yoshikane, R. Casellas, R. Martínez, T. Tsuritani, I. Morita, Integration of IoT, transport SDN, and edge/cloud computing for dynamic distribution of IoT analytics and efficient use of network resources, J. Lightwave Technol. 36 (7) (2018) 1420–1428.

[35] Z. Lv, W. Xiu, Interaction of edge-cloud computing based on SDN and NFV for next generation IoT, IEEE Internet Things J. 7 (7) (2019) 5706–5712.

[36] S. Dong, K. Abbas, R. Jain, A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments, IEEE Access 7 (2019) 80813–80828.

[37] M. De Donno, N. Dragoni, A. Giaretta, A. Spognardi, DDoS-capable IoT malwares: Comparative analysis and mirai investigation, Secur. Commun. Netw. 2018 (1) (2018) 7178164.

[38] B. Bala, S. Behal, AI techniques for IoT-based DDoS attack detection: Taxonomies, comprehensive review and research challenges, Comput. Sci. Rev. 52 (2024) 100631.

[39] L. Sana, M.M. Nazir, J. Yang, L. Hussain, Y.-L. Chen, C.S. Ku, M. Alatiyyah, L.Y. Por, Securing the IoT cyber environment: Enhancing intrusion anomaly detection with vision transformers, IEEE Access (2024).

[40] A. Diaf, A.A. Korba, N.E. Karabadji, Y. Ghamri-Doudane, BARTPredict: Empowering IoT security with LLM-Driven cyber threat prediction, 2025, arXiv preprint arXiv:2501.01664.

[41] E.M. Rudd, A. Rozsa, M. Günther, T.E. Boult, A survey of stealth malware attacks, mitigation measures, and steps toward autonomous open world solutions, IEEE Commun. Surv. Tutor. 19 (2) (2017) 1145–1172.

[42] N. Mishra, S. Pandya, Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review, IEEE Access 9 (2021) 59353–59377.

[43] M. Wazzan, D. Algazzawi, O. Bamasaq, A. Albeshri, L. Cheng, Internet of things botnet detection approaches: Analysis and recommendations for future research, Appl. Sci. 11 (12) (2021) 5713.

[44] N. Pandey, P.K. Mishra, Taxonomy of DDoS attacks and their defense mechanisms in IoT, J. Sci. Res. 65 (2021) 197–207.

[45] P. Viktor, M. Fodor, Examining internet of things (IoT) devices: A comprehensive analysis, in: 2024 IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics, SAMI, IEEE, 2024, pp. 000115–000120.

[46] J. Xie, F.R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, Y. Liu, A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges, IEEE Commun. Surv. Tutor. 21 (1) (2018) 393–430.

[47] N. Ashodia, K. Makadiya, Detection and mitigation of DDoS attack in software defined networking: A survey, in: 2022 International Conference on Sustainable Computing and Data Communication Systems, ICSCDS, IEEE, 2022, pp. 1175–1180.

[48] I.A. Valdovinos, J.A. Pérez-Díaz, K.-K.R. Choo, J.F. Botero, Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions, J. Netw. Comput. Appl. 187 (2021) 103093.

[49] G.A. Jaafar, S.M. Abdullah, S. Ismail, et al., Review of recent detection methods for HTTP DDoS attack, J. Comput. Netw. Commun. 2019 (2019).

[50] Y. Xing, H. Shu, H. Zhao, D. Li, L. Guo, Survey on botnet detection techniques: Classification, methods, and evaluation, Math. Probl. Eng. 2021 (2021) 1–24.

[51] S.Y. Diaba, T. Anafo, L.A. Tetteh, M.A. Oyibo, A.A. Alola, M. Shafie-Khah, M. Elmusrati, SCADA securing system using deep learning to prevent cyber infiltration, Neural Netw. 165 (2023) 321–332.

[52] Z. Long, H. Yan, G. Shen, X. Zhang, H. He, L. Cheng, A transformer-based network intrusion detection approach for cloud security, J. Cloud Comput. 13 (1) (2024) 5.

[53] A.B. Bhutto, X.S. Vu, E. Elmroth, W.P. Tay, M. Bhuyan, Reinforced transformer learning for VSI-DDoS detection in edge clouds, IEEE Access 10 (2022) 94677–94690.

