Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.072493

ARTICLE

FRF-BiLSTM: Recognising and Mitigating DDoS Attacks through a Secure Decentralized Feature Optimized Federated Learning Approach

Sushruta Mishra[1], Sunil Kumar Mohapatra[2], Kshira Sagar Sahoo[3], Anand Nayyar[4] and Tae-Kyung Kim[5,*]

[1]School of Computer Engineering, Kalinga Institute of Industrial Technology (Deemed to be University), Bhubaneswar, 751024, Odisha, India

[2]Department of Computer Science and Engineering, Centurion University of Technology and Management, Bhubaneswar, 761211, Odisha, India

[3]Department of Computer Science and Engineering, SRM University-AP, Amaravati, 522240, Andhra Pradesh, India

[4]School of Computer Science, Duy Tan University, Da Nang, 550000, Vietnam

[5]Department of Management Information Systems, Chungbuk National University, Cheongju-Si, 28644, Chungcheongbuk-Do, Republic of Korea

*Corresponding Author: Tae-Kyung Kim. Email: kimtk@chungbuk.ac.kr

Received: 28 August 2025; Accepted: 21 October 2025; Published: 12 January 2026

ABSTRACT: With an increase in internet-connected devices and a dependency on online services, the threat of Distributed Denial of Service (DDoS) attacks has become a significant concern in cybersecurity. The proposed system follows a multi-step process, beginning with the collection of datasets from different edge devices and network nodes. To verify its effectiveness, experiments were conducted using the CICDoS2017, NSL-KDD, and CICIDS benchmark datasets alongside other existing models. Recursive feature elimination (RFE) with random forest is used to select features from the CICDDoS2019 dataset, on which a BiLSTM model is trained on local nodes. Local models are trained until convergence or stability criteria are met while simultaneously sharing the updates globally for collaborative learning. A centralised server evaluates real-time traffic using the global BiLSTM model, which triggers alerts for potential DDoS attacks. Furthermore, blockchain technology is employed to secure model updates and to provide an immutable audit trail, thereby ensuring trust and accountability among network nodes. This research introduces a novel decentralized method called Federated Random Forest Bidirectional Long Short-Term Memory (FRF-BiLSTM) for detecting DDoS attacks, utilizing the advanced Bidirectional Long Short-Term Memory Networks (BiLSTMs) to analyze sequences in both forward and backward directions. The outcome shows the proposed model achieves a mean accuracy of 97.1% with an average training delay of 88.7 s and testing delay of 21.4 s. The model demonstrates scalability and the best detection performance in large-scale attack scenarios.

KEYWORDS: Bi-directional long short-term memory network; distributed denial of service (DDoS); cybersecurity; federated learning; random forest

1 Introduction

In an era where digital environments influence every aspect of our lives, cyber threats, such as Distributed Denial of Service (DDoS) attacks, are becoming increasingly common and sophisticated. DDoS attacks pose a continuous and evolving threat to network stability, leading to disruptions and financial losses. These attacks aim to overload online services by flooding them with traffic. Conventional DDoS detection techniques have struggled to keep pace with the ever-evolving strategies employed by hostile actors.

According to a 2023 report, there was a significant 85% increase in network-layer DDoS attacks in 2022, with approximately 80 petabytes of malicious traffic flooding the network. DDoS attacks attempt to bring down online services by flooding their network infrastructures. They are distinguished by their distributed structure, which follows the enormous number of requests they produce. It is a diverse threat, with objectives ranging from money extortion to hacktivism. These attacks are dynamic and ever-evolving. Modern and dynamic detection methods are crucial as cyber attackers continue to refine their tactics. Edge devices are essential for ensuring uninterrupted service and maintaining network integrity. They must be protected from DDoS attacks. As the initial line of defence, edge devices are vulnerable to DDoS attacks because they are directly exposed to incoming traffic. Unregulated DDoS attacks targeting edge devices may result in service interruptions, network congestion, and, in extreme situations, total unavailability. By securely detecting and suppressing malicious traffic, DDoS mitigation of edge devices keeps them from advancing further into the network architecture. This preemptive approach reduces the possibility of collateral damage to downstream systems while also protecting the functionality and dependability of online services. Blockchain technology is being increasingly examined for its cybersecurity applications due to its ability to ensure immutability, transparency, and trustworthiness. In the context of distributed DDoS detection, blockchain can securely record model updates and authenticate contributions from nodes, thereby enhancing collaborative learning. DDoS attacks pose a significant threat to network stability, particularly at the edge, where rapid detection is crucial. While Machine Learning–Based (ML-Based) solutions provide flexibility, their centralised design raises privacy and scalability issues, leading to the adoption of federated learning (FL). As an effective tool in the cybersecurity arsenal, Machine Learning (ML) enables networks to learn and adapt based on patterns identified in data. ML algorithms can be trained to identify abnormal traffic trends that indicate an ongoing attack. Conventional ML models are centralized, which raises privacy problems and makes it difficult to use them in situations involving personal information. With its decentralized method for model training, Federated Learning (FL) provides an effective method for privacy concerns linked with centralized ML. The model in the FL paradigm is trained simultaneously over different edge devices. Each user can participate in the model training while maintaining the privacy of their local data due to the decentralized technique. FL, which allows collective model training without sacrificing data privacy, has demonstrated effectiveness in several application domains.

An innovative approach used in this study to enhance DDoS detection systems' performance is the integration of random forest and BiLSTM within an FL environment. The adaptable ensemble learning technique, random forest, is excellent at selecting relevant features from a dataset. The interaction within these elements enables the model to identify intricate patterns characteristic of DDoS attacks. BiLSTM is a form of Recurrent Neural Network (RNN) intended to collect temporal dependencies in sequential data bi-directionally. The motivation behind this research stems from the need for a dynamic solution to address the challenges of centralized methods, specifically in enhancing privacy, accuracy, and scalability through the use of FL for DDoS detection. FL allows for the accommodation of input diversity among network nodes, resulting in a more resilient and universal model, whereas FRF-BiLSTM's integration facilitates the collection of complex periodic connections.

The method used to detect DDoS is dynamic and can detect dynamic attack behaviours, ensuring that the detection system remains resistant to attacks that evolve. The decentralized approach used by FL protects individual data by enabling model training on local devices. By ensuring that private data remains within individual nodes, this distributed learning paradigm mitigates concerns about privacy related to centralised data collection. The model is better able to adapt to various traffic patterns because it can be trained on multiple network nodes. The detection method becomes accurate and efficient in addressing the distinct qualities of every network node by combining information from several sources. Despite advances

in ML and FL for DDoS detection, three challenges remain. First, high-dimensional data requires effective feature selection to prevent overfitting and reduce costs. Second, trust and security concerns arise when aggregating updates, risking tampering or poisoning. Finally, many models fail to capture bidirectional temporal dependencies vital for identifying subtle, evolving DDoS patterns. Therefore, this research proposes a federated learning framework that combines FRF-BiLSTM as a novel approach to enhance cybersecurity defences against the constantly evolving DDoS attack landscape.

Objectives of the Paper

The objectives of the paper are:

- To assess and mitigate DDoS network attacks while addressing the privacy concerns associated with centralized models.
- To develop a novel FL approach for DDoS detection, i.e., 'FRF-BiLSTM' model, using distributed data sources to train a global BiLSTM predictive model.
- Feature selection is performed on the dataset using a combined random forest and recursive feature elimination method. In addition, local BiLSTM models are trained, and updates are aggregated through FL on the global model.
- To deploy a global BiLSTM model on a centralized server for real-time traffic evaluation and detection of DDoS attacks, and to integrate blockchain, ensuring secure storage, verification, and decentralised management of global BiLSTM model updates.

Organization of Paper

The rest of the paper is organized as: Section 2 enlighens literature review. Section 3 highlights proposed methodology. Section 4 presents the results and analysis. And, finally, Section 5 concludes the paper with future scope.

