INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research

International Journal of Web Engineering and Technology

ISSN online: 1741-9212 - ISSN print: 1476-1289
https://www.inderscience.com/ijwet

A distributed framework for distributed denial-of-service attack detection in internet of things environments using deep learning

Wawire Amisi Silas, Lawrence Nderu, Dalton Ndirangu

DOI: 10.1504/IJWET.2024.10062503

Article History:

Received:	08 September 2023
Last revised:	21 November 2023
Accepted:	30 November 2023
Published online:	29 April 2024

A distributed framework for distributed denial-of-service attack detection in internet of things environments using deep learning

Wawire Amisi Silas*, Lawrence Nderu and Dalton Ndirangu

Jomo Kenyatta University of Agriculture and Technology,
United States International University Africa,
USIU Road, Off Thika Road, Exit 7, Kenya
Email: swawire@gmail.com
Email: lawrence_nderu@live.com
Email: dndirangu@usiu.ac.ke
*Corresponding author

Abstract: Internet of things (IoT) networks dominate industries, homes, organisations, and other aspects of life owing to their automation capabilities. However, IoT networks are vulnerable to attacks, especially distributed denial-of-service (DDoS) attacks, as they tend to have low computational capabilities and are highly diverse. While current research shows the potential of utilising deep learning methods to detect DDoS attacks, there is a lack of a framework that can be used to deploy an effective deep learning algorithm to detect DDoS attacks in heterogeneous IoT environments. Accordingly, this paper developed a DDoS detection framework based on the CNN-BiLSTM model, which can be deployed in a distributed network and includes adequate pre-processing. Simulations were also done to demonstrate the application of the framework and its effectiveness.

Keywords: machine learning; artificial intelligence; internet of things; IoT; deep learning; convolutional neural networks; CNNs; BiLSTM; distributed denial-of-service; DDoS.

Reference to this paper should be made as follows: Silas, W.A., Nderu, L. and Ndirangu, D. (2024) 'A distributed framework for distributed denial-of-service attack detection in internet of things environments using deep learning', *Int. J. Web Engineering and Technology*, Vol. 19, No. 1, pp.67–87.

Biographical notes: Wawire Amisi Silas is a graduate student at the Department of Computing at Jomo Kenyatta University of Technology and Agriculture. His interests include internet of things, artificial intelligence, and cyber security.

Lawrence Nderu is a Lecturer and the Chairperson of the Department of Computing at Jomo Kenyatta University of Agriculture and Technology Kenya. He is an experienced university educator with a demonstrated history of working in the higher education industry. He is skilled in software development (Python, Java, C#, C++, R, JavaScript, Android, PHP, C++, and databases. Additionally, possesses a Doctor of Philosophy – PhD focused in Computer Science from University of Vincennes in Saint-Denis, Paris, France. He is also the Founder of JKIAN, which is a digital solutions hub providing digital climate smart agriculture and trade solutions to small and medium scale farmers.

Dalton Ndirangu is an Associate Professor in the School of Science and Technology at the United States International University-Africa. He received his BSc in Statistics and Computer from the University of Nairobi and an MSc in Computer Science from National University of Science and Technology, Bulawayo, Zimbabwe. He earned his Doctorate in Information Technology from Jomo Kenyatta University of Agriculture and Technology (JKUAT). He has a long career spanning several organisations and has consulted with a number of organisation and institutions in Kenya and beyond. He is an accomplished scholar and has supervised many postgraduate students. His research interests include data science, artificial intelligence, and software engineering.

1 Introduction

The rapid growth of internet of things (IoT) networks has ushered in a new era of connectivity hence transforming lives and industries. It has brought about unprecedented levels of efficiency, automation, and convenience by enabling data exchange and seamless communication among everyday devices (Aboubakar et al., 2022). From healthcare systems to smart homes and cities, IoT has revolutionised the way human beings interact with technology. At its core, the IoT trend encompasses incorporating computing and communication capabilities into everyday devices. Accordingly, the typical IoT network comprises sensors and actuators, computing, and connectivity (Thoutam, 2021). Other components might include security and cloud-based analytics, platforms, and visualisation. Because of the immense benefits of IoT networks, the market for such systems has grown exponentially in recent years. According to Fortune Business Insights (2023), the global IoT market size was valued at $544 billion in 2022 and it is projected to grow to $3.35 trillion by 2030, which is a compound annual growth rate (CAGR) of 26.1%. However, alongside the widespread adoption of IoT, there are inherent vulnerabilities that pose significant challenges. These vulnerabilities have made IoT environments attractive targets for malicious actors, leading to a surge in security threats, especially distributed denial of service (DDoS) attacks.

A DDoS attack happens when a network or system is flooded with a devastating quantity of traffic or requests, rendering it unable to respond to authentic users. In the context of IoT, DDoS attacks can be especially overwhelming. Since IoT networks comprise many interconnected devices, each with its computing abilities, these attacks can exploit the collective power of compromised IoT devices to launch large-scale assaults. DDoS attacks targeting IoT networks can compromise the availability, integrity, and confidentiality of the data being transmitted (Bhattacharjya, 2022). Whether it is sensitive personal information, industrial secrets, or real-time operational data, unauthorised access to IoT data can have major repercussions, including espionage, identity theft, and financial fraud. Moreover, the reliability, trustworthiness, and utilisation of IoT systems themselves can be weakened by DDoS attacks. Indeed, the potential benefits of IoT adoption may be overshadowed by concerns about privacy, safety, and the overall resilience of the infrastructure.

The detection of DDoS attacks in IoT environments presents a significant challenge, primarily due to the unique characteristics of these networks. Traditional machine learning methods, which were primarily designed for conventional networks, exhibit

deficiencies when applied to IoT environments for DDoS attack detection (Aktar and Nur, 2023). The limitations of traditional machine learning methods in IoT environments arise from the distinct features of IoT networks: heterogeneity and resource limitations. These networks encompass a wide range of heterogeneous devices with different computational capabilities, memory constraints, and communication protocols. Consequently, it becomes difficult to devise uniform detection mechanisms that cater to the diverse nature of IoT devices and ensure accurate and efficient DDoS attack identification across the entire network. However, the detection of DDoS attacks can benefit from machine learning, especially deep learning, when implemented using a suitable framework (Ali et al., 2023). Deep learning is a subset of machine learning that focuses on artificial neural networks (ANNs) with multiple layers.

This paper aims to make a significant contribution to the field by providing a reliable and efficient framework that can be used to create solutions for detecting DDoS attacks in IoT environments. By leveraging the power of deep learning techniques and considering the unique characteristics of IoT networks, the research enabled the development of a robust framework capable of accurately identifying DDoS attacks in real-time. The outcomes of this research have the potential to greatly enhance the security posture of IoT networks, empowering organisations to detect and respond to DDoS attacks swiftly. By implementing the distributed detection framework, organisations can gain better visibility into the security of their IoT deployments, mitigate the risks posed by DDoS attacks, and ensure the continuous and reliable operation of their IoT-enabled systems and services. In addition to protecting the security and privacy of their systems, the framework can enhance the adoption of IoT-enabled services. Ultimately, the research aims to contribute to the advancement of IoT security practices and support the sustainable growth and adoption of IoT technologies across different industries and sectors. The objectives of the study are listed below.