[54] Z. Zhang, L. Wang, An efficient intrusion detection model based on convolutional neural network and transformer, in: 2021 Ninth International Conference on Advanced Cloud and Big Data, CBD, IEEE, 2022, pp. 248–254.

[55] S. Mishra, P.S. Chatterjee, A systematic survey on DDoS attack and data confidentiality issue on cloud servers, in: 2021 19th OITS International Conference on Information Technology, OCIT, IEEE, 2021, pp. 273–278.

[56] N. Hoque, D.K. Bhattacharyya, J.K. Kalita, Botnet in DDoS attacks: Trends and challenges, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2242–2270.

[57] S. Sadeghpour, N. Vlajic, Click fraud in digital advertising: A comprehensive survey, Computers 10 (12) (2021) 164.

[58] A. Borys, A. Kamruzzaman, H.N. Thakur, J.C. Brickley, M.L. Ali, K. Thakur, An evaluation of IoT DDoS cryptojacking malware and mirai botnet, in: 2022 IEEE World AI IoT Congress, AIIoT, IEEE, 2022, pp. 725–729.

[59] D. Stiawan, M.Y. Idris, R.F. Malik, S. Nurmaini, N. Alsharif, R. Budiarto, Investigating brute force attack patterns in IoT network, J. Electr. Comput. Eng. 2019 (1) (2019) 4568368, [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2019/4568368. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1155/2019/4568368.

[60] S. Kothari, S. Joshi, Analysis of android applications to detect botnet attacks, in: 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing, ICSIDEMPC, IEEE, 2020, pp. 144–150.

[61] M. Nalini, R.K. Dhanraj, B. Balusamy, V. Abirami, K. Kavya, G. Aishwaryalakshmi, Bot-based process triggering by incoming E-mails and documents, Hyperautomation Next-Gener. Ind. (2024) 177–205.

[62] F.E. Ayo, J.B. Awotunde, S.O. Folorunso, R. Panigrahi, A. Garg, A.K. Bhoi, Bot-FFX: A robust and efficient framework for fast flux botnet (FFB) detection, Wirel. Pers. Commun. (2024) 1–24.

[63] C.D. McDermott, F. Majdani, A.V. Petrovski, Botnet detection in the internet of things using deep learning approaches, in: 2018 International Joint Conference on Neural Networks, IJCNN, IEEE, 2018, pp. 1–8.

[64] S. Islam, H. Elmekki, A. Elsebai, J. Bentahar, N. Drawel, G. Rjoub, W. Pedrycz, A comprehensive survey on applications of transformers for deep learning tasks, Expert Syst. Appl. (2023) 122666.

[65] A. Wang, W. Chang, S. Chen, A. Mohaisen, Delving into internet DDoS attacks by botnets: Characterization and analysis, IEEE/ACM Trans. Netw. 26 (6) (2018) 2843–2855.

[66] S. Baruah, Botnet detection: Analysis of various techniques, Int. J. Comput. Intell. IoT 2 (2) (2019).

[67] A.B. de Neira, B. Kantarci, M. Nogueira, Distributed denial of service attack prediction: Challenges, open issues and opportunities, Comput. Netw. 222 (2023) 109553.

[68] T.E. Ali, Y.-W. Chong, S. Manickam, Machine learning techniques to detect a DDoS attack in SDN: A systematic review, Appl. Sci. 13 (5) (2023) 3183.

[69] M. Tan, A. Iacovazzi, N.-M.M. Cheung, Y. Elovici, A neural attention model for real-time network intrusion detection, in: 2019 IEEE 44th Conference on Local Computer Networks, LCN, IEEE, 2019, pp. 291–299.

[70] A.S. Kumar, S. Raja, N. Pritha, H. Raviraj, R.B. Lincy, J.J. Rubia, An adaptive transformer model for anomaly detection in wireless sensor networks in real-time, Meas.: Sens. 25 (2023) 100625.

[71] M.R. Kadri, A. Abdelli, L. Mokdad, et al., Survey and classification of DoS and DDoS attack detection and validation approaches for IoT environments, Internet Things (2023) 101021.