2 Literature Survey

Several significant studies are currently being conducted in this domain. Priya et al. [1] proposed a self-operating ML model using random forest, Naive Bayes, and K-Nearest Neighbors (KNN), achieving 98.5% accuracy that can run on any commodity hardware. The model addressed the limitations of previous models, which often required specific protocols and substantial features for detection, by effectively identifying various forms of DDOS attacks. Using the Netflow feature selection and machine learning, Hou et al. [2] proposed a real-time detection method for DDoS attacks with an accuracy of 99%. The proposed technique performed better against stealthy and diverse DDoS attacks. Ujjan et al. [3] proposed a method that made use of deep learning, adaptive polling, SDN, sFlow, and Snort IDS to mitigate DDOS attacks in Internet of Things networks. This all-inclusive technology delivered increased detection accuracy, reduced processing head, and customization for a wide range of IoT network parameters. Huang et al. [4] introduced two strategies to mitigate the threat to ML systems from adversarial samples, specifically in detecting DDOS probability-weighted packet saliency attacks and genetic attacks. The effectiveness of these techniques in dodging the DDOS detectors was demonstrated by empirical evidence. Aydın et al. [5] introduced an LSTM-CLOUD-based DDoS detection and prevention system in various cloud environments. The proposed method achieved an accuracy of 99.83%. The proposed model detects attacks and activates defence mechanisms, highlighting the effectiveness of LSTM in ensuring information security in the cloud. Li et al. [6] proposed a Federated Intrusion Detection System (FIDS) utilizing federated learning for a Network Intrusion Detection System (NIDS). FIDS addresses different non-id datasets, which improve training stability and convergence rate by introducing prototypical weights and extracting other features.

To address the challenges in Federated Learning, Doriguzzi-Corin and Siracusa [7] presented FLAD, an adaptive approach for detecting DDoS attacks. FLAD dynamically assigns computation based on the difficulty of the attack profile, thereby outperforming other state-of-the-art algorithms in terms of accuracy and convergence time. Li et al. [8] combined federated learning with edge computing and IoT security, proposing a protocol to address networked zombies and detect malicious code. The federated learning-based detection protocol improved accuracy, further reducing mitigation time and increasing the costs of attackers within a collaborative defence alliance, as validated through comprehensive evaluations. Hayat et al. [9] proposed a multilevel DDoS mitigation approach for IoT using a blockchain-based framework. The strategy was implemented using Hyperledger Calliper, which achieved a 35% improvement in throughput, a 40% reduction in latency, and 25% better CPU performance compared to a state-of-the-art solution. Manikumar and Uma Maheswari [10] have introduced a system that combines machine learning and blockchain to prevent DDoS attacks. The research utilized KNN, decision tree, and random forest algorithms, along with blockchain's real-time analysis, to achieve 95% accuracy, thereby providing enhanced security. Rahim et al. [11] examined SDN architecture and security flaws, reviewed DDoS detection methods, including statistical, machine learning, policy, and moving target defences, assessed their strengths and limitations, and suggested future research to improve SDN security. Kumari and Jain [12] analysed attack types, security weaknesses, and mitigation strategies in insecure IoT devices to detect DDoS attacks. Ali et al. [13] proposed a review of various deep learning techniques for detecting DDoS attacks in SDN networks. Changanti et al. [14] utilised blockchain technology to mitigate DDoS attacks on SDN and IoT devices, also considering the current challenges and potential solutions that can be implemented in the future. Authors presented reviews that categorize blockchain-based DDoS solutions and explored various fields, including deployment location, network type, attacker, and other hybrid solutions, highlighting the challenges faced and the future scope of DDoS mitigation. Mustapha et al. [15] proposed a novel methodology to identify DDoS attacks that can evade conventional detection techniques by combining GAN and LSTM. Rizvi et al. [16] employed various classification algorithms, including random forest, K-nearest neighbours, and decision trees, for DDoS attack detection. The authors further utilised a genetic algorithm to select the optimal features from the dataset, thereby increasing the performance of the classifiers. H et al. [17] proposed a Deep Neural Network-Powered Bi-Directional Cross-Generative Adversarial Network (GAN) to distinguish between DDoS attacks and Flash Crowd Assaults (FCAs) on web applications. The authors used an ensemble feature selection method and a bidirectional cross-GAN. The model effectively differentiated between the two types of attacks. Jawahar et al. [18] introduced a novel system that combined machine learning and blockchain technology for the real-time detection and mitigation of DDoS attacks. The system used an Artificial Neural Network (ANN), trained on the CICDDoS2019 dataset, and also utilizes blockchain to store and block malicious IP addresses. Dogra and Taqdir [19] addressed DDoS attacks by minimizing the attack surface by reducing resource exposure to ports and applications. The proposed method employed preprocessing techniques, including normalisation, feature extraction, and optimal feature selection using KMLP. Results show a 10% improvement in packet drop ratio and a 7% increase in network lifetime, demonstrating the effectiveness of enhanced DDoS prevention measures. Cil et al. [20] focused on countering DDoS attacks using ANN-based defence models, highlighting their effectiveness in adapting to evolving attack patterns and mitigating DDoS threats. Ahmed et al. [21] explored Federated Learning in converged ICT environments, covering architectures, applications, and challenges. Authors aimed to show how FL can boost privacy, efficiency, scalability, security, and suggest strategies for future sustainable FL deployment. Table 1 reviews DDoS detection studies employing federated learning and deep learning methods.

Table 1: Summary of recent DDoS detection studies using federated learning and deep learning approaches

Author(s)	Dataset(s) Used	Methodology	Key findings
Anjum et al. [22]	CICIoT-2023	Federated Graph Neural Network (GraphFedAI) with correlation-based feature selection	High detection accuracy (noted improvement in non-IID IoT setting)
Munaweera et al. [23]	5G core simulated metrics + logs	Federated Learning, Deep Packet Inspection and Hybrid Multimodal Framework	Reported higher accuracy and quicker convergence compared to the baseline
Fotse et al. [24]	CICDDoS2019	Federated LSTM for distributed DDoS detection	Achieved 99.42% accuracy
Lee et al. [25]	Public flow-based datasets (non-IID)	Federated Learning with DBSCAN clustering and LSTM	Improved detection on the imbalanced datasets
Almadhor et al. [26]	IoT heterogeneous network data	ML Hybrid approach with adaptive thresholding	99.78% accuracy, precision at 99.80%, recall at 99.74%
Tymoshchuk et al. [27]	CICDDoS2019 & Custom	Comparative study of Deep Learning Models with BERT, RNN and CNN	Accuracy exceeds 98% for transformer-based models
Mandela and Etyang [28]	Custom flooding attack dataset	Feed-forward Neural Network	99.35% accuracy and F1 ≈ 0.99
Alfatemi et al. [29]	CICIDS/Public DDoS dataset	Deep Residual Neural Network with SMOTE oversampling	99.98% accuracy for the balanced test set
Dilworth and Gudla [30]	BCCC-cPacket-Cloud-DDoS-2024	Positive Unlabeled Learning with Ensemble ML models (XGBoost, RF, SVM)	F_1-score exceeds 98% for the ensemble

Existing works often overlook multi-attack scenarios, real-time detection on resource-constrained edge devices, and robustness against adversarial attacks in federated learning. Many models rely on imbalanced or incomplete datasets, which can cause overfitting and poor generalization. Scalability and efficiency also present challenges, particularly for heterogeneous IoT and cloud deployments. Addressing these gaps is crucial for developing privacy-preserving, adaptive, and resilient DDoS detection systems that can effectively manage real-time, multi-attack scenarios. Wang et al. [31] introducde a federated learning scheme that handles partial, low-quality data while maintaining privacy. It employed a distributed Paillier cryptosystem to safeguard participant data during training. A notable research gap exists in understanding and mitigating adversarial attacks, particularly in the context of federated learning models. Techniques to enhance the robustness of global models against such attacks are urgently needed. Scalability and adaptability to different edge environments are significant concerns, as most research focuses on a single-edge device. In many

existing models, a primary concern is the use of imbalanced datasets. Instead of focusing on significant features, studies have used a complete dataset for model training, which results in less generalization and overfitting of the model. Most of the works focus on single-attack detection, while ignoring multi-attack scenarios. Additionally, models have yet to successfully implement real-time attack detection on edge devices. Moreover, recently designed models exhibit a complex structure with lower accuracy in attack prediction, resulting in higher computational delay, which is undesirable for real-time cyber attack assessment. The proposed work addresses existing gaps by developing a privacy-preserving federated learning framework for real-time detection of multiple DDoS attacks across diverse edge and IoT networks.