1 To design and implement a distributed DDoS detection framework based on deep learning.

2 To evaluate the performance of the framework using appropriate metrics and a real-world dataset.

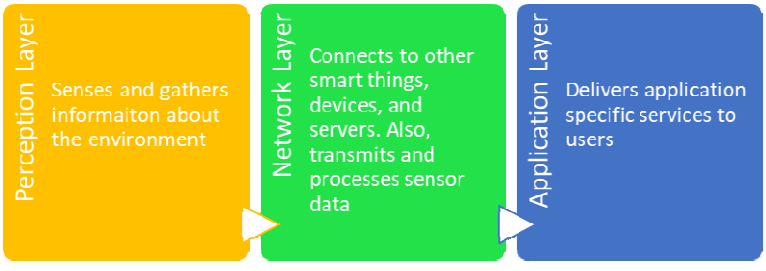
2 Literature review

IoT networks, which are an integral part of the IoT ecosystem, consist of interconnected devices that support communication and data exchange. These networks connect physical devices, sensors, actuators, and other objects to the internet, allowing them to sense and exchange data, perform automated actions, and enable remote control and monitoring (HaddadPajouh et al., 2021). Understanding the architecture, components, and communication protocols of IoT networks is crucial for comprehending the unique challenges they present. IoT networks typically follow a decentralised architecture, often referred to as the edge computing paradigm. The edge element denotes that data processing, storage, and communication take place closer to the IoT devices themselves. This decentralised approach minimises latency, reduces bandwidth usage, and allows for localised decision-making and data analytics (Mtowe and Kim, 2023). According to Sethi and Sarangi (2017), the architecture often comprises three layers: the perception layer (devices and sensors), the network layer (connectivity and communication), and the

application layer (data processing and analysis). Figure 1 illustrates the common architecture of IoT networks.

IoT devices possess specific characteristics that differentiate them from traditional computing devices. According to Noaman et al. (2022), these features include limited computational capability and heterogeneity. As such, IoT networks are vulnerable to a myriad of attacks. To start with, IoT systems exhibit weak authentication mechanisms (Williams et al., 2022). Many IoT devices employ weak or default credentials, making them vulnerable to brute-force attacks and unauthorised access. Additionally, the lack of end-to-end encryption in IoT networks exposes data to interception and unauthorised access (Williams et al., 2022). Furthermore, IoT devices often lack robust update mechanisms, making them susceptible to attacks targeting vulnerabilities in their firmware. Finally, IoT devices are susceptible to physical tampering, which poses a significant risk to IoT devices deployed in uncontrolled or publicly accessible environments. Attackers can manipulate devices, extract sensitive data, inject malicious code, or gain unauthorised physical access, compromising the security and integrity of the entire network.

Figure 1 Architecture of IoT networks (see online version for colours)



The consequences associated with exploiting the vulnerabilities can be dire. For example, weak security measures can result in unauthorised access to personal information, compromising user privacy. This can lead to identity theft, unauthorised surveillance, and misuse of personal data. In addition, compromised IoT devices can serve as platforms for launching malware attacks, such as botnets, which can propagate across networks and disrupt critical infrastructure or conduct large-scale cyberattacks. Attackers can also exploit vulnerabilities in IoT devices to launch DoS attacks, overwhelming networks with an influx of traffic and rendering them unavailable or unreliable. This can cause severe disruptions in critical systems or services. The deployment of IoT networks often entails making a trade-off between security and usability. Based on Di Nocera and Tempestini (2022), implementing strong security measures often introduces complexity, inconvenience, and additional costs. Balancing security requirements with user experience and ease of use is crucial to ensure wider adoption and acceptance of IoT technologies.

There are several types of DDoS attacks affecting IoT networks. SYN floods take advantage of the three-way handshake mechanism in TCP/IP to overwhelm IoT devices with a flood of connection requests (Lygerou et al., 2022). By exhausting the device's resources, SYN floods render it unable to accept legitimate connections. Furthermore, TCP ACK floods flood IoT devices with a high volume of TCP ACK packets, consuming their processing power and network bandwidth. This attack exhausts device resources,

leading to service unavailability. Additionally, TCP exhaustion attacks exploit IoT device limitations by establishing and maintaining many legitimate TCP connections simultaneously, ultimately exhausting the device's available connection slots.

The application layer of the IoT framework is vulnerable to diverse attacks. Applications often have vulnerabilities that can be exploited in an attack. For instance, HTTP/HTTPS floods target IoT devices by overwhelming them with a massive number of HTTP/HTTPS requests. These floods aim to exhaust the device's processing power, memory, or network resources (Lygerou et al., 2022). Similarly, DNS amplification attacks entail amplifying the volume of traffic directed towards IoT devices. By exploiting the DNS protocol's recursive query functionality, attackers achieve greater attack bandwidth, overwhelming the targeted devices. Some DDoS attacks focus on exploiting vulnerabilities in IoT-specific protocols, such as MQTT, CoAP, or SNMP (Lygerou et al., 2022). By targeting weaknesses in these protocols, attackers can disrupt IoT device functionality and compromise the entire IoT ecosystem.

The primary objective of DDoS attacks in IoT environments is to disrupt the availability of IoT services, rendering them inaccessible to legitimate users. This can cause financial losses, reputational damage, and potential safety risks in critical IoT deployments (Mishra and Pandya, 2021). DDoS attacks can also aim to exhaust the network, computational, or memory resources of IoT devices, rendering them incapable of performing their intended functions. This resource depletion affects the overall performance and reliability of IoT systems (Mishra and Pandya, 2021). Moreover, DDoS attacks can act as a smokescreen, diverting attention from other malicious activities, such as data breaches, unauthorised access, or malware propagation. Attackers often leverage DDoS attacks to mask their true objectives and exploit security vulnerabilities within the IoT environment.

Detecting DDoS attacks in IoT environments is a crucial task to ensure the security and availability of IoT systems. Numerous methods and techniques have been proposed in the literature to detect DDoS attacks effectively. Firstly, flow-based detection methods focus on analysing network flows, which are a sequence of packets sharing common characteristics (e.g., source and destination IP addresses, ports, protocols) (Adedeji et al., 2023). These methods involve collecting flow data from IoT devices and applying various analysis techniques, such as entropy-based analysis, statistical analysis, or pattern recognition, to identify anomalies or patterns indicative of DDoS attacks (Singh and Bhandari, 2020). Flow-based approaches are advantageous in IoT environments as they provide a lightweight solution suitable for resource-constrained IoT devices.

Collaborative filtering techniques leverage the collective knowledge and behaviour of multiple IoT devices or network entities to detect DDoS attacks. These approaches involve sharing information and aggregating feedback from various IoT devices in real-time to identify abnormal patterns or deviations from the expected behaviour (Gaurav and Singh, 2017). Collaborative filtering can help detect sophisticated DDoS attacks that may exhibit low-level attack traffic, making them difficult to detect using traditional methods.

Statistical approaches, which entail analysing statistical properties of network traffic to identify DDoS attacks, can also be utilised. These methods often use metrics such as traffic volume, packet rate, inter-packet arrival time, and packet size distribution (Banitalebi et al., 2021). Deviations from the expected statistical patterns can indicate the presence of an ongoing DDoS attack. Statistical-based approaches can be efficient for detecting both volumetric and application-layer DDoS attacks in IoT environments. Some

authors have adopted hybrid approaches that combine multiple detection techniques to improve the accuracy and robustness of DDoS attack detection. Such approaches leverage the strengths of different methods to overcome their individual limitations and enhance the overall detection capability.