[72] S. Karnani, H.K. Shakya, Mitigation strategies for distributed denial of service (DDoS) in SDN: a survey and taxonomy, Inf. Secur. J.: A Glob. Perspect. 32 (6) (2023) 444–468.

[73] E.P. Farias, A.C.J. Tavares, M. Nogueira, A runtime DDoS attack detection technique based on stochastic mathematical model, in: 2023 IEEE Latin-American Conference on Communications, LATINCOM, IEEE, 2023, pp. 1–6.

[74] Z.A. Alrahmani, K. Elleithy, DDoS attack forecasting using transformers, in: 2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress, DASC/PiCom/CBDCom/CyberSciTech, IEEE, 2023, pp. 0911–0915.

[75] A. Salam, F. Ullah, F. Amin, M. Abrar, Deep learning techniques for web-based attack detection in industry 5.0: A novel approach, Technologies 11 (4) (2023) 107.

[76] N. Ahuja, G. Singal, D. Mukhopadhyay, DLSDN: Deep learning for DDOS attack detection in software defined networking, in: 2021 11th International Conference on Cloud Computing, Data Science & Engineering, Confluence, IEEE, 2021, pp. 683–688.

[77] M. Mittal, K. Kumar, S. Behal, Deep learning approaches for detecting DDoS attacks: A systematic review, Soft Comput. 27 (18) (2023) 13039–13075.

[78] L. Yu, S. Yanlong, Z. Ying, Stateful protocol fuzzing with statemap-based reverse state selection, 2024, arXiv preprint arXiv:2408.06844.

[79] D.S. Eswari, P. Lakshmi, A survey on detection of DDoS attacks using machine learning approaches, Turk. J. Comput. Math. Educ. 12 (11) (2021) 4923–4931.

[80] K. Shaukat, S. Luo, V. Varadharajan, I.A. Hameed, M. Xu, A survey on machine learning techniques for cyber security in the last decade, IEEE Access 8 (2020) 222310–222354.

[81] P.S. Saini, S. Behal, S. Bhatia, Detection of DDoS attacks using machine learning algorithms, in: 2020 7th International Conference on Computing for Sustainable Global Development, INDIACom, IEEE, 2020, pp. 16–21.

[82] Y. Lu, Artificial intelligence: A survey on evolution, models, applications and future trends, J. Manag. Anal. 6 (1) (2019) 1–29.

[83] M. AbdulRaheem, I.D. Oladipo, A.L. Imoize, J.B. Awotunde, C.-C. Lee, G.B. Balogun, J.O. Adeoti, Machine learning assisted snort and zeek in detecting DDoS attacks in software-defined networking, Int. J. Inf. Technol. 16 (3) (2024) 1627–1643.

[84] T. Kaluarachchi, A. Reis, S. Nanayakkara, A review of recent deep learning approaches in human-centered machine learning, Sensors 21 (7) (2021) 29.

[85] S. Dasari, R. Kaluri, An effective classification of DDoS attacks in a distributed network by adopting hierarchical machine learning and hyperparameters optimization techniques, IEEE Access (2024).

[86] A. Behera, K.S. Sahoo, T.K. Mishra, M. Bhuyan, A combination learning framework to uncover cyber attacks in IoT networks, Internet Things 28 (2024) 101395, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660524003366.

[87] M. Hassanin, M. Keshk, S. Salim, M. Alsubaie, D. Sharma, Pllm-cs: Pre-trained large language model (LLM) for cyber threat detection in satellite networks, Ad Hoc Netw. 166 (2025) 103645.

[88] J.G. Almaraz-Rivera, J.A. Perez-Diaz, J.A. Cantoral-Ceballos, Transport and application layer DDoS attacks detection to IoT devices by using machine learning and deep learning models, Sensors 22 (9) (2022) 3367.

[89] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J.M. Moura, P. Eckersley, Explainable machine learning in deployment, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 2020, pp. 648–657.

[90] M.M. Taye, Understanding of machine learning with deep learning: architectures, workflow, applications and future directions, Computers 12 (5) (2023) 91.