3 Proposed Methodology

In this research, a novel federated learning framework is proposed that integrates random forest-based feature selection with a BiLSTM architecture for enhanced DDoS detection, as shown in Fig. 1. This framework achieves high accuracy while preserving privacy and reducing computational costs. The CICDDoS2019 dataset [32] was used to test the proposed research, which is available for download from the Canadian Institute for Cybersecurity website. This dataset includes network traffic data for thirteen different types of DDoS attacks. Each attack type is stored in a separate PCAP (packet capture) file, resulting in a total of 85 PCAP files in the dataset. The attack types are NTP, NetBIOS, LDAP, DNS, MSSQL, Portmap, UDP-Lag, WebDDoS, TFTP, UDP Flood, SYN Flood, SNMP, and SSDP. As shown in Tables 2 and 3, these types are accompanied by essential dataset columns and their meanings.

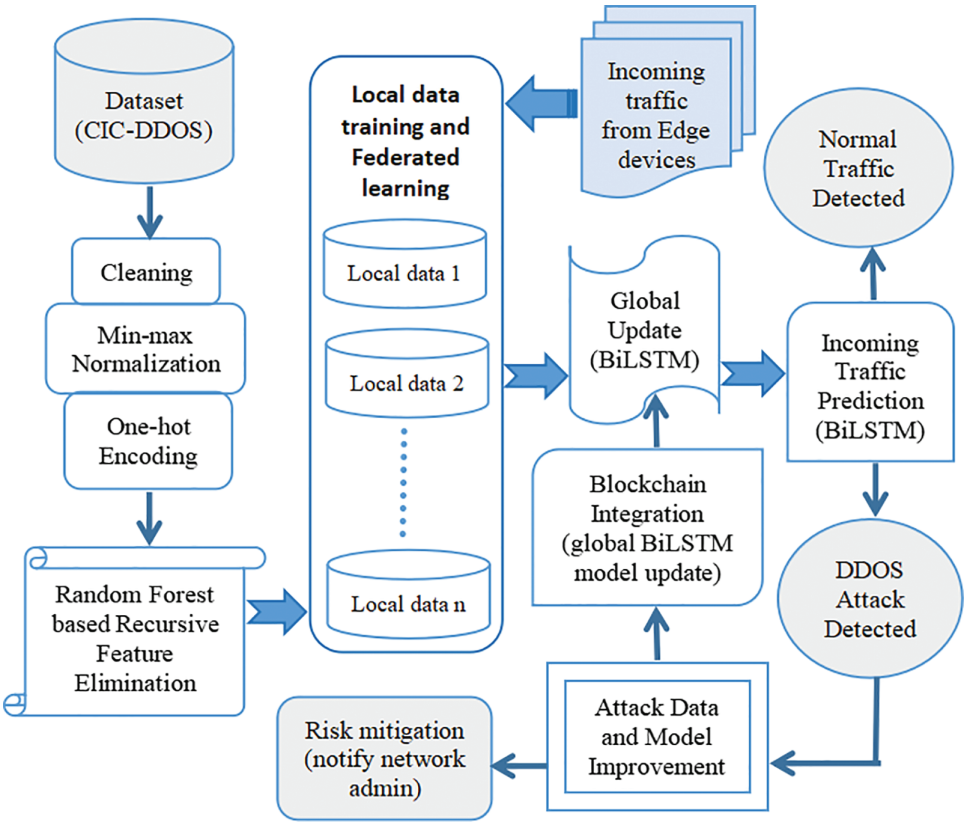


Figure 1: Proposed FRF-BiLSTM model for DDoS attack detection using federated learning

Table 2: Various types of DDoS attacks

Attack types	Descriptions
DNS LDAP MSSQL NTP NetBIOS Portmap	These types of attacks are also called reflection-based attacks. Here, the attackers do not flood the target directly with traffic, but by exploiting NTP or DNS services offered by different third-party servers. A small and manipulated request is sent by the attacker to the servers using the target's spoofed IP address, and the victim is flooded with unwanted traffic, allowing the attackers to use amplifying techniques to execute DDoS attacks.
SNMP	Targets vulnerability in the network management protocol and executes amplified attacks.
SSDP	Exploits the simple service discovery protocol and floods the target with amplified responses.
TFTP	Targets the vulnerability in Trivial File Transfer Protocol and floods the target with data, eventually crashing it.
Syn Flood	A stream of TCP SYN packets is sent to the target to exhaust the server's resources.
UDP Flood	Large amounts of UDP packets are used to overwhelm the target.
UDPLag	Strategically timed UDP packets are sent to cause network congestion and lag (like in online gaming environments).
WebDDoS	Overwhelming the web servers with HTTP requests to cause service disruptions.

Table 3: CICDDoS2019 dataset columns with descriptions

Feature name	Description	Data type
Flow ID	Unique identifier for each flow	Integer
Source IP	IP address of the flow's source	String
Destination IP	IP address of the flow's destination	String
Source port	Port number used by the source	Integer
Destination port	Port number used by the destination	Integer
Protocol	Transport protocol used (e.g., TCP, UDP, ICMP)	String
Timestamp	Time of the flow's start (in seconds since epoch)	Integer
Flow duration	Duration of the flow in microseconds	Integer
Total Fwd Packets	Total number of forward packets in the flow	Integer
Total backward packets	Total number of backward packets in the flow	Integer
Total length of fwd packets	Total length of all forward packets in the flow	Integer
Total length of bwd packets	Total length of all backward packets in the flow	Integer
Fwd packet length max	Maximum length of a forward packet in the flow	Integer
Fwd packet length min	Minimum length of a forward packet in the flow	Integer
Bwd packet length max	Maximum length of a backward packet in the flow	Integer

(Continued)

Table 3 (continued)

Feature name	Description	Data type
Bwd packet length min	Minimum length of a backward packet in the flow	Integer
Flow Bytes/s	Flow bytes per second	Float
Flow Packets/s	Flow packets per second	Float

The proposed methodology begins with the CICDDoS2019 dataset, which comprises labelled traffic data for various DDoS attacks. The raw PCAP files are preprocessed to extract structured features, followed by applying a Random Forest-based feature selection method to identify the most relevant attributes. These selected features are then used to train a Bidirectional Long Short-Term Memory (BiLSTM) model, capable of capturing temporal patterns in the traffic data. To maintain data privacy and minimize central processing, the model is trained in a federated learning (FL) environment, where edge devices train locally and share only model updates. Blockchain technology is integrated to secure and verify these updates, ensuring data integrity and transparency. The trained system monitors incoming traffic in real-time, classifying it as normal or DDoS-related, and triggers risk mitigation actions when an attack is detected. The proposed model architecture is illustrated in Fig. 1.

The challenge of class imbalance in DDoS detection has been addressed by incorporating class-weighted loss functions during local model training, ensuring that minority attack classes are given higher importance. Additionally, it is planned to integrate oversampling techniques (e.g., SMOTE) at the client level in future work to further mitigate class skew in the federated learning process. During the model development, after data collection, the next step is data pre-processing, which involves data cleaning and normalization. Data cleaning removes unnecessary data, improves productivity, and decision quality. This includes addressing outliers by deleting rows with missing values and imputing missing data using respective features. Data normalization is done using Eq. (1). One-hot encoding converts features to numeric form, followed by min-max normalization to scale features to (0, 1).

$$a_{i,j} = \frac{a_{i,j} - \min\left(a_{i,j}\right)}{\max\left(a_{i,j}\right) - \min\left(a_{i,j}\right)} \tag{1}$$

where '$a_{i,j}$' represents the eigenvalues of row i and column j.

After preprocessing the data, the next step is to select the features from the CICDDoS2019 dataset. In the proposed research, recursive feature elimination with cross-validation (RFECV) is employed in conjunction with the random forest ensemble method for feature extraction. The pseudo-code is shown in Algorithm 1.

Algorithm 1: Random forest based recursive feature elimination

Let a be the dataset with b variables and a binary outcome.
Let b be a ranked list of variables, ordered by their relevance.
Let c be the process of finding the optimal values for tuning parameters of the Random Forest model.
Let d be the process of training the random forest model.
Let e be initialized to b
While e ≥ 2, do:
Step 1: Split dataset A into training (70%), testing (15%), and validation (15%) sets.
Step 2: Perform 10-fold cross-validation.

