



Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

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

Full Length Article

Binary classification of Low-Rate DoS attacks using Long Short-Term Memory Feed-Forward (LSTM-FF) Intrusion Detection System (IDS)

Suhaila ZeinElabideen Omer ^{a,b,*}, Fazirulhisyam Hashim ^{a,b,*} , Aduwati Sali ^{a,b,c},
Faisul Arif Ahmad ^a

^a Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400, Malaysia

^b Research Centre of Excellence for Wireless and Photonic Network (WiPNET), Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400, Malaysia

^c Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

ARTICLE INFO

Keywords:

Deep Learning (DL)
Low-Rate DoS (LR-DoS)
Intrusion Detection System (IDS)
Recurrent Neural Network (RNN)
Long-Short Term Memory (LSTM)

ABSTRACT

The data and size of networks have grown substantially due to the rapid development of the Internet and other communication techniques. This has led to the development of numerous new types of attacks, making it harder for network security to detect intrusions accurately. The goal of a Denial of Service (DoS) attack is to overwhelm a target with malicious traffic, exhausting its processing power and network bandwidth. Traditional DoS attacks rely on brute force techniques, making them easier to detect, whereas low-rate and slow attacks pose a greater threat due to their stealthy nature. These attacks target application or server resources with a prolonged trickle of traffic, requiring minimal bandwidth yet making mitigation challenging. Their low resource footprint allows them to degrade or deny service to legitimate users while remaining undetected for extended periods. This research introduces an advanced Intrusion Detection System (IDS) that utilizes a hybrid Long Short-Term Memory Feedforward (LSTM-FF) Neural Network to tackle existing challenges in detecting low-rate DoS (LR-DoS) attacks. Unlike previous models, our approach combines temporal sequence learning with feature refinement, thereby improving the detection of LR-DoS. Additionally, we incorporate automated feature selection using Random Forest, which optimizes efficiency while maintaining interpretability. For model training and evaluation, we use the CIC-DOS2017 dataset, which includes eight distinct types of LR-DoS attacks. To enhance generalizability, we also utilize the CSE-CIC-IDS2018 dataset and the newly introduced LR-HR-DDOS2024 dataset, specifically designed for Software-Defined Networking (SDN)-based environments. To address the class imbalance, we implement a stratified k-fold cross-validation strategy, ensuring robust performance across various attack scenarios. To thoroughly evaluate model performance, we adopt a comprehensive set of metrics, including accuracy, precision, recall, F1-score, specificity, False Alarm Rate (FAR), and ROC-AUC. This ensures a well-rounded validation of our approach. The model surpassed all previous state-of-the-art models with an impressive accuracy of 99.70%, precision of 99.47%, specificity of 99.97%, and an F1-score of 97.52%, all while retaining a low FAR of roughly 0.03%. The LSTM-FF approach also worked well in multi-class classification, with a 99.54% accuracy rate, 93.19% precision, 99.59% specificity, 90.28% F1 score, and 0.40% FAR.

1. Introduction

The rapid expansion of computer networks and the proliferation of various applications that impact all aspects of human life have made them accessible to individuals and groups alike, including for commercial use. The widespread adoption of the Internet of Things (IoT) and cloud-based services has created a heightened demand for robust cybersecurity measures and effective protection tools. This is necessary

to prevent cyberattacks that can result in substantial financial losses and significant damage to extensive networks. One particularly insidious type of cyberattack is the Denial of Service (DoS) attack, which aims to disrupt network services by overwhelming them with excessive traffic. Among these attacks, low-rate DoS (LR-DoS) attacks are especially challenging to detect due to their subtlety and covert nature.

Intrusion Detection Systems (IDSs) are widely used to enhance network security. An IDS is a sophisticated tool or process that analyzes

* Corresponding authors.

E-mail addresses: suhazbelal72@gmail.com, gs57150@student.upm.edu.my (S.Z. Omer), fazirul@upm.edu.my (F. Hashim).

<https://doi.org/10.1016/j.jestch.2025.102049>

Received 31 July 2024; Received in revised form 20 February 2025; Accepted 18 March 2025

Available online 10 April 2025

2215-0986/© 2025 The Author(s). Published by Elsevier B.V. on behalf of Karabuk University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

the behavior of a system or network by monitoring malicious or illegal activities and promptly identifying and responding to these threats. The primary purpose of an IDS is to safeguard a system against misuse, intrusions, and cyberattacks [1]. IDS can be categorized into three distinct groups based on their placement. The first type of Intrusion Detection System (IDS) is the Host-based IDS (HIDS), which is installed on a single machine and identifies attacks by examining the critical files of the operating system. Meanwhile, Network-based IDS (NIDS) are typically installed on network routers or switches, and Hybrid-based IDS (HIDS) combines both HIDS and NIDS to provide comprehensive security [2].

IDS can also be categorized into two types based on their method of detecting intrusions, namely signature-based IDS (SIDS) and anomaly-based IDS (AIDS). SIDS employs matching techniques to identify pre-established and already present intrusion patterns in the system's signature database. This is sometimes known as misuse detection or knowledge-based detection [3]. On the other hand, AIDS is founded on the notion of precisely delineating a standard pattern for regular behavior. Any departure from this typical pattern is seen as abnormal conduct or an aberration; this category is sometimes referred to as behavior-based detection. The critical benefit of IDS, which has garnered the interest of numerous academic researchers, lies in their capacity to adapt the typical activity profile to suit various applications and networks [4]. IDS encounters numerous obstacles, such as enhancing detection precision, minimizing false positives, and identifying novel attacks. Recently, many researchers focused on the development of IDS using Machine Learning (ML) techniques to tackle these problems [5].

ML is the procedure of automating the acquisition of knowledge from instances. It detects anomalies by distinguishing between regular and abnormal categories [6]. Various methods utilizing ML techniques have been suggested in the literature to identify and detect network intrusions. These include Random Forest (RF), k-Nearest Neighbor (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and Logistic Regression (LR) [7,8,9,10,11]. While many ML-based IDS approaches have demonstrated improved detection accuracy, the increasing volume of internet traffic and the rapid emergence of new technologies are generating vast amounts of complex, multidimensional data.

Meanwhile, the complexity of attack scenarios has increased, leading to a greater reliance on deep ML to address the growing security risks effectively [12]. Deep learning (DL) is a highly regarded study field that is now being focused on to overcome these constraints. DL is a sophisticated branch of ML that can surpass some limitations of ML and shallow learning. Initial investigations into DL have demonstrated that super-learning can enhance the analysis of network data, enabling faster identification of anomalies [13]. In principle, the DL approach follows a process similar to ML but automates feature extraction. DL can acquire intricate and conceptual high-level characteristics, a fundamental attribute of this field. This is advantageous as processing such features is time-consuming and demands specialized expertise [14,15].

Authors in [16,17] show that deep neural networks effectively enhance the design of IDS. They are well suited for modeling a high-accuracy classification model and outperform conventional ML techniques for categorizing data into two or more classes. For example, Recurrent Neural Networks (RNNs) are particularly effective for modeling sequential data, making them ideal for analyzing network traffic, which often exhibits temporal dependencies. RNNs can handle sequences of varying lengths, capture nonlinear relationships, and retain the memory of previous inputs. Variants of RNNs, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), have been developed to address the vanishing gradient problem and improve learning performance.

In light of this, we utilize an RNN model that is built on long short-term memory feed-forward (LSTM-FF) as a system for detecting LR-DoS attacks. In principle, we utilize the CIC-DOS2017 dataset, accessible to the public, from the University of New Brunswick (UNB) data repository. We employed CIC-DOS2017 as the primary dataset to train

the model and evaluate its performance. By addressing the limitations of traditional IDS approaches and leveraging advanced DL techniques, this study contributes to the development of a more effective and adaptive IDS, capable of identifying sophisticated LR-DoS attacks. The paper's contributions encompass:

1. Development of an IDS to detect LR-DoS attacks using a DL technique capable of automatically performing feature engineering by identifying hidden patterns in data.
2. The models were developed using deep recurrent neural networks (RNNs), chosen for their ability to handle variable-length sequences, perform non-linear mapping, and retain memory of previous inputs. The proposed models were tested with three types of RNNs, namely simple RNN, LSTM, and GRU. Experimental results indicated that the proposed models outperformed many traditional ML approaches.
3. To improve the ML model's performance in attack detection, we determine and apply the decision threshold, also known as the optimal threshold, during classification. This enhances detection accuracy, as reflected by recall (sensitivity), but may slightly increase false alarms, potentially restricting legitimate user access. Intrusion Prevention Systems (IPS), however, with appropriate verification methods can mitigate this issue.