[91] P. Jha, G. Singh, A. Kumar, D. Agrawal, Y.S. Patel, J. Forough, NetProbe: Deep learning-driven DDoS detection with a two-tiered mitigation strategy, in: Proceedings of the 26th International Conference on Distributed Computing and Networking, 2025, pp. 402–407.

[92] H. Polat, O. Polat, A. Cetin, Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models, Sustainability 12 (3) (2020) 1035.

[93] T. Saranya, S. Sridevi, C. Deisy, T.D. Chung, M.A. Khan, Performance analysis of machine learning algorithms in intrusion detection system: A review, Procedia Comput. Sci. 171 (2020) 1251–1260.

[94] T.T. Nguyen, V.J. Reddi, Deep reinforcement learning for cyber security, IEEE Trans. Neural Netw. Learn. Syst. (2021).

[95] N.M. Yungaicela-Naula, C. Vargas-Rosales, J.A. Perez-Diaz, SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning, IEEE Access 9 (2021) 108495–108512.

[96] G. Agrafiotis, E. Makri, I. Flionis, A. Lalas, K. Votis, D. Tzovaras, Image-based neural network models for malware traffic classification using PCAP to picture conversion, in: Proceedings of the 17th International Conference on Availability, Reliability and Security, 2022, pp. 1–7.

[97] S. Salmi, L. Oughdir, Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network, J. Big Data 10 (1) (2023) 1–25.

[98] A.A. Alahmadi, M. Aljabri, F. Alhaidari, D.J. Alharthi, G.E. Rayani, L.A. Marghalani, O.B. Alotaibi, S.A. Bajandouh, DDoS attack detection in IoT-Based networks using machine learning models: A survey and research directions, Electronics 12 (14) (2023) 3103.

[99] K.S. Kumavat, J. Gomes, Survey of detection techniques for DDoS attacks, in: 2022 3rd International Conference on Intelligent Engineering and Management, ICIEM, IEEE, 2022, pp. 657–663.

[100] Y.B. Sanap, P. Aher, A comprehensive survey on detection and mitigation of DDoS attacks enabled with deep learning techniques in cloud computing, in: 2023 6th International Conference on Advances in Science and Technology, ICAST, IEEE, 2023, pp. 149–154.

[101] S. salman Qasim, S.M. NSAIF, Advancements in time series-based detection systems for distributed denial-of-service (ddos) attacks: A comprehensive review, Babylon. J. Netw. 2024 (2024) 9–17.

[102] P. Zhang, F. He, H. Zhang, J. Hu, X. Huang, J. Wang, X. Yin, H. Zhu, Y. Li, Real-time malicious traffic detection with online isolation forest over SD-Wan, IEEE Trans. Inf. Forensics Secur. 18 (2023) 2076–2090.

[103] A. Banitalebi Dehkordi, M. Soltanaghaei, F.Z. Boroujeni, The DDoS attacks detection through machine learning and statistical methods in SDN, J. Supercomput. 77 (3) (2021) 2383–2415.

[104] S. Gyawali, Y. Qian, R.Q. Hu, Machine learning and reputation based misbehavior detection in vehicular communication networks, IEEE Trans. Veh. Technol. 69 (8) (2020) 8871–8885.

[105] M. Guastalla, Y. Li, A. Hekmati, B. Krishnamachari, Application of large language models to DDoS attack detection, in: International Conference on Security and Privacy in Cyber-Physical Systems and Smart Vehicles, Springer, 2023, pp. 83–99.

[106] S. Sattarpour, A. Barati, H. Barati, EBIDS: efficient BERT-based intrusion detection system in the network and application layers of IoT, Clust. Comput. 28 (2) (2024) 138.

[107] J. Zhang, L. Pan, Q.-L. Han, C. Chen, S. Wen, Y. Xiang, Deep learning based attack detection for cyber-physical system cybersecurity: A survey, IEEE/CAA J. Autom. Sin. 9 (3) (2021) 377–391.

[108] F.M. Shiri, T. Perumal, N. Mustapha, R. Mohamed, A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU, 2023, arXiv preprint arXiv:2305.17473.

[109] Y. Luo, X. Chen, N. Ge, W. Feng, J. Lu, Transformer-based device-type identification in heterogeneous IoT traffic, IEEE Internet Things J. 10 (6) (2022) 5050–5062.