(Continued)

Algorithm 1 (continued)

Step 3: Train the random forest with optimized tuning parameters for the variables.
Step 4: Evaluate model performance using the testing set and mean decrease in impurity.
Step 5: Let g be the feature importance vector of f.
Step 6: Let h be the variable with the lowest importance score in the g vector.
Step 7: Remove h.
Step 8: Decrement e by 1
end while loop.
Step 9: Let i be the variable in a, not in (j,. . ., b*).
Step 10: Let k = (i,..., b*).
Step 11: Return k.

In this method, the initial feature set includes all features present in the dataset. A fixed number of 10 cross-validation folds is used, with mean decrease in impurity serving as a key measure. The dataset is split into 70:15:15 for training, testing, and validation sets, respectively. Subsequently, a random forest model is trained on the current set of features using cross-validation. Importance scores for each feature are computed, and those with the lowest scores are eliminated. After each iteration, the model's performance is evaluated via cross-validation. This approach helps prevent overfitting and ensures that the selected features generalize well to unseen data. The process ends when the desired number of features is reached and the model's performance stops improving with the removal of additional features. After RFECV completes, 49 features are selected from the dataset. To maintain consistency among all local models, the chosen features are assigned to each edge server so they share similar features. For local model training on each edge server, the BiLSTM is used for sequential data analysis. Reducing to 49 features streamlined model training and inference without significant accuracy loss. Static feature selection may limit the detection of novel or zero-day DDoS attacks, which could show new feature interactions. Future work will develop adaptive feature selection that dynamically updates to detect concept drift and threats, utilising online learning or periodic retraining with updated metrics.

Key formulas and definitions related to the BiLSTM model are discussed below.

Forget Gate: The forget gate, as shown in (2), tells which information from the previous cell state should be discarded. It outputs a value between 0 and 1.

$$f_t = \sigma \tag{2}$$

Input Gate: The input gate in (3) decides which new information from the input x_t should be added to the cell state. It outputs a value between 0 and 1.

$$i_t = \sigma \tag{3}$$

Candidate Cell: The candidate cell, as shown in (4), the state represents new candidate values that could be added to the cell state. The tanh activation function ensures values are between −1 and 1.

$$\widetilde{C}_t = \tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right) \tag{4}$$

Output Gate: The output gate, as shown in (5), decides what next hidden state h_t should be. It regulates the information flow from the cell state to the hidden state.

$$o_t = \sigma\left(w_o \cdot \left[h_{t-1}, x_t\right] + b_0\right) \tag{5}$$

where, w_f, w_i, w_c, w_o represents the weight matrix for the forget gate, input gate, candidate cell state, and output gate, respectively. Besides, b_f, b_i, b_c, b_o represent the bias terms for the forget gate, input gate, candidate cell state, and output gate, respectively. σ represnts the sigmoid activation function. $[h_{t-1}, x_t]$ represents the concatenation of the previous hidden state and the current input at time t.

In the case of DDoS detection, BiLSTM is employed to learn the temporal data patterns and relationships within the network traffic data. In a standard LSTM, information flows in one direction through the sequence. BiLSTM processes the sequences in both forward and backward directions, as shown in Fig. 2.

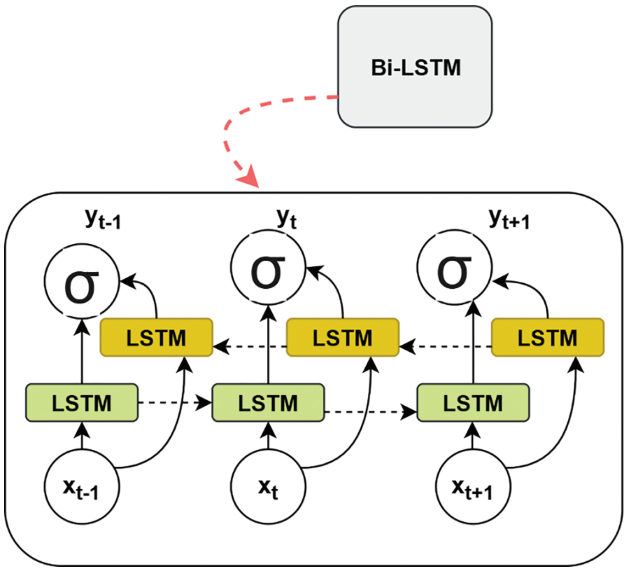


Figure 2: BiLSTM architecture

The forward LSTM processes the input sequence from the beginning to the end, producing a forward hidden state. $\vec{h_t}h_{t-1}$ represents the hidden state of the LSTM cell at the previous step $(t-1)$ and C_{t-1} represents the cell state of the LSTM cell at the previous step (t − 1), as shown in (6).

$$\vec{h_t} = \text{LSTM}\left(x_t, \vec{h_{t-1}}, \vec{C_{t-1}}\right) \tag{6}$$

The backward LSTM process the input sequence from the end to the beginning, producing a backward hidden state $\acute{h_t}$, x_t represents the input at the current step (t) h_{t+1} represents the hidden state of the backward LSTM cell at the next step $(t+1)$ and C_{t+1} represents the cell state of backward LSTM cell at next step $(t+1)$ as shown in (7).

$$\acute{h_t} = \text{LSTM}\left(x_t, \acute{h_{t+1}}, \acute{C_{t+1}}\right) \tag{7}$$

The final BiLSTM output is the concatenation of the forward (h_t) and backward $(\acute{h_t})$ hidden states, as shown in (8).

$$h_t^{\text{BiLSTM}} = \left[\vec{h_t}, \acute{h_t}\right] \tag{8}$$

The local BiLSTM model is trained on the local dataset of each edge server. During training, the model learns to identify patterns and relationships that indicate normal and potentially malicious network behaviour. After each iteration, if the local model on an edge server shows improvement, its weights are

shared with the global server. The global server aggregates these updates from all edge servers to update the global model, as illustrated in Fig. 3. In federated learning, the global model update refers to the process of aggregating and combining the knowledge learned by individual local models on edge servers into a unified global model. The proposed research utilises federated averaging to generate the global model through aggregate updates. The updated global model is distributed back to the edge servers.

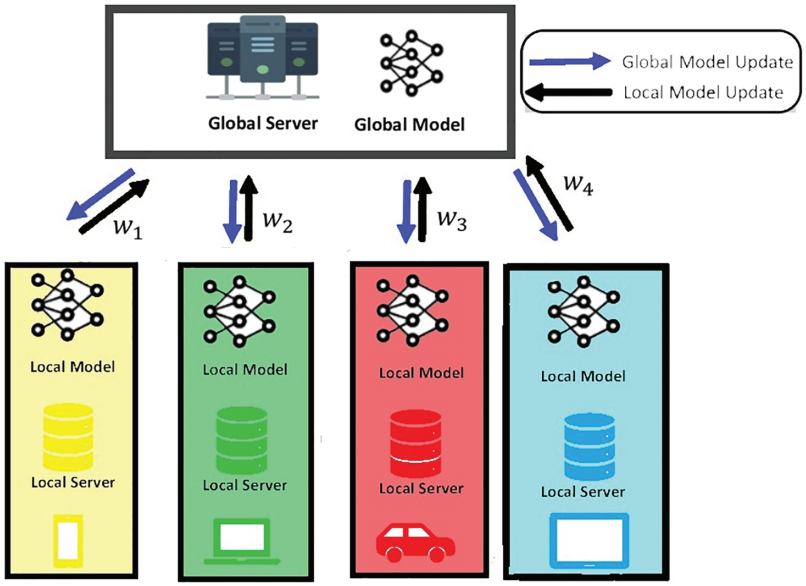


Figure 3: Global model update

The proposed research has also used blockchain to securely store and validate global model updates. Instead of directly sending updates to the central server, edge devices write them to a blockchain network. Each update gets stored as a block, cryptographically linked to the previous one, creating an immutable chain. The smart contracts ensure data integrity and prevent tampering by leveraging their decentralized and immutable nature. Smart contracts on the blockchain are designed to manage and validate global model updates. These contracts define various rules and conditions for the acceptance or rejection of updates. They ensure that only valid and authorized updates are integrated into the global model. Next, the central server analyzes attack data, including attack patterns and logs. It periodically updates the global model with this information to enhance its detection capabilities. The central server then analyses attack reports and network data to identify the type, severity, and source of the attack. Federated learning is well-suited for detecting DDoS attacks, but cannot be directly applied to mitigation strategies. Network administrators are notified to take appropriate action for mitigation, such as traffic rerouting, IP blocking, and rate limiting. In traffic rerouting, actual users are directed to alternate routes (e.g., secondary servers) to ensure critical services are accessible and the flooded system is bypassed. In IP blocking, the system attempts to prevent the attacker from sending further malicious traffic by identifying their IP addresses and blocking their access to the services. In rate limiting, overwhelming the system with requests is controlled by limiting the maximum number of requests a user can send.