The subsequent sections of this work are as follows: Section 2 presents an overview of neural network methods. Section 3 presents a literature review on recurrent neural network intrusion detection models. Section 4 provides an overview of the dataset utilized in this study. Section 5 provides a detailed explanation of the system, while Section 6 presents an experiment analysis and a discussion of the outcomes. Section 7 discusses the challenges and limitations associated with the performance of LSTM-based model, while Section 8 discusses our conclusions and potential future work.

2. Overview of neural networks

2.1. Feed-forward neural networks (FFNN)

Neural networks transmit signals in either one direction or a recurrence pattern. The most fundamental type of neural network is the feed-forward architecture. The name "feed-forward" comes from input signals initially fed into the input layer and then forwarded to the next layer without any loops. This network architecture was designed to accurately associate a predetermined input size, such as an image, with a predetermined output size. To transition from one layer to the next, a cluster of units computes a weighted sum of its inputs from the preceding layer and then applies a non-linear function to the resulting value [18]. Fig. 1 illustrates the configuration of the feedforward network.

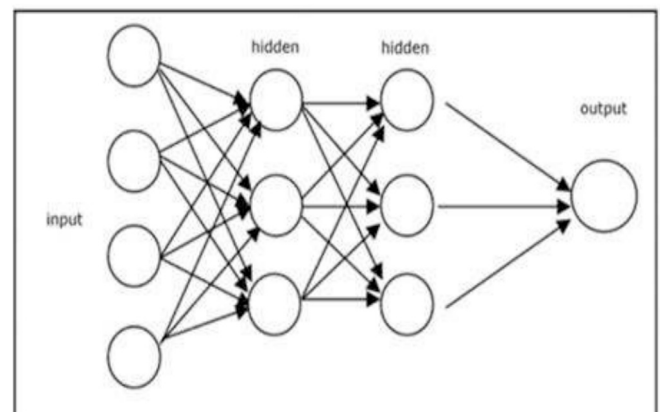


Fig. 1. Feed-forward network [19].

2.2. Recurrent neural networks (RNN)

Recurrent neural networks (RNNs) are a type of neural network that is particularly effective in representing sequences of varying lengths. In addition to the usual output and input layers, it comprises hidden layers of nodes connected recurrently. Neural networks refer to the hidden layers as memory states because they allow the networks to retain sequential knowledge and data from future time steps. A feedback loop structure regulates the historical data, allowing the recurrent neural network to incorporate the output from the prior time step into the current step's input during network training. This iterative paradigm efficiently enables the network to consider past values [20]. So, RNNs, unlike feed-forward networks, propagate data forward and backward from later processing to earlier stages. Fig. 2 shows the difference between the RNN and FFNN.

When trained with stochastic gradient descent, simple RNNs struggle to learn long-term relationships encoded in input sequences due to vanishing gradients. To tackle this problem, researchers introduced the long Short-Term Memory (LSTM) technique as a practical gradient-based approach [22].

2.3. Long short-term memory (LSTM)

RNNs are widely used in research on sequential data, such as audio and text. Nevertheless, RNNs consisting of tanh or sigma cells fail to acquire the pertinent information from the input data when there is a significant disparity between the inputs. Long Short-Term Memory (LSTM) is a type of RNN that gets rid of long-term dependence by building gate functions into the structure of the cells [23]. Three distinct gates, the input gate, the output gate, and the forget gate, update a memory cell in the LSTM unit, which stores data. Fig. 3 depicts the structure of the LSTM unit.

At time t , x_t is the LSTM cell's input data, and h_{t-1} in the LSTM output at the previous moment is the memory cell's value. h_t is the LSTM cell's output, σ (the sigmoid function); we denote the weight parameter as w , the bias parameter as b , and the activation functions as $\tanh$. The following are the individual processes that comprise the LSTM unit computation procedure [24].

$$i_t = \sigma(w_{xi}x_t + w_{hi}h_{t-1} + w_{ci}c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(w_{xf}x_t + w_{hf}h_{t-1} + w_{cf}c_{t-1} + b_f) \quad (2)$$

$$c_t = f_t c_{t-1} + i_t \tanh(w_{xc}x_t + w_{hc}h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(w_{xo}x_t + w_{ho}h_{t-1} + w_{co}c_{t-1} + b_o) \quad (4)$$

$$h_t = o_t \tanh(c_t) \quad (5)$$

3. Related works

LR-DoS attacks are difficult to detect as they mimic legitimate traffic

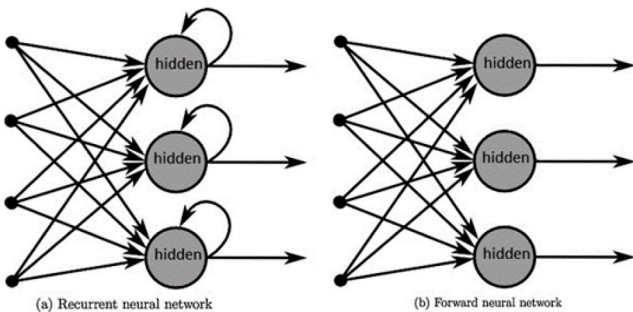


Fig. 2. The difference between RNN and FFNN [21].

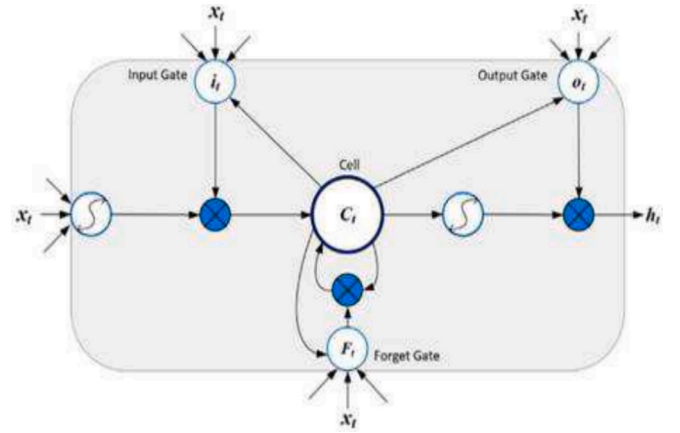


Fig. 3. LSTM unit structure [24].

and blend with normal network activity. Consequently, conventional DoS detection systems struggle to identify them. Researchers have leveraged deep learning techniques to enhance network analysis and improve intrusion detection. DL-based IDS solutions have gained significant traction and demonstrated high effectiveness. This section reviews the most relevant methodologies for our research.

In [25], the authors proposed ID models based on DNN architectures such as recurrent neural networks and convolutional neural networks. They evaluate these deep models, trained on the NSLKDD dataset, using the NSLKDDTest + and NSLKDDTest21 test dataset. To assess their credibility, they compared the models to some of the most well-known machine learning ID algorithms, including K-NN, RF, ELM, SVM, DT, and Naive Bayes. On test datasets, deep learning showed an accuracy of 85 % for the DCNN model and 89 % for the LSTM model, suggesting that it is a practical and encouraging technique for use in information security. The researchers in [26] devised and executed a deep IDS comprising a feedforward model and a DNN model (FFDNN). They also incorporated a Feature Extraction Unit (FEU) that used filter-based methods to provide optimal subsets of data with minimal redundancy. The filtering technique yields a total of 21 features. The selection technique relies on traits that offer significant information gain (IG) regarding the class. The model undergoes testing and evaluation using the NSL-KDD dataset. According to the results, the FFDNN model did better at binary and multiclass classification when using the whole feature space and had less FEU than the DT, RF, SVM, NB, and KNN models.

A novel IDS framework was proposed in [27]. By combining the Sequence Forward Selection (SFS) method to generate feature subsets from the original feature collection and the Decision Tree (DT) model to estimate the cost and accuracy of each combination of features, a hybrid SFS/DT feature selection model was created. Based on the highest accuracy score, the best feature subset was chosen. The models were trained using the NSL-KDD and ISCX datasets. 12 and 3 features were selected for each dataset, representing the best-selected feature subsets. The two subsets were used to train different types of (RNNs), including conventional RNN, LSTM, and GRU. The model was evaluated, and according to the results, the SFS/DT model significantly reduces processing time and memory consumption. Considering the three techniques, the LSTM model achieved the highest accuracy on both IDS datasets.