[110] S. Haider, A. Akhunzada, I. Mustafa, T.B. Patel, A. Fernandez, K.-K.R. Choo, J. Iqbal, A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks, IEEE Access 8 (2020) 53972–53983.

[111] J. Kim, J. Kim, H. Kim, M. Shim, E. Choi, CNN-based network intrusion detection against denial-of-service attacks, Electronics 9 (6) (2020) 916.

[112] B. Lindemann, B. Maschler, N. Sahlab, M. Weyrich, A survey on anomaly detection for technical systems using LSTM networks, Comput. Ind. 131 (2021) 103498.

[113] K. Smagulova, A.P. James, A survey on LSTM memristive neural network architectures and applications, Eur. Phys. J. Spec. Top. 228 (10) (2019) 2313–2324.

[114] S. Yeom, C. Choi, K. Kim, LSTM-based collaborative source-side DDoS attack detection, IEEE Access 10 (2022) 44033–44045.

[115] A. Thangasamy, B. Sundan, L. Govindaraj, A novel framework for DDoS attacks detection using hybrid LSTM techniques, Comput. Syst. Sci. Eng. 45 (3) (2023).

[116] H. Aydın, Z. Orman, M.A. Aydın, A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment, Comput. Secur. 118 (2022) 102725.

[117] S. Luo, Z. Zhao, Q. Hu, Y. Liu, A hierarchical CNN-transformer model for network intrusion detection, in: 2nd International Conference on Applied Mathematics, Modelling, and Intelligent Computing, CAMMIC 2022, vol. 12259, SPIE, 2022, pp. 853–860.

[118] L.D. Manocchio, S. Layeghy, W.W. Lo, G.K. Kulatilleke, M. Sarhan, M. Portmann, Flowtransformer: A transformer framework for flow-based network intrusion detection systems, Expert Syst. Appl. 241 (2024) 122564.

[119] X. Wang, D. Pi, X. Zhang, H. Liu, C. Guo, Variational transformer-based anomaly detection approach for multivariate time series, Measurement 191 (2022) 110791.

[120] T. Lin, Y. Wang, X. Liu, X. Qiu, A survey of transformers, AI Open (2022).

[121] H. Çavşi Zaim, E.N. Yolaçan, FPE-Transformer: A feature positional encoding-based transformer model for attack detection, Appl. Sci. 15 (3) (2025) 1252.

[122] S. Khan, M. Naseer, M. Hayat, S.W. Zamir, F.S. Khan, M. Shah, Transformers in vision: A survey, ACM Comput. Surv. 54 (10s) (2022) 1–41.

[123] J.D.M.-W.C. Kenton, L.K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NaacL-HLT, vol. 1, Minneapolis, Minnesota, 2019, 2.

[124] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI Blog 1 (8) (2019) 9.

[125] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.

[126] J. Alammar, The illustrated transformer, 2018, https://jalammar.github.io/illustrated-transformer/.

[127] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, et al., A survey on visual transformer, 2020, arXiv preprint arXiv:2012.12556.

[128] S. Raschka, The illustrated transformer, 2023, https://sebastianraschka.com/blog/2023/self-attention-from-scratch.html.

[129] N. Adaloglou, Why multi-head self attention works: math, intuitions and 10+1 hidden insights, 2021, https://theaisummer.com/self-attention/.

[130] A. Zhang, Z.C. Lipton, M. Li, A.J. Smola, Dive into deep learning, 2021, arXiv preprint arXiv:2106.11342.

[131] L.F.R. Ribeiro, Explicando o multi-head attention no model transformer – deep learning e NLP, 2021, [Online]. Available: https://www.youtube.com/watch?v=6ZMtcSE34ps. Acessado em: 11 jul. 2024.

[132] Z. Wu, H. Zhang, P. Wang, Z. Sun, RTIDS: A robust transformer-based approach for intrusion detection system, IEEE Access 10 (2022) 64375–64387.

[133] N. Xing, S. Zhao, Y. Wang, K. Ning, X. Liu, A dynamic intrusion detection system capable of detecting unknown attacks, Int. J. Adv. Comput. Sci. Appl. 14 (7) (2023).