The pseudo-code provided in Algorithm 2 outlines the complete process for federated learning-based DDoS detection and mitigation. It begins with data collection and pre-processing, followed by recursive feature elimination for feature selection. Local BiLSTM models are trained further on the selected features, and their updates are shared with the global server. The centralized server deploys this model for real-time

traffic evaluation. A blockchain-based smart contract ensures model integrity and supports decentralized decision-making. The system finally analyses attack data, updates the global model, and provides mitigation techniques. To evaluate the proposed model, FRF-BiLSTM, using a federated learning-based DDoS detection system, extensive experiments were conducted with different datasets and attack scenarios. The performance of the model is evaluated using standard metrics, namely F1-score, Precision, Accuracy, and Recall, as shown in (9)–(12), respectively.

$$\text{F1} - \text{score} = 2\frac{\text{PR*R}}{\text{PR} + \text{R}} \tag{9}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TN} + \text{FP} + \text{TP} + \text{FN}} \tag{11}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

where True Positive (TP) is the model correctly predicts an attack, False Positive (FP) is the model incorrectly predicts an attack when it is normal behavior, True Negative (TN) is the model correctly predicts normal behavior, and False Negative (FN) is the model incorrectly predicts normal behavior when it is actually an attack. Integrating blockchain into federated learning introduces significant computational overhead, necessitating additional processing for consensus, which in turn increases latency, storage requirements, and energy consumption. While blockchain enhances security and transparency, it also raises complexity and resource demands, necessitating a balance between performance and security design.

Algorithm 2: Federated learning-based DDoS detection and mitigation pseudo code

Step 1: Collect local datasets from distributed sources (e.g., edge devices, network nodes).
Step 2: Preprocess and clean the local datasets (e.g., data cleaning and data normalization).
Step 3: Gather features from all local nodes.
Step 4: Use random forest to perform recursive feature elimination for feature selection.
Initialize $n\_estimators$ = 100, max $\_depth$ = 10, $min\_samples\_leaf$ = 5.
Step 5: Update the global feature set based on the selected features.
for each local node in the distributed sources,
do
Step 6: Initialize local BiLSTM model parameters.
n_layers = 3, Hidden units per layer = 64, Optimizer = Adam, learning_rate = 0.005
while the convergence criteria are not met
do
Step 7: Train the local BiLSTM model on the local dataset using the selected features.
Step 8: Evaluate local BiLSTM model performance.
if improvement_criteria_met then
Step 9: Share local BiLSTM model updates with the global server.
end
end
end
for each global training epoch do
Step 10: Aggregate local BiLSTM model updates on the global server.

(Continued)

Algorithm 2 (continued)

Step 11: Update the global BiLSTM model with the aggregated updates.
end
Step 12: Deploy the global BiLSTM model to a centralized server.
For each incoming traffic at the centralized server do
Step 13: Evaluate traffic using the global BiLSTM model.
If deviations from normal behavior are detected, then
Step 14: Trigger alerts and flag potential DDoS attacks.
end
end
Step 15: Smart Contract:
Step 16: Stores and manages global BiLSTM model updates.
Step 17: Verifies the integrity and provenance of the BiLSTM model.
Step 18: Enables decentralized decision-making for attack detection.
Step 19: Gather attack data and feedback from local nodes, and analyse the attack patterns and performance of the BiLSTM model.
Step 20: Identify areas for model improvement and update global BiLSTM model based on feedback.
For each identified DDoS attack,
do
Step 21: Initiate mitigation measures (e.g., traffic filtering, IP blocking).
end
Step 22: Return the final decision for each local node whether an attack is a DDoS attack or not.
Step 23: Return mitigation results.

4 Results and Analysis

For evaluating the effectiveness of the model, the training delay and testing delay are computed as metrics for comparison with other models, as shown in Table 4. Various other relevant models were compared against the FRF-BiLSTM model, and these models were evaluated using four network attack datasets: CICDDoS2019 [32], CICDoS2017 [33], NSL-KDD [34], and CICIDS [35]. The proposed model yielded optimal results in terms of the delay components. It takes the least time to train and test the unseen data compared to other models, with a mean training delay of 88.7 s and a mean testing delay of 21.4 s.

Table 4: Comparison of training and testing times of the proposed model with other models across datasets

Dataset	Model	Training time (s)	Testing time (s)
CICDDoS2019	CNN-LSTM	107.6	30.2
	CNN-BILSTM	128.4	52.3
	FRF-BiLSTM	85.3	18.9
	DNN	143.5	49.13
CICDOS2017	CNN-LSTM	103.6	32.4
	CNN-BILSTM	156.2	62.7
	FRF-BiLSTM	91.5	23.9
	DNN approach	192.5	69.3

(Continued)

Table 4 (continued)

Dataset	Model	Training time (s)	Testing time (s)
NSL-KDD	CNN-LSTM	133.5	40.4
	CNN-BILSTM	123.7	48.6
	FRF-BiLSTM	82.3	20.2
	DNN approach	169.9	58.7
CICIDS	CNN-LSTM	134.7	36.3
	CNN-BILSTM	116.2	37.6
	FRF-BiLSTM	95.7	22.6
	DNN approach	166.7	54.7

To further validate its robustness and generalisation, additional evaluations were conducted using all four datasets, including the proposed model and other existing models. The proposed model demonstrated consistent and high-performance results, showcasing its effectiveness across various datasets and reliability in real-world scenarios. Different parameters, such as accuracy, precision, recall, and F1-score metrics, were taken into consideration. Fig. 4 shows the accuracy rate analysis of the evaluation. The FRF-BiLSTM model achieved a maximum mean accuracy of 97.1%, outperforming other models. In comparison, the DNN approach recorded a slight dip in accuracy. Precision is an important evaluation metric for any predictive model that determines the segment of true positive samples in the DDOS attack dataset among all data samples categorized as positive. An optimal mean precision value of 96.9% is achieved using the FRF-BiLSTM model with all four datasets, indicating that very few false positive samples are present. Fig. 5 shows the overall outcome.

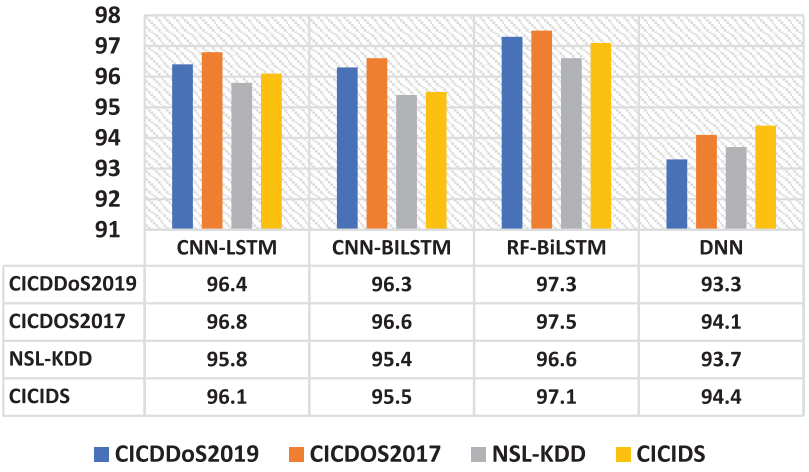
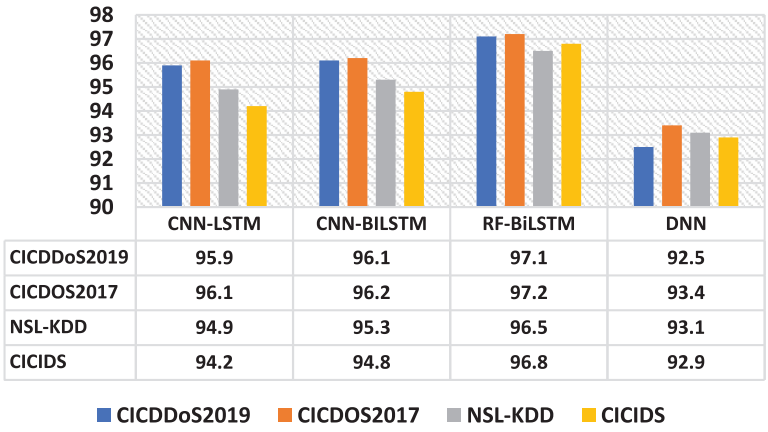