Network Intrusion Detection Systems (NIDS) can be made more efficient according to a novel design proposed in [28]. This intrusion detection system uses (RNN) with (GRU) to store and automatically extract features. As for non-linear classification choices, it has a multi-layer perceptron probability and generates the output. The KDD-99 and NSL-KDD datasets were used to train the model. The system shows remarkable performance, according to the empirical findings. The GRU model outperforms LSTM, particularly when detecting DOS attacks.

However, it is an improved and simplified version of the LSTM. Success was attained through training that made use of DL techniques.

In [29], an attention-based intrusion detection model was implemented by combining a hierarchical attention mechanism with a Bidirectional Gated Recurrent Unit (BiGRU). The hierarchical attention mechanism is made up of two attention mechanisms. The first is feature-based attention, which decides which feature should be the focus. Second. The authors utilize slice-based attention to group numerous bits of traffic information from nearby seconds, as they believe this approach is crucial in determining the present traffic type. This paper conducts tests using the UNSW-NB15 dataset. Utilizing attention mechanisms, the model achieves an accuracy exceeding 98.76 % and FAR below 1.2 %, outperforming specific cutting-edge methods.

In their study, the authors [30] introduced a highly effective method for detecting new distributed DoS (DDoS) attacks in real-world scenarios. They achieved this by utilizing a specific form of RNN known as a GRU. They developed the DIDDOS model and evaluated its performance using the CICDOS2019 dataset. They assess the suggested approach against various classification algorithms, such as sequential minimal optimization (SMO), NB, and RNN. The study of the experimental results demonstrates that the GRU algorithm has promising accuracy, precision, recall, and F1 score performance.

Singh N. et al. [31] suggested a new GRU model called WideDeep that uses comprehensive transfer learning (TL) to help with the problems of dealing with time series and point data that are multidimensional and have many variables. The system consists of a comprehensive regression component and a deep classification component. The suggested approach combines the ability to memorize the regression model with the ability to generalize the GRU model. The fundamental concept of employing a transfer learning framework involves initially training an initial model and subsequently fine-tuning the entire initial model on the same dataset iteratively until it achieves optimal performance. The KDDCup 99 (10 %) dataset was used with numerous dimensions, while the UNSW-NB15 dataset was used as data from a multivariate time series to evaluate the model's performance. WideDeep achieves peak performance briefly during training and outperforms most classification algorithms, including Naive Bayes (NB), RNN, and GRU.

In [32], the researchers created a model for detecting network intrusions that combine (CNN) with (BiLSTM) models. They evaluate the performance of this model using many datasets, including UNSW-NB15, NSL-KDD, and InSDN (which is dedicated to SDN). Since these datasets were unbalanced, the researchers used random oversampling to balance them. To determine the most optimal collection of features, they utilized the RF algorithm and recursive feature elimination (RFE), ultimately identifying ten features. The results indicate that the suggested model attained a high level of accuracy and necessitated a reduced training time.

Instead of focusing on general types of network attacks, the authors at [33] specifically concentrate on DoS attacks. They created a model based on the CNN model. The system produces two categories of intrusion images: colorful and gray. It generates 18 designs by manipulating different hyperparameters, including picture type, kernel size, and convolutional layers. Each design underwent binary and multilayer categorization. The KDD and CSE-CIC-IDS-2018 datasets were used for training and testing purposes. The performance of the model was compared to that of the RNN model. The experiment results demonstrate that compared to the RNN model, the CNN model can accurately detect specific DoS attacks with comparable attributes.

Most network intrusion datasets combine intrusion samples with many normal samples, resulting in a data imbalance problem that allows major and minor categories to be available in the same dataset. In the minor category, there usually needs to be more samples for model training, which directly impacts accuracy and raises errors in the detection process. To address this issue, Jiang, Kaiyuan, et al. [34] have devised an intrusion detection algorithm that integrates mixed sampling with an extensive hierarchical network. The mixed sampling method

employs the OSS technique to decrease the number of noisy samples in the majority category while simultaneously utilizing the SMOTE technique to augment the number of minority samples. This hybrid approach yields a well-balanced dataset prepared to comprehensively understand the characteristics of minority samples while significantly reducing the duration of training. In the following phase, an extensive hierarchical network model was developed by integrating CNN with BiLSTM to simultaneously analyze and capture traffic data's time and space attributes. The NSL-KDD and UNSW-NB15 datasets were used for model validation. The suggested model surpasses other classifier methods and attains superior accuracy, precision, and recall rate.

Perez-Diaz, Jesus Arturo, et al. [35]. The authors designed and implemented a modular framework to identify and reduce low-rate DDoS threats in software-defined networking (SDN). They explain that the design's modularity allows for seamless substitution of any module without impacting the other modules of the framework. The proposed intrusion detection system module wants to find network flows using machine learning models that have already been trained. These models include RT, RF, J48, REP Tree, SVM, and MLP. The CIC-DOS2017 dataset was used for performance assessment. The evaluation findings indicate that the MLP achieves an accuracy rate of 95 % in detecting attacks. The Intrusion Prevention System (IPS) efficiently counteracts the identified attacks by the Intrusion Detection Module (IDS), proving the architecture's practicality in identifying and countering low-rate DDoS threats. The architecture was developed in a virtual environment using the ONOS controller and Mininet on VirtualBox.

In [36], the researchers created an LR-DoS attack detection system using a mixed DNN with a single-dimensional CNN and GRU. The valid internet traffic in the data center was recorded for a comprehensive and successful evaluation of the suggested method. Simultaneously, various LR-DoS attacks on the website's mirror in the lab were conducted to capture malicious traffic. The LR-DoS detection approach successfully identified attack assaults in varying HTTP traffic, achieving an impressive average detection rate of 98.68 %. This performance surpasses that of other advanced detection methods by a substantial margin.

The authors in [37] have developed a hybrid deep learning system to enhance the security of the SDN networks. They achieved this by using the CNN model and introducing a novel regularization technique called SD-Reg. SD-Reg focuses on the deviation from the weight matrix mean to solve the overfitting issue and enhance CNN's effectiveness in detecting previously unknown intrusion instances. The suggested models use a Convolutional Neural Network (CNN) architecture along with machine learning techniques like SVM, KNN, and RF. The latest issued SDN dataset, InSDN, is used for training and evaluation. The models were evaluated using two sets of features: 48 and 9. The goal of using fewer features to train the models was to create a lightweight NIDS without significantly lowering the efficacy of the detection system. The experimental findings indicate that the combined utilization of CNN and RF algorithms yielded superior performance, resulting in enhanced accuracy for intrusion detection.

In [38], an SDN defense architecture was presented to protect against unauthorized access and distributed denial of service attacks. The authors configure the SDN controller by examining each single IP flow record. In contrast to other techniques that analyze traffic data across different time intervals, this study works directly on individual IP flow. It categorizes IP flows as regular or anomalous, enabling quicker detection and mitigation actions. After being trained using the GRU deep learning algorithm, the model was evaluated on the CIC-DOS2019 and CIC-IDS2018 public datasets. The GRU model is a viable option for the suggested system because it offers excellent detection rates, and more flows evaluated per second compared to other models.

To provide a clearer understanding of the aforementioned existing works, we have systematically reviewed their methodologies and identified their limitations. A detailed breakdown of these challenges is presented in Appendix A. However, for the benefit of readers, we have summarized their key limitations into specific categories, as outlined in

Table 1

Critical comparison of the limitations identified in previous related works.