Machine learning-based approaches have gained significant attention in DDoS attack detection due to their ability to learn and recognise complex patterns in network traffic data. Machine learning enables the transition from signature-based DDoS detection methods, which are unable to detect novel attacks. According to Adedeji et al. (2023), signature-based detection entails comparing known attack signatures with current traffic patterns to detect attacks. This implies that only attacks that have been identified previously can be detected. While signature-based approaches are highly accurate when it comes to detecting known attacks, they cannot detect the same attacks when the signatures are altered. Machine learning alleviates this issue by providing a mechanism for detecting novel attacks. Although traditional learning methods have shown immense success in anomaly detection, they continue to undeform as compared to deep learning (Bahashwan et al., 2023). Examples of these traditional methods include SVMs, K-means clustering, and decision trees.

Deep learning leverages deep neural networks to automatically extract relevant features and classify network traffic. Deep learning is inspired by the structure and function of the human brain's neural networks (Ali et al., 2023). It is characterised by the utilisation of deep neural networks, which consist of multiple layers of interconnected artificial neurons. These networks are capable of learning hierarchical representations of data, enabling them to automatically extract meaningful features from raw input. The most used deep learning architectures include convolutional neural networks (CNNs) for image processing, recurrent neural networks (RNNs) for sequential data analysis, and generative adversarial networks (GANs) for generating synthetic data.

The development of efficient training algorithms has been a significant advancement in deep learning. Backpropagation, coupled with stochastic gradient descent, is the cornerstone algorithm for training deep neural networks (Tian et al., 2023). However, variations such as batch normalisation, dropout, and adaptive learning rate optimisation techniques have enhanced training efficiency, convergence speed, and generalisation performance. Researchers have introduced innovative deep-learning architectures to address specific challenges. For example, CNNs have demonstrated remarkable success in image classification, object detection, and semantic segmentation tasks (Sanzana et al., 2022). RNNs, with variations like long short-term memory (LSTM) and gated recurrent units (GRUs), excel in modelling sequential data, enabling tasks like language translation and sentiment analysis (Sanzana et al., 2022). Attention mechanisms have improved the performance of deep learning models by selectively focusing on relevant information. Transfer learning has gained prominence in deep learning, allowing models to leverage knowledge learned from one task or domain to improve performance on related tasks with limited data. Pre-training on large-scale datasets, such as ImageNet, followed by fine-tuning on target tasks, has become a common practice to achieve state-of-the-art results in various applications.

Convolutional neural network-bidirectional long short-term memory (CNN-BiLSTM) can be an effective algorithm for detecting DDoS attacks. Essentially, CNN-BiLSTM is a deep learning architecture that combines the power of CNNs and bidirectional long short-term memory (BiLSTM) networks (Lu et al., 2023). It is commonly used for tasks involving sequential data, which makes it ideal for DDoS traffic classification. CNNs

excel at capturing spatial hierarchies and local patterns, making them ideal for image-processing tasks (Lu et al., 2023). However, they have limited ability to model long-range dependencies in sequential data. The bidirectional nature of the BiLSTM allows it to effectively model dependencies in both directions, making it well-suited for tasks where context from both past and future is crucial, such as natural language processing.

The CNN-BiLSTM architecture, therefore, combines the strengths of CNNs in capturing local features and spatial hierarchies with the ability of BiLSTMs to model long-range dependencies in sequential data. This architecture is commonly used for text classification tasks, where the CNN component processes the input text at a local level, extracting relevant features, while the BiLSTM component captures the contextual information across the entire sequence (Halder and Chatterjee, 2020). The creation of the typical CNN-BiLSTM model starts with the input layer, which represents the input data, such as word embeddings or one-hot encoded vectors (Lu et al., 2023). Next convolutional filters are applied to capture local features and patterns in the input. Thereafter pooling layers are used to down-sample the feature maps to reduce dimensionality. There are also the BiLSTM layers that capture long-range dependencies by processing the feature maps in both forward and backward directions (Halder and Chatterjee, 2020). Subsequently, fully connected layers are implemented to perform classification or regression based on the learned representations. Finally, the output layer generates the final output, such as predicted labels or probabilities. By combining CNN and BiLSTM layers, the model can effectively learn hierarchical representations and contextual information from sequential data, making it a powerful architecture for various classification tasks.

Various studies have examined the performance of different deep learning models in the classification of DDoS traffic. For example, Aswad et al. (2023) examined the effectiveness of RNN, CNN, LSTM, and CNN-BiLSTM in detecting and distinguishing DDoS traffic from legitimate traffic. The study established that the ensemble model comprising CNN and BiLSTM was the most effective. Likewise, Roopak et al. (2019) compared the effectiveness of four deep learning models: MLP, CNN, LSTM, and CNN+LSTM. Based on the findings, the final model (CNN+LSTM) had the best accuracy. Diro and Chilamkurti (2018) proposed a distributed architecture for IoT networks in which fog nodes were utilised to train models and host attack detection systems. At the same time, master nodes were designed to conduct collaborative parameter sharing and optimisation. The architecture made use of a deep learning model, which was tested using the NSL-KDD dataset (Diro and Chilamkurti, 2018). The findings demonstrated that a distributed model was more effective as compared to a centralised one.

While the utilisation of machine learning methods to detect DDoS attacks in IoT environments has been widely examined, there lacks a framework that can be deployed to detect DDoS attacks in a distributed architecture while performing the required pre-processing tasks. Lawal et al. (2021), for example, studied the detection of DDoS attacks in IoT networks using the k-NN classification algorithm. Similarly, Kumar et al. (2021) made use of random forest (RF) and XGBoost to detect DDoS. Accordingly, there is a need for a framework that leverages an effective deep learning algorithm and that can be deployed in a distributed network architecture to detect DDoS attacks. The framework must be able to work well in different environments (wired, wireless, and a combination of both wired and wireless connectivity). Furthermore, it should help in detecting different types of DDoS attacks, including zero-day attacks, with optimal accuracy.

The detection of DDoS attacks is largely centralised despite the ongoing adoption of fog computing. According to Alghazzawi et al. (2021), deep learning detection methods can be deployed strategically across a distributed network of computing nodes, situated at fog or edge layers. The distributed approach could enhance the responsiveness and efficiency of DDoS detection by minimising the latency associated with transmitting large volumes of data to a centralised location for analysis. As such, the detection framework presented in this paper aligns with the conventional three-layered IoT architecture. To start with, the cloud layer is designed to possess the global model for detecting DDoS attacks. This model is constituted by the current gradient weights obtained from fog nodes. The fog nodes, which represent the second layer of the architecture, are designed to train the deep learning model in a distributed format using pre-processed data obtained from the edge layer. A coordinating master is implemented in the fog layer to manage parameter optimisation, validation, and exchange. To evaluate the framework, the confusion matrix was used to display the predicted and actual labels of a dataset

3 Methodology

3.1 Dataset

The dataset that was utilised to evaluate the proposed framework is called DoS/DDoS-MQTT-IoT (Alatram et al., 2023). A key strength of this dataset is that it includes data obtained using the message queuing telemetry protocol (MQTT), which has become a popular protocol for machine-to-machine IoT communications. Since this is the only current real-world dataset available that includes MQTT data, it can be utilised to evaluate the effectiveness of the countermeasures implemented to deal with modern attacks targeting IoT systems (Alatram et al., 2023). The creation of the dataset entailed constructing a physical IoT testbed and generating a large volume of IoT data, which encompassed the standard MQTT traffic and ten denial-of-service scenarios. MQTT is a messaging protocol designed for efficient device communication, particularly in resource-limited or unreliable networks that characterise contemporary IoT systems (Sanjuan et al., 2020). It follows the publish-subscribe model in which devices can publish messages on specific topics, and others can subscribe to those topics to receive messages.