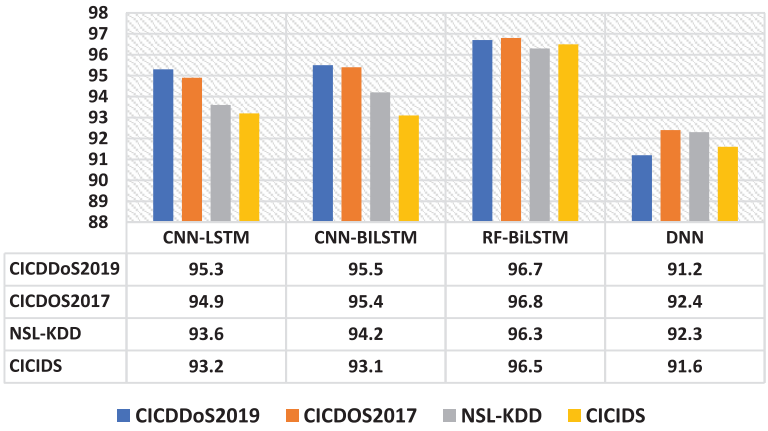
Figure 4: Accuracy analysis of the FRF-BiLSTM model and other models with various datasets

Along with precision, recall is another vital measure of validation, and it is observed that the proposed model computed an optimum recall value of 96.5% when evaluated with four considered datasets. The recall metric computes the portion of true positive samples among all data samples in the DDOS attack dataset. The proposed model achieved a very high recall value, indicating the presence of very few false negatives. The summarized result is depicted in Fig. 6.



	CNN-LSTM	CNN-BILSTM	RF-BiLSTM	DNN
CICDDoS2019	95.9	96.1	97.1	92.5
CICDOS2017	96.1	96.2	97.2	93.4
NSL-KDD	94.9	95.3	96.5	93.1
CICIDS	94.2	94.8	96.8	92.9

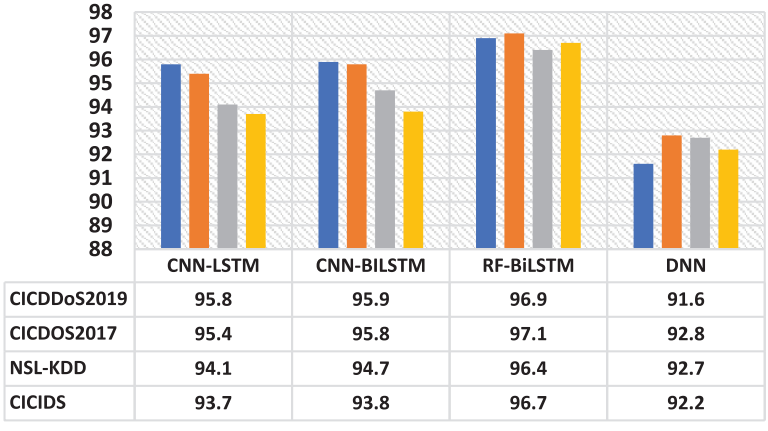
Figure 5: Precision analysis of the FRF-BiLSTM model and other models with various datasets



	CNN-LSTM	CNN-BILSTM	RF-BiLSTM	DNN
CICDDoS2019	95.3	95.5	96.7	91.2
CICDOS2017	94.9	95.4	96.8	92.4
NSL-KDD	93.6	94.2	96.3	92.3
CICIDS	93.2	93.1	96.5	91.6

Figure 6: Recall analysis of the FRF-BiLSTM in comparison with other models across various datasets

The F1-score represents the harmonic mean value of precision and recall, reflecting a balanced parameter that denotes the model's accuracy. Fig. 7 illustrates the F1-score metric analysis of the proposed model in comparison to others. Here, the mean F1-score value is compared to other models, achieving a value of 96.7%.



	CNN-LSTM	CNN-BILSTM	RF-BiLSTM	DNN
CICDDoS2019	95.8	95.9	96.9	91.6
CICDOS2017	95.4	95.8	97.1	92.8
NSL-KDD	94.1	94.7	96.4	92.7
CICIDS	93.7	93.8	96.7	92.2

Figure 7: F1-score comparison of the FRF-BiLSTM model and other models across various datasets

Table 5 describes the importance of evaluating the performance and robustness of the proposed BiLSTM model. The F1-score serves as a balanced measure of the proposed model's effectiveness in detecting DDoS attacks. The consistently high F1-scores (around 0.98) across various attack numbers indicate that the proposed model generally performs well in correctly classifying both normal and attack instances. However, low F1 standard deviations (around 0.01 in the table) across different attack numbers suggest that the model's F1-score is not significantly affected by the number of attacks present. This indicates robustness and reliability in its performance. Observing the high F1-scores and low standard deviations across different attack numbers in the table suggests that the proposed BiLSTM model is likely effective and consistent in detecting DDoS attacks within the CICDDoS2019 dataset, regardless of the number of attacks present.

Table 5: Standard deviation and mean of F1-score corresponding to increases in DDoS attacks

Number of attacks	F1-Score (Approx.)	F1 StdDev (Approx.)
1	0.95	0.05
2	0.96	0.04
3	0.96	0.04
4	0.96	0.04
5	0.96	0.04
6	0.97	0.03
7	0.97	0.03
8	0.97	0.03
9	0.97	0.03
10	0.98	0.02
11	0.98	0.02
12	0.98	0.02
13	0.98	0.02
14	0.98	0.02
15	0.98	0.02
16	0.98	0.02
17	0.98	0.02
18	0.99	0.01
19	0.99	0.01
20	0.99	0.01

The ROC curve and Precision-Recall curve of the proposed model are shown in Fig. 8. The graph indicates that the FRF-BiLSTM model yields the highest ROC curve among the other models.

The metrics shown in Table 6 indicate that the overall performance of the blockchain system is notably good and effective. The Transactions Per Second (TPS) value is 5.568, demonstrating a reasonable throughput and indicating the capability to handle a moderate volume of transactions efficiently. These updates occur asynchronously and at relatively infrequent intervals (e.g., after each training round or epoch), rather than at the same rate as real-time traffic flows. As such, the current TPS rate is sufficient for maintaining model integrity and auditability without interfering with the real-time detection pipeline, which is handled by the deployed BiLSTM model on the centralized server.

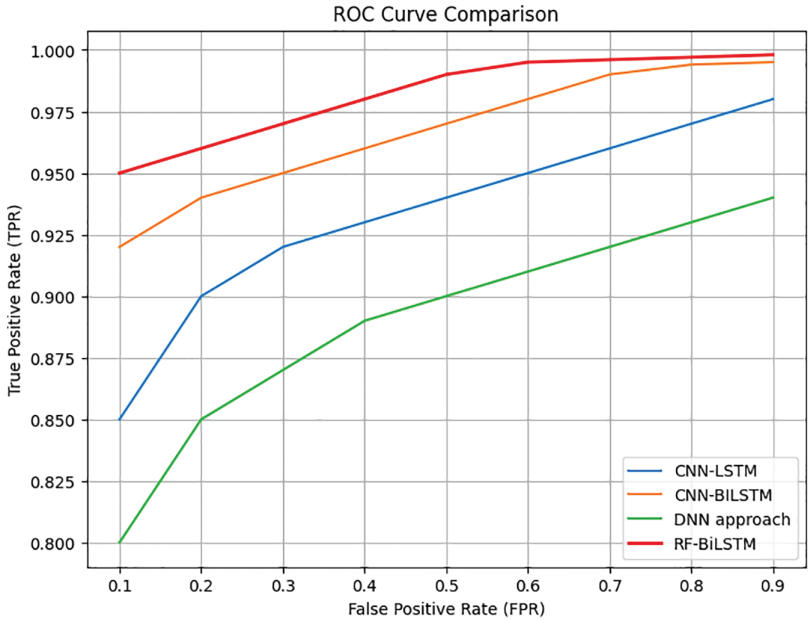


Figure 8: ROC curve on CICDDoS2019 dataset

Table 6: Performance metrics evaluating the efficiency of the blockchain network

Metrics	Value
TPS	5.568
TPC	0.002
TPMS	0.106
TPDIO	0.273
TPND	0.236

Furthermore, the low Transactions Per CPU (TPC) value of 0.002 indicates the system's ability to confirm transactions efficiently. Additionally, metrics such as Transactions Per Memory Second (TPMS), Transactions Per Disk I/O (TPDIO), and Transactions Per Network Data (TPND) demonstrate good efficiency in resource utilization, indicating an optimally configured system that effectively utilizes mining, disk, and network resources to process transactions quickly and reliably. These metrics suggest that the blockchain system is well-equipped to handle updation tasks effectively.