[134] M. Wang, N. Yang, N. Weng, Securing a smart home with a transformer-based IoT intrusion detection system, Electronics 12 (9) (2023) 2100.

[135] R. Verma, S. Chandra, RepuTE: A soft voting ensemble learning framework for reputation-based attack detection in fog-IoT milieu, Eng. Appl. Artif. Intell. 118 (2023) 105670.

[136] A. Ba, F. Lorenzi, J. Ploennigs, Monitoring of IoT systems at the edges with transformer-based graph convolutional neural networks, in: 2022 IEEE International Conference on Edge Computing and Communications, EDGE, IEEE, 2022, pp. 41–49.

[137] Y. Sun, L. Hou, Z. Lv, D. Peng, Informer-based intrusion detection method for network attack of integrated energy system, IEEE J. Radio Freq. Identif. 6 (2022) 748–752.

[138] R. Kozik, M. Pawlicki, M. Choraś, A new method of hybrid time window embedding with transformer-based traffic data classification in IoT-networked environment, Pattern Anal. Appl. 24 (4) (2021) 1441–1449.

[139] B. Wang, Y. Jiang, Y. Liao, Z. Li, DDoS-MSCT: A DDoS attack detection method based on multiscale convolution and transformer, IET Inf. Secur. 2024 (1) (2024) 1056705, [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/2024/1056705. arXiv: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/2024/1056705.

[140] M.F. Trujillo-Guerrero, S. Román-Niemes, M. Jaén-Vargas, A. Cadiz, R. Fonseca, J.J. Serrano-Olmedo, Accuracy comparison of CNN, LSTM, and transformer for activity recognition using IMU and visual markers, IEEE Access 11 (2023) 106650–106669.

[141] A. Pölz, A.P. Blaschke, J. Komma, A.H. Farnleitner, J. Derx, Transformer versus LSTM: A comparison of deep learning models for karst spring discharge forecasting, Water Resour. Res. 60 (4) (2024) e2022WR032602.

[142] S. Keele, et al., Guidelines for Performing Systematic Literature Reviews in Software Engineering, Technical Report, 2007, Technical report, ver. 2.3 ebse technical report. ebse.

[143] V. Wannmacher Pereira, D. Baretta, Leitura de e-book em formato linear e em formato de mapa conceitual: compreensão, processamento e estratégias, Forma y Función 33 (1) (2020) 147–171.

[144] V.W. Pereira, A predição na teia de estratégias de compreensão leitora, Confluência (2010) 81–91.

[145] M. Li, D. Han, D. Li, H. Liu, C.-C. Chang, MFVT: An anomaly traffic detection method merging feature fusion network and vision transformer architecture, EURASIP J. Wirel. Commun. Netw. 2022 (1) (2022) 39.

[146] C. Fang, W. Mi, P. Han, L. Zhai, A method of network traffic anomaly detection based on packet window transformer, in: 2022 7th IEEE International Conference on Data Science in Cyberspace, DSC, IEEE, 2022, pp. 199–205.

[147] J. Chen, H. Zhou, Y. Mei, G. Adam, N.D. Bastian, T. Lan, Real-time network intrusion detection via decision transformers, 2023, arXiv preprint arXiv:2312.07696.

[148] Y. Li, X. Yuan, W. Li, An extreme semi-supervised framework based on transformer for network intrusion detection, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 4204–4208.

[149] T.P. Nguyen, H. Nam, D. Kim, Transformer-based attention network for in-vehicle intrusion detection, IEEE Access (2023).

[150] P. Wang, X. Wang, Y. Song, J. Huang, P. Ding, Z. Yang, TransIDS: A transformer-based approach for intrusion detection in internet of things using label smoothing, in: 2023 4th International Conference on Computer Engineering and Application, ICCEA, IEEE, 2023, pp. 216–222.

[151] T. Zhang, Y. Wang, RLFAT: a transformer-based relay link forged attack detection mechanism in SDN, Electronics 12 (10) (2023) 2247.

[152] R. Yao, N. Wang, P. Chen, D. Ma, X. Sheng, A CNN-transformer hybrid approach for an intrusion detection system in advanced metering infrastructure, Multimedia Tools Appl. 82 (13) (2023) 19463–19486.