Key Limitations	Descriptions
Dataset Limitations	– Many studies rely on legacy datasets like NSL-KDD, KDD99, and UNSW-NB15, which do not fully capture modern network complexities or evolving attack patterns [25–29,31–34]. These datasets often lack realistic low-rate DoS scenarios.
Attack Detection Gaps	– The detection of low-frequency or sophisticated attack types, including low-rate DoS attacks, remains a significant challenge due to imbalanced datasets and insufficient feature representation [25–32]. – Models trained on older datasets often fail to generalize about unseen or complex attacks [25–29,31–34].
Model Complexity and Computational Challenges	– Deep learning models such as CNN, GRU, and Hybrid approaches (e.g., CNN-BiLSTM) introduce high computational overhead, requiring substantial training time and resources, which may be impractical for real-time network deployment [25,28,30,32–34,36–38]. – Hierarchical attention mechanisms and transfer learning models further add complexity, impacting scalability and deployment efficiency [29,31].
Feature Selection Deficiencies	– Many studies neglect comprehensive feature selection methods, impacting model accuracy and creating difficulties in identifying intricate traffic patterns [25,30,33,35,37,38]. – The conversion of traffic features into image formats for CNN processing is innovative but may result in the loss of important data [25,32–34].
Evaluation Limitations	– Many works rely heavily on basic evaluation metrics (accuracy, precision, recall) without adequately focusing on metrics that better capture low-frequency attack detection, such as false negative rates [25–27,30–34,38].

Table 1. Furthermore, **Table 1** also contrasts these existing methods with our proposed LSTM-FF approach, highlighting how our solution effectively addresses the identified gaps.

In particular, the proposed model for network intrusion detection addresses several limitations found in previous works. It utilizes diverse datasets, including LR-HR-DDOS2024 and CSE-CIC-IDS2018, to enhance generalizability. Additionally, our proposed model employs stratified k-fold cross-validation to manage data imbalance, which improves the detection of rare attacks.

This hybrid architecture integrates LSTM and FFNN layers to effectively capture temporal dependencies and enhance feature refinement. To reduce the risk of overfitting, dropout regularization is implemented. Additionally, feature selection is automated through the RF technique, which minimizes input dimensions and improves both efficiency and interpretability. A comprehensive set of evaluation metrics including accuracy, precision, recall, F1-score, specificity, False Alarm Rate (FAR), and ROC-AUC provides a robust assessment of the model's performance, thereby enhancing its reliability in detecting LR-DoS attacks.

4. Dataset

Researchers curated the CIC-DOS2017 dataset [39] to analyze and identify DoS attacks targeting OSI layer 7. The dataset includes normal network traffic and eight different low-rate attack variations, such as HULK [40], Goldeneye [41], DDOSim [42], Slowloris [41], RUDY [43], Slowbody2 [44], Slow-headers [45], and Slowread [45]. The CIC-FlowMeter tool was used to transform the data packets from a standard network into network traffic statistics suitable for implementation in an SDN environment.

4.1. Dataset pre-processing

To prevent the model from attributing IP addresses, protocols, and ports as offensive points and generalizing, the dataset was modified by removing the flow ID, IP source address, IP destination address, ports, and protocol features. The dataset is checked for infinite, null, and duplicate values. Subsequently, the flows were encoded with benign and attack flows and labeled 0 and 1, respectively, for binary classification. Benign flows were labeled 0 for multiclassification, while attack flows were labeled 1—8. After pre-processing, the dataset consists of 208,159 benign flows and 13,752 attack flows. Despite the imbalanced nature of the data collection, we intentionally maintained this imbalance to replicate a real-world scenario. **Table 2** illustrates the breakdown of the different traffic categories in the dataset.

4.2. Impact of dataset imbalance

The CIC-DOS2017 dataset contains a significantly higher proportion of benign traffic compared to attack flows, which closely resembles real-world network conditions where malicious activity is often subtle and infrequent. However, this imbalance poses several challenges for any ML/DL models. One major issue is the tendency to favor benign traffic, making it difficult to detect minority-class attacks effectively. As a result, the model may produce more false negatives, particularly for LR-DoS attacks, where malicious activity blends seamlessly with normal traffic. Moreover, the imbalance can create a false sense of reliability, as a model may appear highly accurate simply because most traffic is classified as benign, even if it struggles to identify actual threats.

Despite these challenges, maintaining the dataset in its naturally imbalanced state enhances the study's real-world applicability. Rather than training on artificially balanced data, the proposed model is evaluated under conditions that closely resemble operational network environments. To address the potential impact of class imbalance, we employed the following techniques:

- **Stratified K-Fold Cross-Validation:** Ensuring that each training and validation fold retains the original class distribution, preventing overfitting to benign flows while improving generalization.
- **Comprehensive Performance Metrics:** In addition to accuracy, we report precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUROC). These metrics provide a more balanced assessment of detection performance across both majority and minority classes.

By incorporating these mitigation strategies, we ensure that the proposed LSTM-FF model remains effective in detecting minority-class attacks while preserving its applicability in real-world intrusion detection scenarios.

4.3. Feature reduction

While choosing a feature is not mandatory for DL algorithms, it does contribute to accelerating the total detection time for attacks. The

Table 2

Traffic categories dataset.

x	Traffic type	Number of flows
0	Benign	208,159
1	ddossim	4,991
2	slowread	2,641
3	slowloris	1,850
4	slowheaders	1,651
5	hulk	714
6	rudu	697
7	goldeneye	630
8	Slowbody2	578

Random Forest (RF) model employs the feature importance technique to select a subset of significant features by quantifying the significance of a feature on a continuum ranging from 0 to 1, where 0 signifies insignificance, and 1 indicates indispensability. Each unique attribute receives a priority value from the RF algorithm, as shown in Table 3. Our dataset has discarded 14 features deemed unimportant.

4.4. Hyperparameter optimization

Hyperparameter optimization is the process of identifying the optimal combination of hyperparameter values to achieve maximum data performance in the shortest amount of time. Most ML algorithms have default hyperparameter values. However, default values only sometimes work well in different datasets. So, we need to adjust them to get the right combination. To optimize the performance of ML classifiers, we used a combination of grid and random searches to train the parameter settings of each classifier and identify the most optimal values. At the same time, the selection of hyperparameter values for the DL models may also rely on the ones reported in the literature and trial-and-error method. Table 4 presents the used hyperparameters for each classifier.

5. The structure and design of the proposed LSTM-FF model

This paper presents a deep learning system, LSTM-Feedforward (LSTM-FF), which employs an anomaly-based approach to detect LR-DoS attacks. The LSTM-FF model combines a Feedforward Neural Network (FFNN) with a long short-term memory recurrent neural network (LSTM-RNN).

The network architecture for binary classification consists of an LSTM layer with 64 cells, responsible for extracting LR-DoS features from the input flow. This choice is guided by standard practices in sequence modeling, where 32, 64, and 128 units are commonly used as starting points. The empirical evaluation demonstrated that 64 cells provided a strong balance between performance and computational efficiency. A first intermediate fully connected layer (F1) with 32 neurons follows, learning and compiling the extracted data from the LSTM layer. Next, a second fully connected layer (F2) consisting of 10 neurons reduces the output of the F1 layer, producing a final representation that is

Table 4
Optimal hyperparameters.

classifier	Training hyperparameters
SGD	The loss function is log_loss.
LR	Solver = saga & C = 4
DT	Criterion = gini
SVM	Kernal is RBF & C = 8
RF	Criterion = log_loss & estimators = 45
KNN	The number of neighbors = 2

passed through an output layer for classification.

To mitigate overfitting, we implement dropout layers with a dropout rate of 0.2 after the LSTM and F1 layers. Note that the typical values for dropout rate range from 0.1 to 0.5, and a rate of 0.2 was selected as it provided moderate regularization without significant loss of information. The intermediate layers in the model utilize the Rectified Linear Unit (ReLU) activation function. On the other hand, the output layer employs a sigmoid activation function with a total of 2 neurons. We replicate the values of the LSTM, F1, and F2 layers for the multi-classification task, with C = 128, 64 neurons, and 20 neurons, respectively. With nine neurons, the output layer uses SoftMax activation.

The selection of hyperparameters, such as the number of LSTM cells, dropout rates, learning rates, and other architectural details in our deep learning model, is determined through empirical evaluation to achieve optimal performance. Fig. 4 and Fig. 5 illustrate the LR-DoS attacks detection process framework and the architecture of LSTM-FF, respectively.