To evaluate the impact of blockchain on system performance, we analysed resource use, CPU, memory, and network utilisation during federated learning. The blockchain-secured model updates and manages consensus. Peak operations saw a CPU increase of 18%–22% and a memory usage of 10%–15%, mainly due to cryptography and validation. Despite this, performance remained stable and within the capacity of edge nodes. Table 7 presents a comparative analysis of the federated learning model and the centralized approach in detecting DDOS attacks. The evaluation outcomes generated by applying federated learning to train the model are comparable to those recorded by using the centralized training approach. Thus, federated learning can guarantee high model accuracy while protecting data privacy.

Table 7: Comparison between the training of the federated model and the centralized model in terms of metrics

Dataset	Evaluation metrics	Federated model (%)	Centralised model (%)
CICDDoS2019	Accuracy	97.3	97.5
	Precision	97.1	97.2
	Recall	96.7	96.9
	F1-score	96.9	96.9
CICDOS2017	Accuracy	97.5	97.6
	Precision	97.2	97.4
	Recall	96.8	96.8
	F1-score	97.1	97.4
NSL-KDD	Accuracy	96.6	96.8
	Precision	96.5	96.7
	Recall	96.3	96.5
	F1-score	96.4	96.4
CICIDS	Accuracy	97.1	97.3
	Precision	96.8	96.9
	Recall	96.5	96.5
	F1-score	96.7	96.8

Statistical Validation

To rigorously assess the robustness and significance of the FRF-BiLSTM model's performance, 10 independent experimental runs for each dataset were conducted. For each key metric (accuracy, precision, recall, and F1-score), the mean (μ), standard deviation (σ), and the corresponding 95% confidence intervals (CI) were computed using the formula in Eq. (13)

$$\text{CI} = \mu \pm t_{\alpha/2,n-1} \times \frac{\sigma}{\sqrt{n}} \tag{13}$$

where $t_{\alpha/2,n-1}$ is the critical value from the t-distribution with $n-1$ degrees of freedom at a significance level $\alpha = 0.05$. On the CICDDoS2019 dataset, the FRF-BiLSTM model achieved an average F1-score of 0.969 with a standard deviation of 0.012. The 95% confidence interval was calculated as [0.960, 0.978], indicating a tight range and high reliability of the results. We conducted paired t-tests on F1-scores to verify if performance improvements over CNN-LSTM and DNN were significant. p-values below 0.01 rejected the null hypothesis, confirming FRF-BiLSTM's superiority. These analyses show the model's consistent, robust, and considerable performance across datasets and metrics.

Despite promising results, this study has limitations. It primarily relies on benchmark datasets such as CICDDoS2019 and NSL-KDD, which may not accurately reflect real-world traffic. Experiments lack adversarial testing, crucial for robustness. While federated learning and blockchain enhance privacy and security, the framework's scalability in large, high-throughput networks remains untested, especially due to blockchain's computational overhead. Future work will include incorporating real traffic data, conducting adversarial assessments, and optimising for large-scale deployment.

The BiLSTM model showed several advantages that contributed to its high accuracy. Its ability to capture long-term dependencies and patterns within the network traffic allowed it to detect DDoS attacks. Recent studies have demonstrated that BiLSTM architectures remain highly effective for time-series and sequential

data tasks, particularly in cybersecurity and network anomaly detection. The CNN-BiLSTM-Attention model enhances intrusion detection accuracy and reduces false positives by combining spatial-temporal features, attention, and Equalisation Loss to address class imbalance, as noted by Dai et al. [36]. Unlike unidirectional LSTMs, BiLSTMs can leverage information from both directions in the sequence, which is critical for detecting complex temporal dependencies in DDoS attack patterns. While Transformer models have gained popularity due to their attention mechanisms, they often require larger datasets and computational resources to generalize effectively (Latibari et al. [37]). In contrast, BiLSTM provides a more efficient and interpretable approach for our moderately sized datasets and temporal features. Our experiments align with these findings, showing that BiLSTM achieves superior detection accuracy compared to unidirectional LSTM and Transformers on the CICDDoS2019 dataset. Additionally, the learning abilities of the BiLSTM model allowed it to update its parameters continuously based on the incoming data. The combination of random forest-based recursive feature elimination for feature selection enhanced the model's efficiency, enabling it to focus on the essential features for DDoS detection.

5 Conclusion and Future Scope

DDoS attacks pose a significant threat to network security and privacy, resulting in immense losses. The use of federated learning ensures decentralized model training, thus protecting the privacy of sensitive data. The proposed FRF-BiLSTM model demonstrated both efficiency and accuracy in detecting DDoS attacks. The random forest-based recursive feature elimination selected the most relevant features from the dataset, reducing computational cost. This optimised feature set enabled the BiLSTM model to focus on essential features, resulting in superior performance and surpassing many existing methods in both accuracy and efficiency. The combination of RFE and BiLSTM yielded a strong balance between accurately detecting real-time attacks and minimising false positives, demonstrating the model's effectiveness for practical deployment. Additionally, the model requires minimal training time, which is significantly faster than that of other existing models, making it suitable for real-world applications. These excellent results enable the FRF-BiLSTM model to serve as a powerful and effective tool for protecting networks against DDoS attacks. The model may face high deployment costs and potential dataset bias, requiring periodic retraining to handle evolving DDoS attack patterns.

In the near future, it is planned to expand the proposed model by integrating even more diverse attack types and network configurations, including integrating real-time threat detection, cross-dataset validation, and optimising blockchain to reduce latency, thereby enabling faster responses.

Acknowledgement: The authors would like to express their sincere gratitude to all individuals and institutions that contributed to the successful completion of this research.

Funding Statement: This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2025S1A5A2A01005171), and by the BK21 program at Chungbuk National University (2025).

Author Contributions: All authors contributed significantly to the conception, design, execution, and interpretation of the study. Sushruta Mishra: conceptualization, methodology, writing—original draft. Sunil Kumar Mohapatra: data curation, validation, formal Analysis. Kshira Sagar Sahoo: supervision, investigation, methodology, valiadation, formal review, editing. Anand Nayyar: critical review, editing. Tae-Kyung Kim: supervision, investigation, and final review. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets generated or analyzed during the current study are available from the corresponding author upon reasonable request.