The DoS/DDoS-MQTT-IoT dataset has various attributes that made it ideal for the study: the utilisation of a realistic testbed, collection of realistic traffic data, labelled dataset, and the inclusion of IoT data, MQTT attack data, and MQTT DoS/DDoS attack data (Alatram et al., 2023). Therefore, the setup and arrangement of the simulation environment closely resemble real-world conditions, facilitating accurate testing and experimentation. Additionally, the data was authentic and representative hence mimicking patterns and behaviours of actual network traffic. The DoS/DDoS-MQTT-IoT dataset is also accompanied by descriptive tags or identifiers, allowing for clear categorisation and analysis. The data was generated by IoT devices, with this data comprising instances of security attacks specifically targeting the message queuing telemetry transport (MQTT) protocol (Alatram et al., 2023). Finally, the dataset comprised information about DDoS attacks specifically directed at MQTT protocol implementations.

The development of the DoS/DDoS-MQTT-IoT dataset entailed collecting normal data as well as various variations of DoS attacks against the MQTT protocol. According to Alatram et al. (2023), the abnormal data relating to the various attacks were simulated: CONNECT flooding attack (BF_DoS and BF_DDoS), Delayed CONNECT flooding attack (Delay_DoS and Delay_DDoS), invalid subscription flooding attack (Sub_DoS and Sub_DDoS), CONNECT flooding with WILL payload attack (WILL_DoS and WILL_DDoS), and TCP SYN flooding attack (SYN_DoS and SYN_DDoS). The normal MQTT traffic was captured by utilising the normal states of the protocol. Table 1 summarises the MQTT-IoT datasets, which include both malicious and benign data.

Table 1 MQTT-IoT datasets

Dataset name	*File size*	*Quantity of files*	*#Records per file*
Normal MQTT	50 MB	20	≈ 490000
	200 MB	30	≈ 1900000
BF_DoS	50 MB	20	≈ 510000
	200 MB	10	≈ 2000000
BF_DDoS	50 MB	20	≈ 510000
	200 MB	10	≈ 2000000
Delay_DoS	50 MB	20	≈ 500000
	200 MB	10	≈ 660000
Delay_DDoS	50 MB	20	≈ 510000
	200 MB	10	≈ 2000000
Sub_DoS	50 MB	20	≈ 130000
	200 MB	10	≈ 800000
Sub_DDoS	50 MB	20	≈ 200000
	200 MB	10	≈ 750000
WILL_DoS	50 MB	20	≈ 190000
	200 MB	10	≈ 650000
WILL_DDoS	50 MB	20	≈ 250000
	200 MB	10	≈ 1000000
SYN_DoS	50 MB	33	≈ 500000
	200 MB	10	≈ 1500000
SYN_DDoS	50 MB	20	≈ 500000
	200 MB	10	≈ 1500000

The datasets comprised 30 features as shown in Table 2.

Table 2 Feature description and the associated data type

No	*Description*	*Feature*	*Data type (number or string)*
1	Epoch time	frame.time_epoch	N
2	Frame length	frame.len	N
3	Time delta from previous displayed frame	frame.time_delta_displayed	N

Table 2 Feature description and the associated data type (continued)

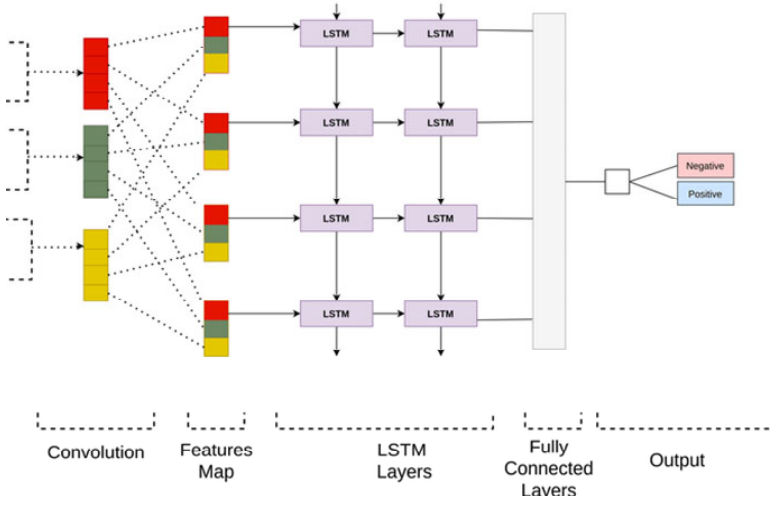
No	*Description*	*Feature*	*Data type (number or string)*
4	Time since reference or first frame	frame.time_relative	N
5	Source IP address	ip.src	S
6	Protocol	ip.proto	S
7	Stream index	tcp.stream	N
8	iRTT	tcp.analysis.initial.rtt	N
9	Time since first frame in this TCP stream	tcp.time_relative	N
10	TCP segment len	tcp.len	N
11	Calculated window size	tcp.window_size	N
12	Syn	tcp.flags.syn	S
13	Reset	tcp.flags.reset	S
14	Acknowledgment	tcp.flags.ack	S
15	Message type	mqtt.msgtype	S
16	QoS level	mqtt.qos	S
17	QoS level flag	mqtt.conflag.qos	S
18	MQTT subscriber QoS	mqtt.sub.qos	N
19	Clean session flag	mqtt.conflag.cleansess	S
20	Keep alive	mqtt.kalive	N
21	User name length	mqtt.username_len	N
22	Password length	mqtt.passwd_len	N
23	Retain	mqtt.retain	N
24	Will retain	mqtt.conflag.retain	S
25	Will flag	mqtt.conflag.willflag	S
26	Will message length	mqtt.willmsg_len	N
27	Will topic length	mqtt.willtopic_len	N
28	Topic length	mqtt.topic_len	N
29	Msg. len	mqtt.len	N
30	Return code	mqtt.conack.val	N

3.2 Deep learning algorithm

CNN-BiLSTM was selected for this study as it is ideal for supervised learning and exhibits excellent performance with highly correlated features. It is a type of neural network architecture used for various sequence-based tasks, including DDoS attack classification. For example, in a study by Aswad et al. (2023), CNN-BiLSTM realised an accuracy of 99.76% and a precision of 98.9% when utilised to detect DDoS attacks. Indeed, ensemble models combine the benefits of each model while suppressing the individual weaknesses hence producing optimal outcomes. Essentially, the CNN-BiLSTM algorithm combines the strengths of two different architectures: BiLSTM and CNN.

Although CNNs are primarily used for image processing, they can also be applied to sequential data like text. In the context of the CNN-BiLSTM model, CNNs are used to extract local features or patterns from the input sequence. A basic one-dimensional convolutional neural network (1D CNN) architecture for text data encompasses applying multiple convolutional filters with different kernel sizes over the input sequence (Alghazzawi et al., 2021). Each filter slides over the sequence and performs a convolution operation, capturing different n-gram features. This is often followed by a max-pooling layer to minimise the dimensionality of the extracted features.