[153] Y. Lan, T. Truong-Huu, J. Wu, S.G. Teo, Cascaded multi-class network intrusion detection with decision tree and self-attentive model, in: 2022 IEEE International Conference on Data Mining Workshops, ICDMW, IEEE, 2022, pp. 1–7.

[154] F. Ullah, S. Ullah, G. Srivastava, J.C.-W. Lin, IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic, Digit. Commun. Netw. 10 (1) (2024) 190–204.

[155] X. Han, S. Cui, S. Liu, C. Zhang, B. Jiang, Z. Lu, Network intrusion detection based on N-gram frequency and time-aware transformer, Comput. Secur. 128 (2023) 103171.

[156] T. Wang, K. Fang, W. Wei, J. Tian, Y. Pan, J. Li, Microcontroller unit chip temperature fingerprint informed machine learning for IIoT intrusion detection, IEEE Trans. Ind. Inform. 19 (2) (2022) 2219–2227.

[157] O. Barut, Y. Luo, P. Li, T. Zhang, R1DIT: Privacy-preserving malware traffic classification with attention-based neural networks, IEEE Trans. Netw. Serv. Manag. (2022).

[158] Y. Liu, L. Wu, Intrusion detection model based on improved transformer, Appl. Sci. 13 (10) (2023) 6251.

[159] Z. Chen, D. Chen, X. Zhang, Z. Yuan, X. Cheng, Learning graph structures with transformer for multivariate time-series snomaly detection in IoT, IEEE Internet Things J. 9 (12) (2021) 9179–9189.

[160] J. Casajús-Setién, C. Bielza, P. Larrañaga, Anomaly-based intrusion detection in IIoT networks using transformer models, in: 2023 IEEE International Conference on Cyber Security and Resilience, CSR, IEEE, 2023, pp. 72–77.

[161] T. Kim, W. Pak, Deep learning-based network intrusion detection using multiple image transformers, Appl. Sci. 13 (5) (2023) 2754.

[162] A. Ghourabi, A security model based on lightGBM and transformer to protect healthcare systems from cyberattacks, IEEE Access 10 (2022) 48890–48903.

[163] Z. Ali, W. Tiberti, A. Marotta, D. Cassioli, Empowering network security: BERT transformer learning approach and MLP for intrusion detection in imbalanced network traffic, IEEE Access 12 (2024) 137618–137633.