5.1. Performance metrics

To assess the performance of each model, we calculate several metrics, such as recall, specificity, accuracy, precision, F1 score, and false alarm rate. This process helps us gain clarity and obtain all of these performance measures. Here is an explanation and derivation of the metrics:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (6)$$

Table 3
Feature importance scores for attributes.

no.	Feature Name	Importance	no.	Feature Name	Importance	no.	Feature Name	Importance
1	Init Bwd Win Byts	0.0539	26	Subflow Bwd Byts	0.0171	51	SYN Flag Cnt	0.0069
2	Active Min	0.0470	27	Fwd IAT Tot	0.0170	52	Fwd Act Data Pkts	0.0068
3	Active Max	0.0452	28	Pkt Len Std	0.0166	53	ACK Flag Cnt	0.0049
4	Flow IAT Min	0.0379	29	Pkt SizeAvg	0.0164	54	Bwd IAT Std	0.0048
5	Fwd Header Len	0.0299	30	Fwd Pkt Len Mean	0.0164	55	Down/Up Ratio	0.0047
6	Active Mean	0.0272	31	PktLenMean	0.0159	56	Fwd Pkt Len Min	0.0025
7	Fwd Pkt Len Max	0.0267	32	Fwd IAT Min	0.0154	57	PSH Flag Cnt	0.0019
8	Fwd IAT Mean	0.0253	33	TotLen Bwd Pkts	0.0153	58	Bwd Pkt Len Min	0.0017
9	Idle Std	0.0239	34	Bwd Pkts/s	0.0153	59	Pkt Len Min	0.0012
10	Bwd Seg Size Avg	0.0226	35	Idle Mean	0.0144	60	Bwd PSH Flags	0.0011
11	Idle Max	0.0226	36	Flow IAT Std	0.0142	61	FIN Flag Cnt	0.0011
12	Flow IAT Max	0.0210	37	TotLen Fwd Pkts	0.0136	62	RST Flag Cnt	0.0001
13	Fwd Pkt Len Std	0.0208	38	Pkt Len Var	0.0135	63	Fwd PSH Flags	0.0000
14	Idle Min	0.0199	39	Fwd Seg Size Avg	0.0125	64	Fwd Seg Size Min	0.0000
15	Flow Pkts/s	0.0198	40	Fwd IAT Std	0.0122	65	Bwd Pkts/b Avg	0.0000
16	Bwd Pkt Len Mean	0.0195	41	Flow Byts/s	0.0119	66	Fwd Byts/b Avg	0.0000
17	Flow Duration	0.0195	42	Subflow Fwd Byts	0.0116	67	URG Flag Cnt	0.0000
18	Active Std	0.0188	43	Bwd IAT Max	0.0113	68	CWE Flag Count	0.0000
19	Fwd IAT Max	0.0188	44	Tot Fwd Pkts	0.0110	69	Init Fwd Win Byts	0.0000
20	Pkt Len Max	0.0187	45	BwdPktLenStd	0.0104	70	ECE Flag Cnt	0.0000
21	Bwd Header Len	0.0182	46	BwdIATMin	0.0102	71	Fwd Blk Rate Avg	0.0000
22	Fwd Pkts/s	0.0181	47	Bwd IAT Tot	0.0080	72	Fwd URG Flags	0.0000
23	Flow IAT Mean	0.0178	48	Subflow Fwd Pkts	0.0077	73	Bwd Byts/b Avg	0.0000
24	Bwd IAT Mean	0.0174	49	Tot Bwd Pkts	0.0075	74	Fwd Pkts/b Avg	0.0000
25	Bwd Pkt Len Max	0.0173	50	Subflow Bwd Pkts	0.0070	75	Bwd URG Flags	0.0000
						76	Bwd Blk Rate Avg	0.0000

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

$$\text{Recall} = \text{Sensitivity} = \frac{TP}{TP + FN} \quad (8)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (9)$$

$$\text{FAR} = \frac{FP}{TN + FP} \quad (10)$$

$$\text{F1score} = 2 \times \frac{(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \quad (11)$$

5.2. Implementation

The performance of the proposed LSTM-FF was evaluated using the CIC-DOS2017 dataset for binary and multiple-class classification experiments. The keras and Scikit-learn DL and ML python libraries are utilized to implement various models. The tests were conducted on Google Colab; the Adam method was used as an optimizer for adjusting the model weights. The learning rate was set to 0.001, and batch number 512 was utilized. The model was trained for 20 epochs for binary classification, while for multi-class classification, it was trained for 30 epochs. This paper manuscript employs sparse categorical cross-entropy as its loss function.

We validate the model's performance using a stratified k-fold procedure, setting k to 10. The stratified k-fold maintains a consistent sample percentage for each class across all folds. We used an early stopping technique to terminate the model when the training loss did not decrease during the training phase. We conducted all the simulations using an HP Pavilion x360 Convertible 14-ba1xx with Windows 10 OS, an Intel (R) Core (TM) i5-8250U CPU @ 1.60 GHz, 1800 MHz, four cores, eight logical processors, and 8 GB of RAM.

6. Experiments, numerical results, and analysis

This section provides an overview of the LSTM-FF model's implementation and presents an analysis of the simulation results. To assess its effectiveness, we compare the model's performance against conventional ML algorithms, all developed and validated using the CIC-DOS2017 dataset.

6.1. Experiment 1: Binary classification

For this experiment, 62 features from the CIC-DOS2017 dataset were employed to perform binary classification (anomaly and normal). The classification's performance is assessed using the confusion matrix as illustrated in Fig. 6.

Table 5 tabulates the LSTM-FF model performance to conventional ML algorithms (i.e., SDG, LG, DT, RF, SVM, and KNN). The model shows better performance than other machine learning models. The F1 rating is 97.52 %, accuracy is 99.70 %, and recall is 95.64 %, which indicates the model's sensitivity to detect anomalies or attacks. This adds to the reality that the LSTM-FF model has a lower rate of false alarms (0.03 %), which is ideal for intrusion detection systems. GRU and RNN follow LSTM on the performance list, with accuracy scores of 99.69 % and 99.68 %, recall scores of 95.86 % and 95.78 %, and F1 scores of 97.49 % and 97.41 %, respectively. KNN outperformed the other machine learning models, scoring 97.10 % for the F1 score, 99.65 % for the accuracy, and 94.84 for the recall.

RF, SVM, and DT followed, with accuracy scores of 99.61 %, 99.56 %, and 99.47 %, recall scores of 94.91 %, 94.26 %, and 95.78 %, and F1 scores of 96.78 %, 96.40 %, and 95.75 %, respectively. The DT has higher sensitivity to detect attacks than the LSTM but detects less normal

traffic, resulting in a higher false alarm rate (FAR = 0.28 %) than the LSTM. The LR and SGD performed the worst among the evaluated approaches, receiving an accuracy score of 97.54 % and 97.39 %, respectively. Their recall scores of 65.45 % and 63.59 % indicate they cannot accurately detect LR-Dos attacks.

These two methods perform relatively poorly. It's crucial to note that both the KNN and LSTM models display identical specificity, precision, and FAR scores. However, the LSTM achieves a higher ROC-AUC score of 99.74 % compared to the KNN's 97.40 %. Fig. 7 illustrates the comparison of performance rates for binary classification.

6.2. The ROC curves

The ROC curve is a graphical representation of the performance of a classification model at different threshold values based on probabilities. The Receiver Operating Characteristics (ROC) curve is used to display the relationship between the False Positive Rate (FPR) and the True

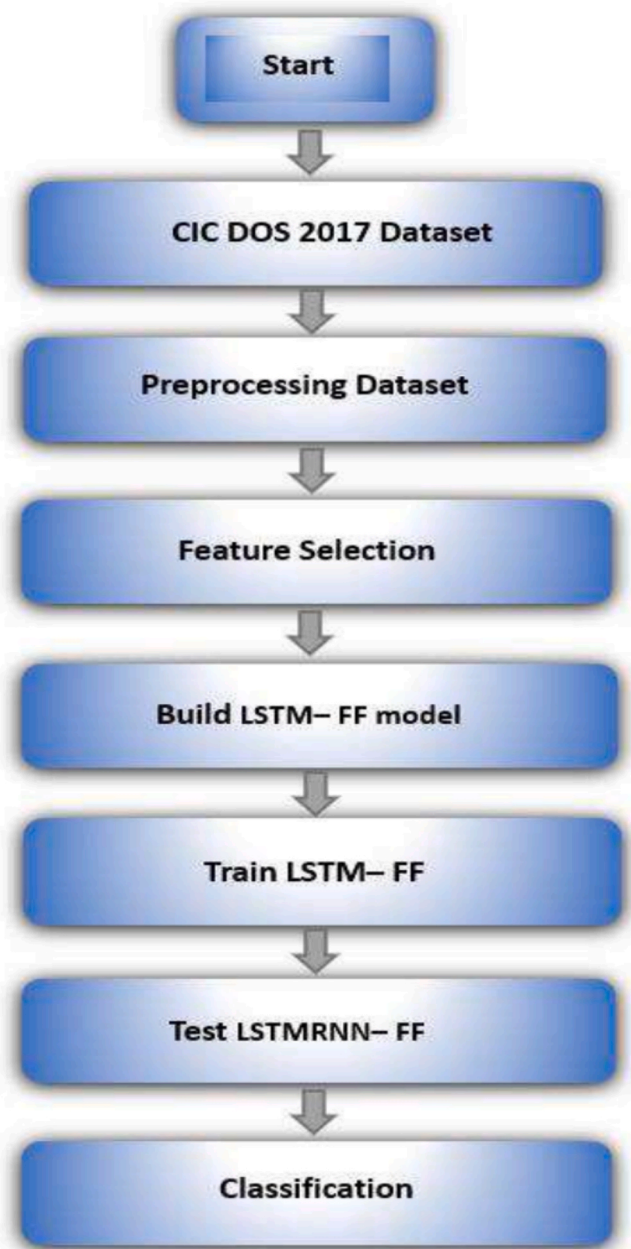


Fig. 4. LSTM- FF framework.