Figure 2 CNN-BILSTM model (see online version for colours)



The LSTM is a type of RNN architecture designed for capturing long-range dependencies in sequences. This approach attempts to address one of the limitations of standard LSTMs in that conventional LSTMs process sequences in a unidirectional manner hence failing to capture the context from both past and future data points (Staffini, 2023). On the contrary, in the BiLSTM, one has two sets of LSTM cells: one set for processing the sequence forward and another set for processing the sequence backward. The outputs from both sets of LSTM cells are concatenated in each phase, crafting a richer representation that encompasses context from both past and future data points.

In this paper, the utilised CNN is one-dimensional and comprises a convolutional layer, pooling later, and a fully connected layer (Zang et al., 2020). The CNN performs convolution operations and pooling operations to capture implicit features from input data. Thereafter, the features extracted are merged and fed into the fully connected layer (Zang et al., 2020). Finally, an activation function is applied to ensure that the output of the neuron is nonlinear. Convolutional layers possess multiple convolutional kernels that are convolved with input information to capture hidden features and form feature maps (Li et al., 2016). A non-linear activation function is utilised to transform feature maps into the output of the convolutional layer. The convolutional layer is expressed as shown below:

$$c_i = f\left(w_i * x_i + b_i\right) \tag{1}$$

where x_i represents the input of the convolution layer, c_i is the i^{th} output feature map, w_i is a weight matrix, b_i is the bias vector, and $f(\cdot)$ denotes the activation function. The rectified linear unit (ReLU) function, which is shown below, is often used as the activation function of CNNs:

$$c_i = f(h_i) = max(0, h_i) \tag{2}$$

where, h_i is the component of feature maps.

Max pooling is implemented to cut down the dimensions of feature maps and prevent over-fitting. This is done by computing the maximum value of an assigned area in feature maps as demonstrated in equations (3) and (4).

$$\gamma(c_i, c_{i-1}) = max(c_i, c_{i-1}) \tag{3}$$

$$p_i = \gamma(c_i, c_{i-1}) + \beta_i \tag{4}$$

where $\gamma(\cdot)$ is the max pooling subsampling function, p_i denotes the output of the maxpooling layer, β_i is the bias.

The fully connected layer computes the final output vector as demonstrated in the equation below:

$$y_i = f(t_i p_i + \delta_i) \tag{5}$$

where y_i denotes the final output vector, δ_i represents bias, and t_i denotes the weight matrix.

In the proposed algorithm, CNN and BiLSTM are combined to improve performance. Therefore, the output of the CNN model is fed into the BiLSTM model. The basic architecture of the BiLSTM model comprises the outputs of forward and backward hidden layers. The outputs of the forward layer and hidden sequences are calculated iteratively using inputs in an orderly fashion (Staffini, 2023). The same is done for the backward layer and hidden sequences but in the opposite direction. These iterations are done using LSTM. Therefore, the additional BiLSTM layer produces an output vector in which each element is calculated using the equation (6):

$$y_t = \sigma(\vec{h}_t, \overleftarrow{h}_t) \tag{6}$$

where $\vec{h}_t$ and $\overleftarrow{h}_t$ are the outputs of forward and backward hidden layers, respectively, and the σ function is utilised to couple the two sequences. The σ function can be a summation, mutilation, average, or concatenation function.

During training, the algorithm was fed with labelled sequences of network traffic data, where each sequence is associated with a label indicating whether it contains a DDoS attack or not. The algorithm's weights were updated through backpropagation to minimise the classification error using a loss function. During inference, the trained algorithm was used to classify new sequences of network traffic data. The algorithm assigned a class label ('normal' or 'DDoS attack') to each input sequence based on its learned patterns and features.

3.3 Distributed DDoS-IoT detection approach

The framework implementation is illustrated in Figure 3. The experimental approach included model training, integration into an IoT network infrastructure, and the detection of attacks as an already pre-processed dataset was leveraged.

Figure 3 DDoS-IoT detection experimental framework (see online version for colours)

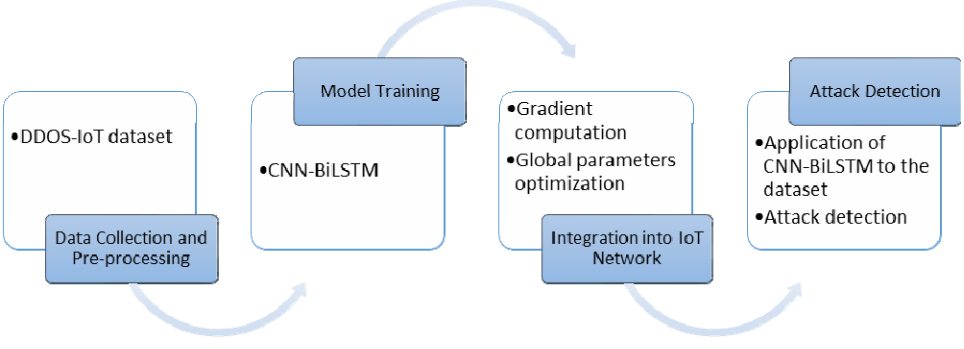
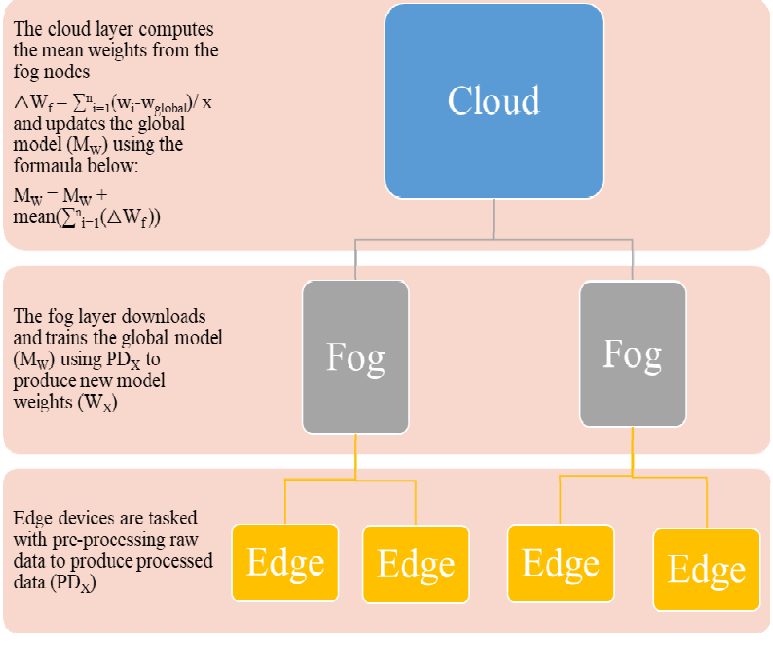


Figure 4 Distributed DDoS-IoT detection framework (see online version for colours)



The training and testing of the CNN-BiLSTM framework using the normal MQTT data derived from the DoS/DDoS-MQTT-IoT dataset were the initial steps. Next, the integration of the detection system into the IoT architecture adopted the three layers of most IoT deployments: edge, fog, and cloud layers. While the edge layer consists of smartphones, smart cars, and other limited-resource devices, the fog layer acts as an intermediary computing layer. Devices in the fog layer have considerable computing and

storage capabilities, and their utilisation increases network performance and minimises latency. The cloud layer offers vast storage capabilities and high-performance servers. Similarly, the deep learning-based attack detection architecture had three layers: the cloud layer had a global model that utilises gradient weights to detect DDoS attacks, the fog layer provided the gradient weights and trained the machine learning model, and the edge layer provided pre-processed data to fog nodes. At the start of the process, each edge device downloaded the global model from the cloud layer for training. As such, the global model was authenticated and updated in the cloud layer.