[164] M.A. Ferrag, M. Ndhlovu, N. Tihanyi, L.C. Cordeiro, M. Debbah, T. Lestable, N.S. Thandi, Revolutionizing cyber threat detection with large language models: A privacy-preserving BERT-based lightweight model for IoT/IIoT devices, 2024, [Online]. Available: https://arxiv.org/abs/2306.14263. arXiv:2306.14263.
[165] N. Alkhatib, M. Mushtaq, H. Ghauch, J.-L. Danger, CAN-BERT do it? Controller area network intrusion detection system based on bert language model, in: 2022 IEEE/ACS 19th International Conference on Computer Systems and Applications, AICCSA, IEEE, 2022, pp. 1–8.
[166] J. Liu, M. Simsek, B. Kantarci, M. Bagheri, P. Djukic, Collaborative feature maps of networks and hosts for AI-driven intrusion detection, in: GLOBECOM 2022-2022 IEEE Global Communications Conference, IEEE, 2022, pp. 2662–2667.
[167] H. Hou, D. Liang, M. Zhang, D. Yuan, A densely stacked attention method for cyberattack detection, J. Inf. Sci. Eng. 39 (4) (2023).
[168] A.I. Alzahrani, A. Al-Rasheed, A. Ksibi, M. Ayadi, M.M. Asiri, M. Zakariah, Anomaly detection in fog computing architectures using custom tab transformer for internet of things, Electronics 11 (23) (2022) 4017.
[169] S. Wang, Z. Li, C. Ding, B. Yuan, Q. Qiu, Y. Wang, Y. Liang, C-LSTM: Enabling efficient LSTM using structured compression techniques on FPGAs, in: Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2018, pp. 11–20.
[170] S. Wang, B.Z. Li, M. Khabsa, H. Fang, H. Ma, Linformer: Self-attention with linear complexity, 2020, arXiv preprint arXiv:2006.04768.
[171] A. Vyas, A. Katharopoulos, F. Fleuret, Fast transformers with clustered attention, Adv. Neural Inf. Process. Syst. 33 (2020) 21665–21674.
[172] U. Javed, K. Ijaz, M. Jawad, I. Khosa, E.A. Ansari, K.S. Zaidi, M.N. Rafiq, N. Shabbir, A novel short receptive field based dilated causal convolutional network integrated with bidirectional LSTM for short-term load forecasting, Expert Syst. Appl. 205 (2022) 117689.
[173] N. Agrawal, S. Tapaswi, An SDN-assisted defense mechanism for the shrew DDoS attack in a cloud computing environment, J. Netw. Syst. Manage. 29 (2) (2021) 12.
[174] M. Cirillo, M. Di Mauro, V. Matta, M. Tambasco, Botnet identification in DDoS attacks with multiple emulation dictionaries, IEEE Trans. Inf. Forensics Secur. 16 (2021) 3554–3569.
[175] M. Ma, L. Han, C. Zhou, Research and application of transformer based anomaly detection model: A literature review, 2024, arXiv preprint arXiv:2402.08975.
[176] R. Doshi, N. Apthorpe, N. Feamster, Machine learning DDoS detection for consumer internet of things devices, in: 2018 IEEE Security and Privacy Workshops, SPW, IEEE, 2018, pp. 29–35.
[177] S. Wang, B. Zheng, Z. Liu, Z. Fan, Y. Liu, Y. Dai, A lightweight intrusion detection system for vehicular networks based on an improved ViT model, IEEE Access 12 (2024) 118842–118856.
[178] S. Ahmad, I. Shakeel, S. Mehfuz, J. Ahmad, Deep learning models for cloud, edge, fog, and IoT computing paradigms: Survey, recent advances, and future directions, Comput. Sci. Rev. 49 (2023) 100568.
[179] H. Bangui, B. Buhnova, Lightweight intrusion detection for edge computing networks using deep forest and bio-inspired algorithms, Comput. Electr. Eng. 100 (2022) 107901.

Euclides Peres Farias Junior He is a Ph.D. student in Computer Science at the Federal University of Paraná (UFPR) and a member of the Computer Security Science Center (CCSC). He has a master's degree in Computer Science from the Pontifical Catholic University of Paraná (PUCPR). He is currently an assistant professor in the Department of Computer Science at the Federal Technological University of Paraná, Santa Helena campus (UTFPR-SH). His research interests include computer networks, computational security, computer forensics, and operating systems.

Anderson Bergamini de Neira received the Ph.D. degree in Computer Science from Federal University of Paraná (UFPR), Brazil (2024). His main research interest includes security in computer networks, especially in solutions that use machine learning to reduce the impacts of DDoS attacks. He is a member of the Center for Computational Security Science research team.

Ligia Francielle Borges is a postdoctoral researcher in the Department of Computer Science at the Federal University of Minas Gerais (UFMG). She is a cybersecurity researcher for the MENTORED project and a recipient of a fellowship from the São Paulo State Science Foundation (FAPESP). She holds a Ph.D. in Computer Science from the Federal University of Paraná (UFPR) and is a member of the Computer Security Science Center (CCSC). She received her M.Sc. degree in Computational Technologies for Agribusiness from the Federal University of Technology, Paraná, and her B.Tech. degree in Computer Networks from the Higher Education Centre of Foz do Iguaçu, Brazil. Her research focuses on computer networks, nanonetworks, and cybersecurity.

Michele Nogueira received the Ph.D. degree in Computer Science from Sorbonne University–UPMC, LIP6, France (2009). She is currently an Associate Professor of the Computer Science Department at Federal University of Minas Gerais. Her research interests include wireless networks, network security and dependability. Her works search to provide resilience to self-organized, cognitive and wireless networks by adaptive and opportunistic approaches. She is the director of the Center for Computational Security sCience (CCSC) research lab. She served as an Associate Technical Editor for the IEEE Communications Magazine and the Journal of Network and Systems Management. She served as chair for the IEEE ComSoc Internet Technical Committee.