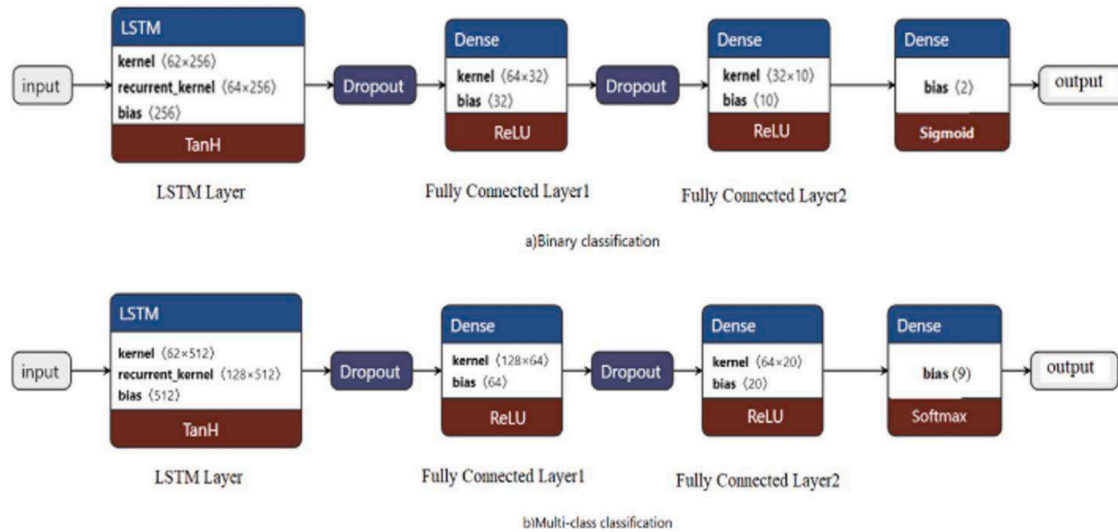


Fig. 5. LSTM- FF architecture.

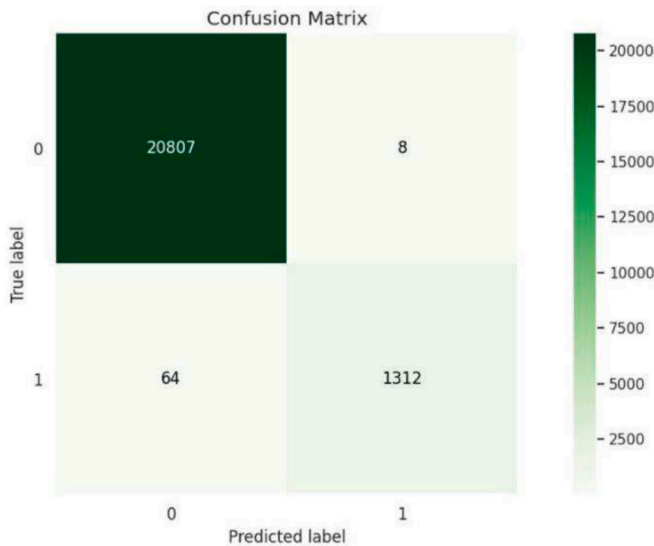


Fig. 6. Confusion matrix for the binary classification.

Positive Rate (TPR) at various categorization thresholds. The ROC's Area Under the Curve (AUC) measures the degree of separability. The AUC metric quantifies the model's ability to differentiate between different classes. The LSTM-FF model has a ROC-AUC of 99.74 %, indicating exceptional classification performance. GRU and RNN achieved accuracy rates of 99.72 % and 98.79 % respectively. Except for SGD and LR, which demonstrate lower performance at 81.60 % and

82.56 %, respectively, most other machine learning models exhibit ROC-AUCs ranging from 97.40 % to 97.75 %, indicating a high level of performance. Fig. 9 depicts the receiver operating characteristics area under the curve (ROC-AUC) for the LFTM_FF model, while Fig. 10 displays the ROC curve for the remaining machine learning models. The ROC curve of our model exhibits a very high true positive rate (TRP) with a significantly low false positive rate (FPR), making it an optimal choice for an IDS.

6.3. A decision threshold for ML algorithms

When working with imbalanced data sets, it is advisable to modify the threshold for the ML models to identify the most effective classifier. Using the ROC curve is one of the main approaches to achieving this. An ideal classifier's Receiver Operating characteristics (ROC) curve should exhibit a high True Positive Rate (TPR) and a low False Positive Rate (FPR). The rationale for utilizing the ROC curve to adjust the threshold is to identify the specific point on the curve that corresponds to the upper-left corner mathematically, determining the threshold value P that fulfills the equation:

$$oneTPR(p) = 1 - FPR(p) \quad (12)$$

where sensitivity = TPR and specificity = 1.

So, we want to calculate the threshold which satisfies the following:

$$P = \operatorname{argmin} |TPR(p) + FPR(p) - 1| \quad (13)$$

Table 6 presents the confusion metrics produced for the models when functioning at their optimal classification threshold. Fig. 10 explains how the FPR and TPR relate to the threshold indicating the upper-left corner using a '•' mark.

Table 5
Comparison of LSTM-FF performance with other ML models for binary classification.

	Accuracy	Precision	Recall	F1 score	Specificity	FAR	ROC-AUC
SGD	97.39 %	91.72 %	63.59 %	75.11 %	99.62 %	0.38 %	81.60 %
LR	97.54 %	92.78 %	65.45 %	76.76 %	99.66 %	0.34 %	82.56 %
DT	99.47 %	95.72 %	95.78 %	95.75 %	99.72 %	0.28 %	97.75 %
SVM	99.56 %	98.63 %	94.26 %	96.40 %	99.91 %	0.09 %	97.08 %
RF	99.61 %	98.72 %	94.91 %	96.78 %	99.92 %	0.08 %	97.41 %
KNN	99.65 %	99.47 %	94.84 %	97.10 %	99.97 %	0.03 %	97.40 %
RNN	99.68 %	99.10 %	95.78 %	97.41 %	99.94 %	0.06 %	98.79 %
GRU	99.69 %	99.17 %	95.86 %	97.49 %	99.95 %	0.05 %	99.72 %
LSTM	99.70 %	99.47 %	95.64 %	97.52 %	99.97 %	0.03 %	99.74 %

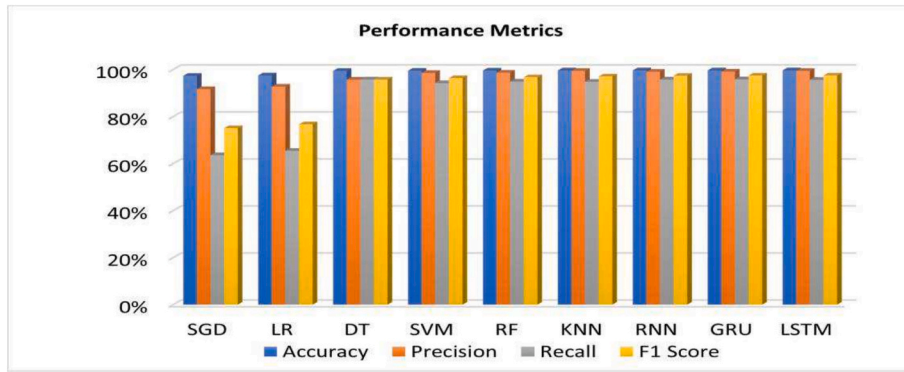


Fig. 7. The evaluation of the accuracy, precision, recall, and F1- measures for each approach in binary classification.