Figure 4 illustrates the developed DDoS-IoT detection framework. It combines pre-processing with the distributed nature of IoT systems to enhance the accuracy of DDoS attack detection.

The update formula $MW = MW + mean\left(\sum_{i=1}^{n} \Delta W_f\right)$ is an important aspect of this framework as it represents an update rule commonly used in the context of neural networks, where MW represents the current weights of a model and ΔW_f represents the change in weights during each iteration.

3.4 Evaluation

The evaluation process aimed to establish the performance of the proposed distributed system with the centralised one. To do so, the deep learning model was deployed on the server for the centralised system and multiple coordinated nodes for the distributed method. Accordingly, we varied the number of machines utilised for training the network as a function of the training accuracy. After hyper-parameter optimisations, the deep learning system had three CNN layers, two BiLSTM layers, and a kernel size of five. The model had 64 batch sizes in 100 epochs and trained with dropout to avoid overheating problems. The LSTM units were 128 and the number of filters was 64. The learning rate was 0.0001 and the Adam optimiser was utilised.

The confusion matrix was utilised to assess the framework's predictive capabilities. The confusion matrix provided a breakdown of the predicted labels and true labels, enabling the calculation of various evaluation metrics such as accuracy, precision, recall, and F1-score. The accuracy metric determined the overall correctness of the predictions, while precision measured the proportion of true DDoS attack instances among the predicted DDoS attacks. Recall evaluated the framework's ability to identify DDoS attacks correctly, and the F1-score provided a balanced measure of the framework's performance. The analysis also included comparing the performance of the proposed model to other machine learning models and performing cross-validation.

4 Findings and discussion

The CNN-BiLSTM model was trained using samples of benign and DDoS traffic derived from the DoS/DDoS-MQTT-IoT dataset and then utilised to classify DDoS attacks using Python code. TensorFlow and Keras were utilised for deep learning tasks. The weighted moving average update formula for model weights was done at the cloud layer in line with the distributed approach of the framework using fog data. The updated model was utilised iteratively to evaluate the detection effectiveness of the distributed framework. The confusion matrix, which includes accuracy, precision, recall, and F-measure, was

created to assess the CNN-BiLSTM model under centralised and distributed approaches. Table 3 gives an overview of how each of the two approaches (distributed and centralised) compared.

Table 3 Classification results

	Accuracy	*Error*	*Precision*	*Recall*	*f-1 score*
CONNECT flooding attack (BF_DDoS)					
Centralised detection	0.9845	0.0239	0.917	0.95	0.964
Distributed detection	0.9986	0.0121	0.99	0.995	0.993
Delayed connect flooding attack (Delay_DDoS)					
Centralised detection	0.9776	0.0437	0.949	0.962	0.965
Distributed detection	0.9976	0.0131	0.991	0.993	0.993
Invalid subscription flooding attack (Sub_DDoS)					
Centralised detection	0.9676	0.0457	0.966	0.964	0.975
Distributed detection	0.9926	0.0141	0.991	0.993	0.989
CONNECT flooding with WILL payload attack (WILL_DDoS)					
Centralised detection	0.9771	0.0146	0.961	0.973	0.985
Distributed detection	0.9973	0.0145	0.988	0.993	0.990
TCP SYN flooding attack (SYN_DDoS)					
Centralised detection	0.9476	0.0601	0.953	0.967	0.955
Distributed detection	0.9977	0.0139	0.991	0.99	0.990

Figure 5 Detection performance (see online version for colours)

Based on the Figures 3 and 4, the distributed algorithm exhibits better performance as compared to a centralised one in all four metrics: accuracy, error, precision, recall, and f-1 score. Figure 5 illustrates the differences in average performance between centralised and distributed detection. As can be seen, the distributed detection exhibits better accuracy, precision, and recall. It also has a lower error rate.

The overall performance of the distributed and centralised models is shown in Table 4.

Table 4 Overall detection performance

	Accuracy	*Error*	*Precision*	*Recall*	*f-1 score*
Centralised detection	0.97088	0.0376	0.9492	0.9632	0.9688
Distributed detection	0.99676	0.0135	0.9902	0.9928	0.991

We then assessed the performance of the proposed CNN-BiLSTM model in predicting DDoS attacks by comparing it to conventional machine learning models. As can be seen in Table 5, the proposed model performs better than the other models.

Table 5 Comparison of the proposed model to other ML models

	Accuracy	*Precision*	*Recall*	*f-1 score*
Centralised detection	97.09	94.92	96.32	96.88
Distributed detection	99.68	99.02	99.28	99.10
XGBoost	76.32	76.14	76.01	76.32
Support vector machine	74.15	74.15	74.11	73.98
Random forest	75.42	75.10	75.12	75.12
KNN	71.01	71.01	70.87	71.16
CNN	85.05	84.15	84.99	83.44
BiLSTM	83.10	86.43	82.04	81.34
RNN	89.12	93.08	89.01	89.26

Additionally, as demonstrated in Table 5, both the centralised and distributed deep learning models outperformed XGBoost, support vector machine (SVM), RF, and KNN. These models were examined in a centralised detection format. XGBoost yielded a score of 76% for four measures: precision, recall, F1-score, and accuracy. This poor performance can be attributed to the fact that XGBoost is difficult to tweak, requires a longer training period, and is vulnerable to overfitting if the data utilised is noisy (Khattak et al., 2021). SVM had an accuracy, precision, recall, and f-1 score of 74% for each of all these measures. According to Khan et al. (2021), SVM performs poorly because it requires longer training times, performs expensive computations, exhibits a lot more complexity, and requires larger size requirements for training and testing. Concerning the RF, the accuracy, precision, recall, and f-1 score were 75%, 75%, 75%, and 75%, respectively. According to Ullah et al. (2021), the RF performs poorly since its legitimate predictions take time, it favours comparable sets of related attributes over bigger sets, and it works poorly with categorical data. Finally, KNN ranked poorly in performance because it had a score of 71% for recall, precision, accuracy, and f-1 score. This is because KNN is sensitive to noisy and irrelevant data and requires a lot of time

when working with large datasets (Ullah et al., 2021). As can be seen in these outcomes, deep learning performs far better than conventional machine learning models.

The performance of the proposed model also outperformed other deep learning models, primarily CNN, BiLSTM, and RNN. For CNN, the accuracy, precision, recall, and f-1 score was 85%, 84%, 85%, and 83%, respectively. Although this performance is better than that of traditional machine learning models, it was inferior to the proposed decentralised model. CNN, on its own, does not produce optimal outputs as it does not work well with textual data and requires a large dataset. Regarding BiLSTM, the accuracy, precision, recall, and f-1 score was 83%, 86%, 82%, and 81%, respectively. BiLSTM also underperformed as it tends to underperform when it comes to extracting features. Lastly, RNN achieved suboptimal performance in terms of accuracy, precision, recall, and f1-score. This could be attributed to the fact that RNN models are unable to manage long-term sequencing. Their inability to retain information for long periods means that they are not suited to DDoS detection tasks. As was the case with conventional machine learning models, the performance of deep learning models was lower than that of the CNN-BiLSTM model. It is also imperative to note that the experiment did not do tests on ensemble algorithms, which could exhibit better performance.