Ethics Approval: No animals were involved in this study. All applicable international, national, or institutional guidelines for the care and use of animals were followed.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Priya SS, Sivaram M, Yuvaraj D, Jayanthiladevi A. Machine learning based DDoS detection. In: Proceedings of the 2020 International Conference on Emerging Smart Computing and Informatics (ESCI); 2020 Mar 12–14; Pune, India. p. 234–7. doi:10.1109/esci48226.2020.9167642.
2. Hou J, Fu P, Cao Z, Xu A. Machine learning based DDos detection through NetFlow analysis. In: Proceedings of the MILCOM 2018—2018 IEEE Military Communications Conference (MILCOM); 2018 Oct 29–31; Los Angeles, CA, USA. p. 1–6. doi:10.1109/MILCOM.2018.8599738.
3. Ujjan RMA, Pervez Z, Dahal K, Bashir AK, Mumtaz R, González J. Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN. Future Gener Comput Syst. 2020;111:763–79. doi:10.1016/j.future.2019.10.015.
4. Huang W, Peng X, Shi Z, Ma Y. Adversarial attack against LSTM-based DDoS intrusion detection system. In: Proceedings of the 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI); 2020 Nov 9–11; Baltimore, MD, USA. p. 686–93. doi:10.1109/ictai50040.2020.00110.
5. Aydın H, Orman Z, Aydın MA. A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment. Comput Secur. 2022;118:102725. doi:10.1016/j.cose.2022.102725.
6. Li J, Zhang Z, Li Y, Guo X, Li H. FIDS: detecting DDoS through federated learning based method. In: Proceedings of the 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom); 2021 Oct 20–22; Shenyang, China. p. 856–62. doi:10.1109/TrustCom53373.2021.00121.
7. Doriguzzi-Corin R, Siracusa D. FLAD: adaptive federated learning for DDoS attack detection. Comput Secur. 2024;137:103597. doi:10.1016/j.cose.2023.103597.
8. Li J, Lyu L, Liu X, Zhang X, Lyu X. FLEAM: a federated learning empowered architecture to mitigate DDoS in industrial IoT. IEEE Trans Ind Inform. 2022;18(6):4059–68. doi:10.1109/TII.2021.3088938.
9. Hayat RF, Aurangzeb S, Aleem M, Srivastava G, Lin JC. ML-DDoS: a blockchain-based multilevel DDoS mitigation mechanism for IoT environments. IEEE Trans Eng Manage. 2024;71:12605–18. doi:10.1109/TEM.2022.3170519.
10. Manikumar DVVS, Uma Maheswari B. Blockchain based DDoS mitigation using machine learning techniques. In: Proceedings of the 2020 2nd International Conference on Inventive Research in Computing Applications (ICIRCA); 2020 Jul 15–17; Coimbatore, India. p. 794–800. doi:10.1109/icirca48905.2020.9183092.
11. Rahim JA, Nordin R, Amodu OA. Open-source software defined networking controllers: state-of-the-art, challenges and solutions for future network providers. Comput Mater Continua. 2024;80(1):747–800. doi:10.32604/cmc.2024.047009.
12. Kumari P, Jain AK. A comprehensive study of DDoS attacks over IoT network and their countermeasures. Comput Secur. 2023;127:103096. doi:10.1016/j.cose.2023.103096.
13. Ali TE, Chong YW, Manickam S. Machine learning techniques to detect a DDoS attack in SDN: a systematic review. Appl Sci. 2023;13(5):3183. doi:10.3390/app13053183.
14. Chaganti R, Bhushan B, Ravi V. A survey on blockchain solutions in DDoS attacks mitigation: techniques, open challenges and future directions. Comput Commun. 2023;197:96–112. doi:10.1016/j.comcom.2022.10.026.
15. Mustapha A, Khatoun R, Zeadally S, Chbib F, Fadlallah A, Fahs W, et al. Detecting DDoS attacks using adversarial neural network. Comput Secur. 2023;127:103117. doi:10.1016/j.cose.2023.103117.
16. Rizvi F, Sharma R, Sharma N, Rakhra M, Aledaily AN, Viriyasitavat W, et al. An evolutionary KNN model for DDoS assault detection using genetic algorithm based optimization. Multimed Tools Appl. 2024;83(35):83005–28. doi:10.1007/s11042-024-18744-5.

17. H SC, Rao KV, Prasad MHMK. Deep neural network empowered bi-directional cross GAN in context of classifying DDoS over flash crowd event on web server. Multimed Tools Appl. 2023;82(24):37303–26. doi:10.1007/s11042-023-15030-8.
18. Jawahar A, Kaythry P, Vinoth Kumar C, Vinu R, Amrish R, Bavapriyan K, et al. DDoS mitigation using blockchain and machine learning techniques. Multimed Tools Appl. 2024;83(21):60265–78. doi:10.1007/s11042-023-18028-4.
19. Dogra A, Taqdir. DDOS attack prevention and validation with metric based ensemble approach. Multimed Tools Appl. 2023;82(28):44147–54. doi:10.1007/s11042-023-15523-6.
20. Cil AE, Yildiz K, Buldu A. Detection of DDoS attacks with feed forward based deep neural network model. Expert Syst Appl. 2021;169:114520. doi:10.1016/j.eswa.2020.114520.
21. Ahmed I, Ahmad M, Jeon G. Federated learning in convergence ICT: a systematic review on recent advancements, challenges, and future directions. Comput Mater Contin. 2025;85(3):4237–73. doi:10.32604/cmc.2025.068319.
22. Anjum M, Dutta AK, Elrashidi A, Shahab S, Aldrees A, Shaikh ZA, et al. GraphFedAI framework for DDoS attack detection in IoT systems using federated learning and graph based artificial intelligence. Sci Rep. 2025;15(1):28050. doi:10.1038/s41598-025-10826-0.
23. Munaweera P, Prasad S, Hewa T, Siriwardhana Y, Ylianttila M. Federated learning-powered DDoS attack detection for securing cyber physical systems in 5G and beyond networks. In: Proceedings of the 14th International Conference on the Internet of Things; 2024 Nov 19–22; Oulu, Finland. p. 273–8. doi:10.1145/3703790.3703822.
24. Fotse YSN, Tchendji VK, Velempini M. Federated learning based DDoS attacks detection in large scale software-defined network. IEEE Trans Comput. 2025;74(1):101–15. doi:10.1109/TC.2024.3474180.
25. Lee YC, Chien WC, Chang YC. FedDB: a federated learning approach using DBSCAN for DDoS attack detection. Appl Sci. 2024;14(22):10236. doi:10.3390/app142210236.
26. Almadhor A, Altalbe A, Bouazzi I, Al Hejaili A, Kryvinska N. Strengthening network DDOS attack detection in heterogeneous IoT environment with federated XAI learning approach. Sci Rep. 2024;14(1):24322. doi:10.1038/s41598-024-76016-6.
27. Tymoshchuk D, Yasniy O, Mytnyk M, Zagorodna N, Tymoshchuk V. Detection and classification of DDoS flooding attacks by machine learning method. arXiv:2412.18990. 2024.
28. Mandela N, Etyang F. Comparative analysis of deep learning models for effective denial of service (DoS) attack detection in network security. J Electr Syst Inf Technol. 2025;12(1):73. doi:10.1186/s43067-025-00267-0.
29. Alfatemi A, Rahouti M, Amin R, ALJamal S, Xiong K, Xin Y. Advancing DDoS attack detection: a synergistic approach using deep residual neural networks and synthetic oversampling. arXiv:2401.03116. 2024.
30. Dilworth R, Gudla C. Harnessing PU learning for enhanced cloud-based DDoS detection: a comparative analysis. arXiv:2410.18380. 2024.
31. Wang H, Wang Q, Ding Y, Tang S, Wang Y. Privacy-preserving federated learning based on partial low-quality data. J Cloud Comput. 2024;13(1):62. doi:10.1186/s13677-024-00618-8.
32. DDoS evaluation dataset (CIC-DDoS 2019) [Internet]. Fredericton, NB, Canada: Canadian Institute for Cybersecurity, University of New Brunswick. 2019 [cited 2025 Apr 27]. Available from: https://www.unb.ca/cic/datasets/ddos-2019.html.
33. CICDoS 2017 [Internet]. Fredericton, NB, Canada: Canadian Institute for Cybersecurity, University of New Brunswick; 2017 [cited 2025 Apr 21]. Available from: https://www.kaggle.com/datasets/dhoogla/cicdos2017.
34. KDD cup 1999 [Internet]. Irvine, CA, USA: University of California, School of Information and Computer Science; 1999 [cited 2025 Apr 21]. Available from: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
35. Intrusion detection evaluation dataset (CIC-IDS2017) [Internet]. Fredericton, NB, Canada: Canadian Institute for Cybersecurity, University of New Brunswick; 2017 [cited 2025 Apr 30]. Available from: https://www.unb.ca/cic/datasets/ids-2017.html.
36. Dai W, Li X, Ji W, He S. Network intrusion detection method based on CNN-BiLSTM-attention model. IEEE Access. 2024;12:53099–111. doi:10.1109/ACCESS.2024.3384528.
37. Latibari BS, Nazari N, Alam Chowdhury M, Immanuel Gubbi K, Fang C, Ghimire S, et al. Transformers: a security perspective. IEEE Access. 2024;12:181071–105. doi:10.1109/access.2024.3509372.