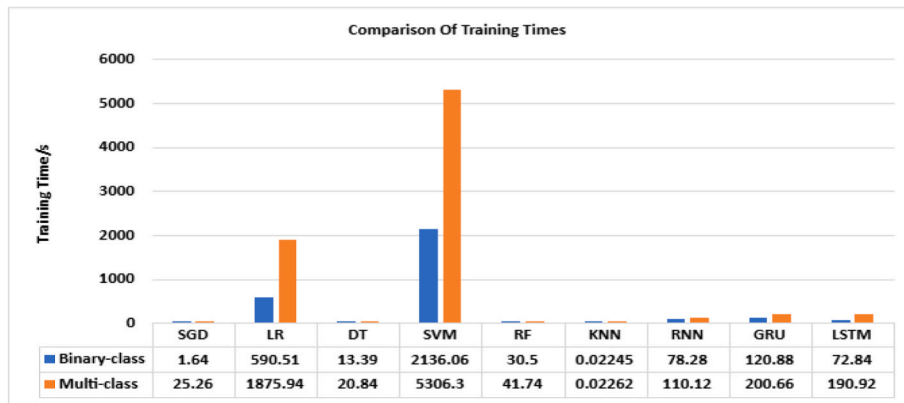


Fig. 8. An analysis of the training durations of machine learning (ML) and deep learning (DL) models.

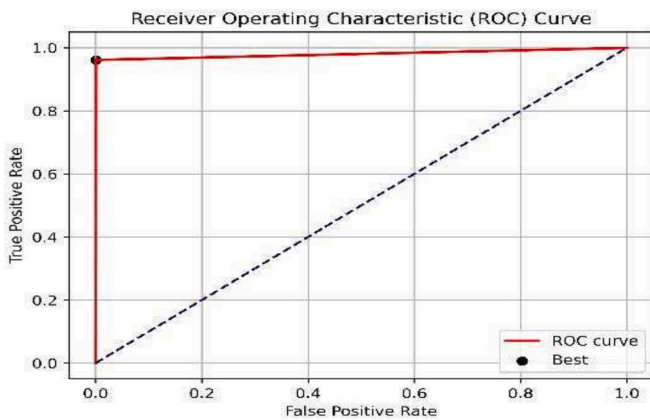


Fig. 9. The receiver operating characteristic (ROC) area under the curve (AUC) of the LSTM-FF model.

The ROC-AUC of SGD is 85.57 %, and the LR is 86.53 %, indicating comparatively poor classification performance. DT worked very close to its optimal threshold value since there was a slight difference in its performance metrics and an ROC-AUC of 97.77 %. The SVM has a higher recall than DT but a higher FAR rate of 1.17 %, and at the same time, it takes a very long time to train. The ROC-AUC for RF and KNN increased from 97.41 % to 98.22 % and 97.40 % to 97.71 % respectively. So, DT and KNN were the best methods. Each of these has a preferred feature for IDs. While DT has the lowest False Alarm Rate (FAR), KNN has a low training time. Fig. 11 shows how the machine-learning method's sensitivity, specificity, and ROC-AUC change when the default threshold-1 (0.5) point is used versus the optimal threshold-2 point that

was calculated.

6.4. Experiment 2: Multiclass classification

The experiment employs the same 62 features from the CICDOS2017 dataset previously utilized for binary classification. Fig. 12 shows the confusion matrix for different class types, and Table 7 compares the performance of the proposed methodology with conventional methods.

Additionally, Fig. 13 illustrates the performance comparison for multi-class classification.

All techniques except for SGD and LR obtained accuracy rates exceeding 99 % based on the observations. Specifically, RF achieved an accuracy rate of 99.68 %, KNN 99.64 %, LSTM 99.54 %, GRU 99.51, RNN 99.48 %, SVM achieved 99.37 %, and DT reached 99.33 %. The RF approach demonstrates superior precision compared to the other models, achieving a rate of 97.10 %. KNN followed closely behind with a rate of 95.52 %. The SVM and LSTM algorithms achieved precision rates of 93.63 % and 93.09 %, respectively, followed by GRU at 92.10 %, RNN at 91.88 %, and DT at 90.52 %. The Random Forest (RF) algorithm achieved the highest performance in terms of the recall measure with an accuracy of 92.66 %. The K-Nearest Neighbors (KNN) algorithm is closely followed with 91.45 %. The rates for the DT, LSTM, RNN, GRU, and SVM models were 89.52 %, 88.01 %, 86.49 %, 85.49 %, and 84.61 % respectively. The recall rates for Logistic Regression (LR) and Stochastic Gradient Descent (SGD) were 64.65 % and 43.55 % respectively.

For the F1 score, RF achieved better outcomes with a 94.71 % rate, followed by KNN at 93.37 %, LSTM at 90.28 %, and DT at 89.94 %. The RNN, SVM, and GRU have close rates of 88.96 %, 88.68 % and 88.34 %. As we can see, while the RF and KNN methods perform slightly better than the LSTM, the LSTM method still produces balanced results with a low False Alarm Rate (FAR) of 0.03 %.

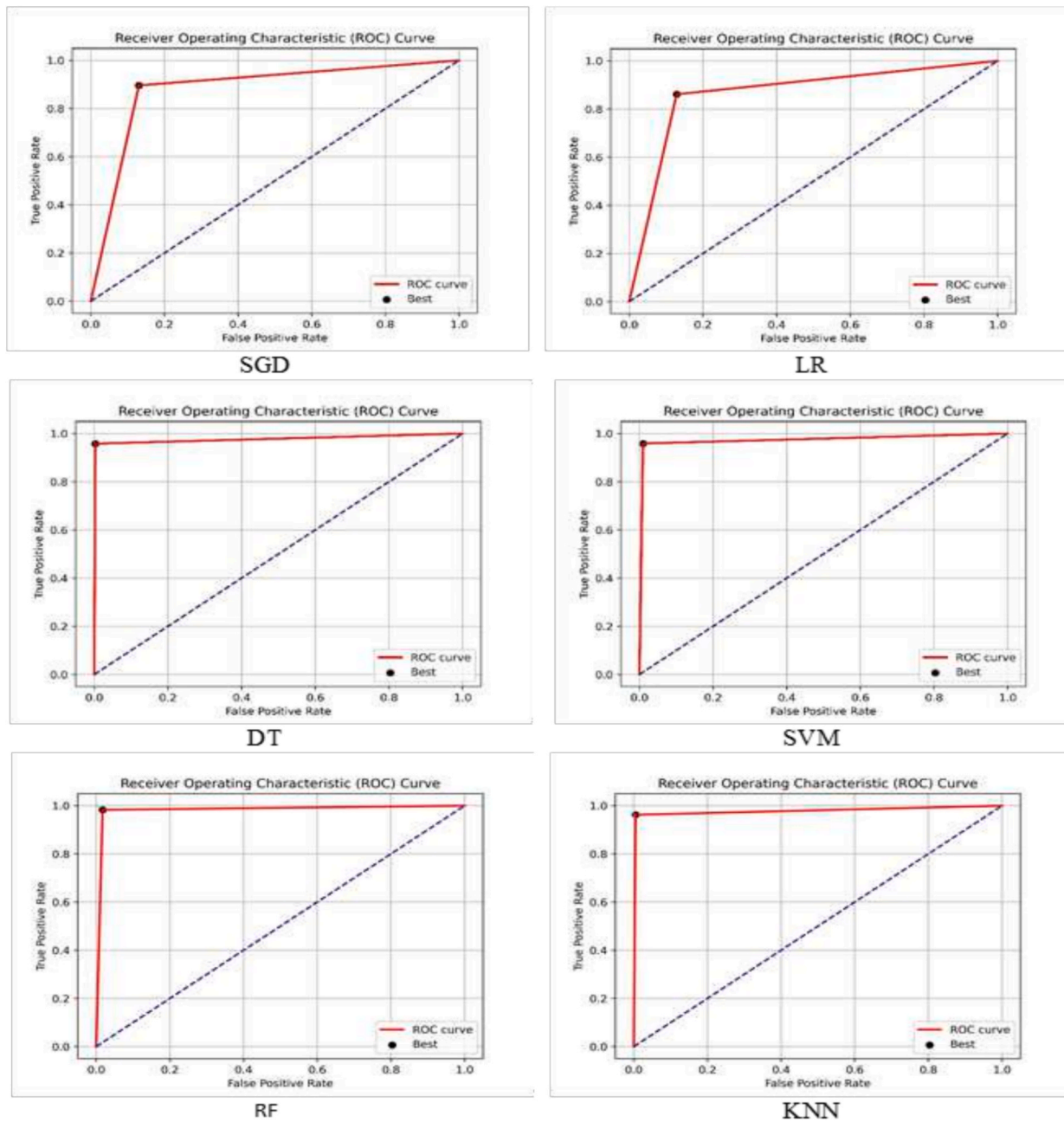


Fig. 10. ROC-AUC for various machine learning classifiers.