The study also encompassed performing cross-validation of the proposed model to estimate how it will perform in practice. Specifically, k-fold validation, which entails splitting the data into k folds or subnets and performing training on all subnets but leaving one out for evaluation purposes. In each iteration, different subnets are reserved for the testing. In this study, K = 10 was used, which means that the data was split into ten subnets, with the subnets having the same magnitude. Therefore, 9-fold subnets were used for training, and 1-fold for testing. The results for the distributed model can be seen in Table 6.

Table 6 Sample randomised 10-fold cross-validation

	Accuracy	*Precision*	*Recall*	*f-1 score*
Fold-1	99.81	99.12	99.28	99.12
Fold-2	99.72	98.73	99.11	99.03
Fold-3	99.73	99.45	99.23	99.55
Fold-4	99.61	99.44	99.41	99.41
Fold-5	99.54	99.23	99.12	99.23
Fold-6	99.41	98.79	98.05	98.96
Fold-7	99.89	99.43	98.99	99.33
Fold-8	99.97	98.01	99.34	98.01
Fold-9	99.23	98.91	99.29	98.71
Fold-10	99.34	99.04	99.5	99.24
Mean	99.625	99.015	99.132	99.059

The evaluations done above suggest that the proposed model can be effective for detecting DDoS attacks. This improved performance could be attributed to the fact that DDoS attacks might attack a specific segment of the network without having a major effect on the whole network. Therefore, while the network might be performing within the normal ranges, some segments might be unavailable. In the proposed distributed

approach, both edge and fog layers play a role in the detection process hence allowing for localised DDoS identification. According to Febro et al. (2019), distributed detection can enhance scalability, allowing the network to accommodate growing network traffic while addressing evolving attack tactics. Essentially, decentralisation enhances redundancy and addresses issues related to geographical diversity. More importantly, distributed detection provides a more comprehensive view of the network, making it easier to identify subtle attack patterns or multiple attack vectors that might go unnoticed in a centralised system.

A possible drawback of the proposed algorithm is that it can increase the computational load for IoT devices situated at the edge and fog layers. In this framework, edge devices would be expected to process raw data to produce processed data. Similarly, devices at the fog layer would be required to download and train the global model using the processed data to produce new model weights that will then be used to update the global model. At the same time, IoT devices are often resource-restrained when it comes to both storage and computational ability. Accordingly, addressing this element is imperative to the successful implementation of the algorithm. Furthermore, the proposed distributed detection algorithm should be part of a comprehensive DDoS mitigation strategy that includes both detection and mitigation techniques. Examples include traffic filtering, traffic diversion, and load balancing (Salva-Garcia et al., 2018). Combining these approaches can help organisations effectively protect their network infrastructure from DDoS attacks.

5 Conclusions

The study demonstrated the effectiveness of deep learning, especially the CNN-BiLSTM model, in DDoS distributed detection situations. Indeed, deep learning models can learn hierarchical representations automatically in data, which is imperative in anomaly detection. Similarly, CNN-BiLSTM can extract relevant features from complex and unstructured raw network traffic data, hence being able to adapt to changing attack patterns without the need for manual feature engineering. Additionally, in DDoS detection, network traffic data is sequential, and the utilisation of BiLSTM can help with capturing sequential dependencies in the data, enabling the model to detect subtle anomalies. Deep learning models are also robust in that they can generalise well to detect new and unseen attacks that may have different characteristics from known attacks.

Deep learning models possess the potential to scale to large datasets and network traffic volumes. This is crucial for handling the vast amount of data generated in network environments, making them suitable for real-world deployments. Furthermore, deep learning models can automate the detection process, allowing for real-time or near-real-time response to DDoS attacks. Automated detection and mitigation are imperative for minimising downtime and service disruptions. Besides CNNs inherently support parallel processing, which can speed up the detection process in the event that the network traffic is massive. Overall, deep learning models excel at anomaly detection, which is essential for DDoS detection since DDoS attacks often involve unusual and unexpected patterns in network traffic.

The findings also illustrated the superiority of distributed DDoS detection versus centralised detection. The detection accuracy is better as the algorithm is able to take into account local conditions in a given fog layer. Additionally, distributed detection can provide lower latency in detecting events and anomalies because data processing and

decision-making occur closer to the source of the data. This is crucial for applications where real-time or near-real-time responses are required. Furthermore, a distributed approach can be more scalable as it distributes the processing load across multiple IoT devices or edge nodes and fog layers (Ali et al., 2023). This can make it easier to handle a large number of IoT devices within a network without overloading a centralised server.

A major strength of this study is that it utilised deep learning combined with a real-world dataset on DDoS attacks in IoT settings to evaluate the distributed algorithm. The deep learning model, CNN-BiLSTM, has been demonstrated to be effective in detecting DDoS attacks. The dataset also included various forms of DDoS attacks, which enabled the evaluation of the proposed algorithm against different attacks. However, the study had a limitation in that the dataset was not categorised into fog networks. To study the distributed model, we had to divide the data into hypothetical fog nodes to evaluate the distributed framework. The study also made use of a single dataset, which calls into question the generalisability of the findings. Accordingly, future research ought to collect data at the fog layer rather than at the cloud layer to evaluate the distributed framework in detail. Additionally, the proposed model ought to be tested on different datasets.

References

Aboubakar, M., Kellil, M. and Roux, P. (2022) 'A review of IoT network management: current status and perspectives', *Journal of King Saud University-Computer and Information Sciences*, Vol. 34, No. 7, pp.4163–4176.

Adedeji, K.B., Abu-Mahfouz, A.M. and Kurien, A.M. (2023) 'DDoS attack and detection methods in internet-enabled networks: concept, research perspectives, and challenges', *Journal of Sensor and Actuator Networks*, Vol. 12, No. 4, p.51, https://doi.org/10.3390/jsan12040051.

Aktar, S. and Nur, A.Y. (2023) 'Towards DDoS attack detection using deep learning approach', *Computers and Security*, Vol. 129, No. 2023, p.103251, https://doi.org/10.1016/j.cose.2023.103251.

Alatram, A., Sikos, L.F., Johnstone, M., Szewczyk, P. and Kang, J.J. (2023) 'DoS/DDoS-MQTT-IoT: a dataset for evaluating intrusions in IoT networks using the MQTT protocol', *Computer Networks*, Vol. 231, No. 2023, p.109809, https://doi.org/10.1016/j.comnet.2023.109809.

Alghazzawi, D., Bamasag, O., Ullah, H. and Asghar, M.Z. (2021) 'Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection', *Applied Sciences*, Vol. 11, No. 24, p.11634, https://doi.org/10.3390/app112411634.

Ali, T. E., Chong, Y. W., and Manickam, S. (2023) 'Machine learning techniques to detect a DDoS attack in SDN: a systematic review', *Applied Sciences*, Vol. 13, No. 5, p.3183, https://doi.org/10.3390/app13053183.

Aswad, F.M., Ahmed, A.M.S., Alhammadi, N.A.M., Khalaf, B.A. and Mostafa, SA. (2023) 'Deep learning in distributed denial-of-service attacks detection method for internet of things networks', *Journal of Intelligent Systems*, Vol. 32, No. 1, https://doi.org/10.1515/jisys-2022-0155.

Bahashwan, A.A., Anbar, M., Manickam, S., Al-Amiedy, T.A., Aladaileh, M.A. and Hasbullah, I.H. (2023) 'A systematic literature review on machine learning and deep learning approaches for detecting DDoS attacks in software-defined networking', *Sensors*, Vol. 23, No. 9, p.4441.

Banitalebi, D.A., Soltanaghaei, M. and Boroujeni, F.Z. (2021) 'The DDoS attacks detection through machine learning and statistical methods in SDN', *The Journal of Supercomputing*, Vol. 77, No. 18, pp.2383–2415.

Bhattacharjya, A. (2022) 'A holistic study on the use of blockchain technology in CPS and IoT architectures maintaining the CIA triad in data communication', *International Journal of Applied Mathematics and Computer Science*, Vol. 32, No. 3, pp.403–413.

Di Nocera, F. and Tempestini, G. (2022) 'Getting rid of the usability/security trade-off: a behavioral approach', *Journal of Cybersecurity and Privacy*, Vol. 2, No. 2, pp.245–256.

Diro, A.A. and Chilamkurti, N. (2018) 'Distributed attack detection scheme using deep learning approach for internet of things', *Future Generation Computer Systems*, Vol. 82, No. 1, pp.761–768.

Febro, A., Xiao, H. and Spring, J. (2019) 'Distributed SIP DDoS defense with P4', in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, April, pp.1–8, IEEE.

Fortune Business Insights (2023) *Internet of Things (IoT) Market* [online] https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307 (accessed 11 October 2023).

Gaurav, A. and Singh, A.K. (2017) 'Super-router: a collaborative filtering technique against DDoS attacks', in *Advanced Informatics for Computing Research: First International Conference, ICAICR 2017*, Jalandhar, India, March 17–18, 2017, Revised Selected Papers, pp.294–305, Springer Singapore.

HaddadPajouh, H., Dehghantanha, A., Parizi, R.M., Aledhari, M. and Karimipour, H. (2021) 'A survey on internet of things security: Requirements, challenges, and solutions', *Internet of Things*, Vol. 14, p.100129, https://doi.org/10.1016/j.iot.2019.100129.

Halder, R. and Chatterjee, R. (2020) 'CNN-BiLSTM model for violence detection in smart surveillance', *SN Computer Science*, Vol. 1, No. 4, p.201.

Khan, A., Khattak, A.M., Asghar, M. Z., Naeem, M. and Din, A.U. (2021) 'Playing first-person perspective games with deep reinforcement learning using the state-of-the-art game-AI research platforms', *Deep Learning for Unmanned Systems*, pp.635–667.

Khattak, A., Asghar, M.Z., Ishaq, Z., Bangyal, W.H. and Hameed, I.A. (2021) 'Enhanced concept-level sentiment analysis system with expanded ontological relations for efficient classification of user reviews', *Egyptian Informatics Journal*, Vol. 22, No. 4, pp.455–471.

Kumar, P., Kumar, R., Gupta, G.P. and Tripathi, R. (2021) 'A distributed framework for detecting DDoS attacks in smart contract-based Blockchain-IoT Systems by leveraging Fog computing', *Transactions on Emerging Telecommunications Technologies*, Vol. 32, No. 6, p.e4112.

Lawal, M.A., Shaikh, R.A. and Hassan, S.R. (2021) 'A DDoS attack mitigation framework for IoT networks using fog computing', *Procedia Computer Science*, Vol. 182, No. 8, pp.13–20.

Li, Y., Hao, Z. and Lei, H. (2016) 'Survey of convolutional neural network', *Journal of Computer Applications*, Vol. 36, No. 9, p.2508.

Lu, G., Liu, Y., Wang, J. and Wu, H. (2023) 'CNN-BiLSTM-Attention: a multi-label neural classifier for short texts with a small set of labels', *Information Processing and Management*, Vol. 60, No. 3, p.103320.

Lygerou, I., Srinivasa, S., Vasilomanolakis, E., Stergiopoulos, G. and Gritzalis, D. (2022) 'A decentralized honeypot for IoT Protocols based on Android devices', *International Journal of Information Security*, Vol. 21, No. 6, pp.1211–1222.

Mishra, N. and Pandya, S. (2021) 'Internet of things applications, security challenges, attacks, intrusion detection, and future visions: a systematic review', *IEEE Access*, Vol. 9, No. 2021, pp.59353–59377.

Mtowe, D.P. and Kim, D.M. (2023) 'Edge-computing-enabled low-latency communication for a wireless networked control system', *Electronics*, Vol. 12, No. 14, p.3181, https://doi.org/10.3390/electronics12143181.

Noaman, M., Khan, M.S., Abrar, M.F., Ali, S., Alvi, A. and Saleem, M.A. (2022) 'Challenges in integration of heterogeneous internet of things', *Scientific Programming*, Vol. 2022, https://doi.org/10.1155/2022/8626882.

Roopak, M., Tian, G.Y. and Chambers, J. (2019) 'Deep learning models for cyber security in IoT networks', in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, January, pp.452–457.

Salva-Garcia, P., Alcaraz-Calero, J.M., Wang, Q., Bernabe, J.B. and Skarmeta, A. (2018) '5G NB-IoT: efficient network traffic filtering for multitenant IoT cellular networks', *Security and Communication Networks*, Vol. 2018, No. 2, pp.1–21.

Sanjuan, E.B., Cardiel, I.A., Cerrada, J.A. and Cerrada, C. (2020) 'Message queuing telemetry transport (MQTT) security: a cryptographic smart card approach', *IEEE Access*, Vol. 8, No. 2020, pp.115051–115062.

Sanzana, M.R., Maul, T., Wong, J.Y., Abdulrazic, M.O.M. and Yip, C.C. (2022) 'Application of deep learning in facility management and maintenance for heating, ventilation, and air conditioning', *Automation in Construction*, Vol. 141, No. 3, p.104445.

Sethi, P. and Sarangi, S.R. (2017) 'Internet of things: architectures, protocols, and applications', *Journal of Electrical and Computer Engineering*, Vol. 2017, https://doi.org/10.1155/2017/9324035.

Singh, M.P. and Bhandari, A. (2020) 'New-flow based DDoS attacks in SDN: taxonomy, rationales, and research challenges', *Computer Communications*, Vol. 154, No. 2020, pp.509–527.

Staffini, A. (2023) 'A CNN–BiLSTM architecture for macroeconomic time series forecasting', *Engineering Proceedings*, Vol. 39, No. 1, p.33, https://doi.org/10.3390/engproc2023039033.

Thoutam, V. (2021) 'An overview on the reference model and stages of IoT architecture', *International Journal of Information Technology and Computer Engineering (IJITC)*, ISSN: 2455-5290, Vol. 1, No. 1, pp.7–14.

Tian, Y., Zhang, Y. and Zhang, H. (2023) 'Recent advances in stochastic gradient descent in deep learning', *Mathematics*, Vol. 11, No. 3, p.682, https://doi.org/10.3390/math11030682.

Ullah, H., Ahmad, B., Sana, I., Sattar, A., Khan, A., Akbar, S. and Asghar, M.Z. (2021) 'Comparative study for machine learning classifier recommendation to predict political affiliation based on online reviews', *CAAI Transactions on Intelligence Technology*, Vo. 6, No. 3, pp.251–264.

Williams, P., Dutta, I.K., Daoud, H. and Bayoumi, M. (2022) 'A survey on security in internet of things with a focus on the impact of emerging technologies', *Internet of Things*, Vol. 19, p.100564, https://doi.org/10.1016/j.iot.2022.100564.

Zang, H., Liu, L., Sun, L., Cheng, L., Wei, Z. and Sun, G. (2020) 'Short-term global horizontal irradiance forecasting based on a hybrid CNN-LSTM model with spatiotemporal correlations', *Renewable Energy*, Vol. 160, No. C, pp.26–41.