Table 6

The performance metrics for the ML models at their optimal classification threshold.

	Accuracy	Precision	Recall	F1-score	Specificity	FAR	ROC-AUC
SGD	85.93 %	28.64 %	85.17 %	42.88 %	85.97 %	14.03 %	85.57 %
LR	87.54 %	31.41 %	85.38 %	45.92 %	87.68 %	12.32 %	86.53 %
DT	99.45 %	95.37 %	95.86 %	95.61 %	99.69 %	0.31 %	97.77 %
SVM	98.65 %	84.45 %	95.93 %	89.83 %	98.83 %	1.17 %	97.38 %
RF	98.32 %	79.60 %	98.11 %	87.89 %	98.34 %	1.66 %	98.22 %
KNN	99.28 %	92.70 %	95.93 %	94.29 %	99.50 %	0.50 %	97.71 %

6.5. Training duration analysis

Training time is a crucial metric in evaluating the computational efficiency of ML and DL models, particularly for real-time IDSs. Models with excessive training times may be impractical for dynamic network environments, where timely adaptation is essential.

Fig. 8 illustrates the training duration of various ML and DL models,

highlighting the balance achieved by the proposed LSTM-FF architecture. Traditional ML models, such as Decision Trees and Random Forest, exhibit the shortest training times due to their simpler structures and fewer parameters. Conversely, transformer-based DL models require significantly longer training due to their complexity and large parameter space.

The proposed LSTM-FF model strikes a balance between

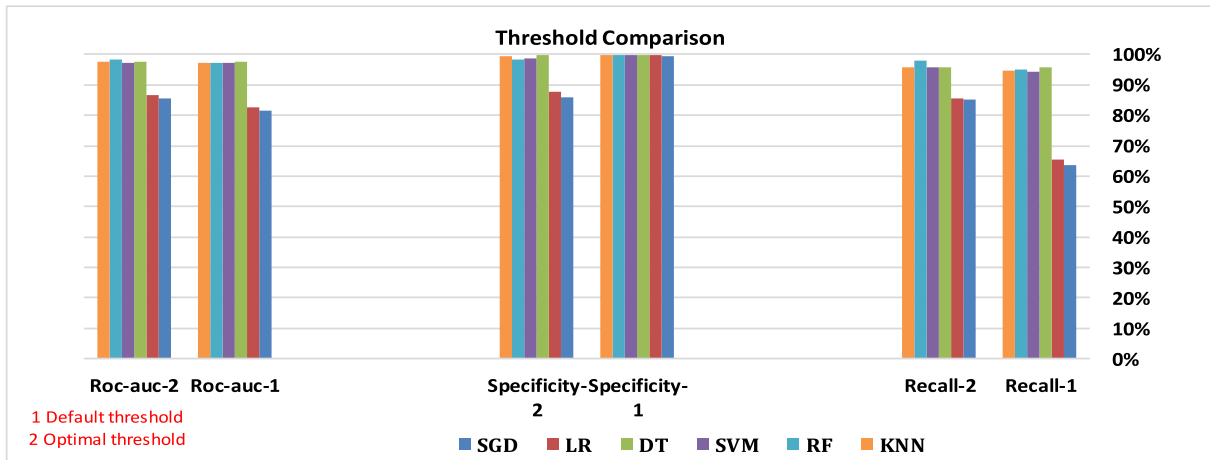


Fig. 11. Comparison of sensitivity, specificity, and ROC-AUC for different threshold.

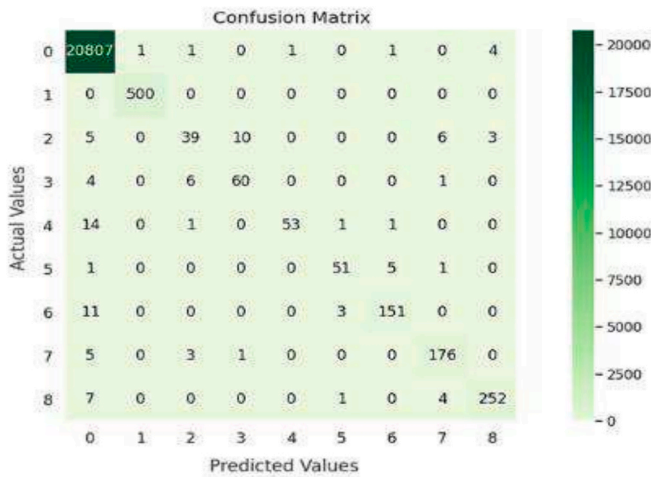


Fig. 12. The confusion matrix for multi-class classification.

computational cost and efficiency. While its training time exceeds that of traditional ML models, it remains considerably lower than more complex architectures like Transformers. This trade-off ensures that the model remains viable for real-time or near-real-time network intrusion detection, where both performance and efficiency are critical.

6.6. Comparison with other DL architectures

To provide a thorough comparison with other deep learning architectures, we conducted a survey that included GRUs, Convolutional Neural Networks (CNNs), transformer-based models, and Autoencoders. We analyzed their applicability to network intrusion detection and compared their key characteristics.

Table 7

Performance comparison between LSTM-FF's multi-classification performance with other ML models.

	Accuracy	Precision	Recall	Specificity	F1-score	FAR
SGD	96.75 %	67.07 %	43.55 %	95.15 %	50.34 %	4.85355 %
LR	97.90 %	84.83 %	64.65 %	97.32 %	71.66 %	2.68098 %
DT	99.33 %	90.52 %	89.52 %	99.44 %	89.94 %	0.55964 %
SVM	99.37 %	93.63 %	84.61 %	99.24 %	88.68 %	0.75814 %
RF	99.68 %	97.10 %	92.66 %	99.59 %	94.71 %	0.41483 %
KNN	99.64 %	95.52 %	91.45 %	99.57 %	93.37 %	0.43399 %
RNN	99.48 %	91.88 %	86.49 %	99.36 %	88.96 %	0.64441 %
GRU	99.51 %	92.10 %	85.49 %	99.56 %	88.34 %	0.44104 %
LSTM	99.54 %	93.09 %	88.01 %	99.59 %	90.28 %	0.40723 %

GRUs, like LSTMs, are designed to capture temporal dependencies but offer reduced computational complexity, making them a lighter alternative for processing sequential data. CNNs excel in extracting spatial features but are less suited for sequential dependencies in network traffic. Transformer-based models use self-attention mechanisms to capture long-range dependencies and process sequences in parallel, making them highly efficient for large datasets. However, their computational complexity and large parameter size pose challenges for real-time IDSs. Autoencoders, commonly used for anomaly detection, can identify abnormal traffic patterns but lack interpretability in distinguishing attack types. Note that the decision to adopt LSTM in this paper was based on its strong balance between temporal feature extraction and computational efficiency, making it a suitable choice for real-time IDSs.

6.7. Robustness analysis with other datasets

While the earlier discussion focused on the primary dataset, CIC-DOS-2017, we have also evaluated the proposed model on additional datasets, namely CSE-CIC-IDS2018 [46] and LR-HR DDoS-2024 for SDN-based networks [47]. This is mainly to account for evolving attack trends and modern network environments, as well as to verify the robustness and generalizability of the proposed LSTM-FF model.

a) CSE-CIC-IDS2018 dataset

The CSE-CIC-IDS2018 dataset is a widely recognized benchmark for research on IDS. This dataset was developed through collaboration between the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC). Its purpose is to simulate real-world cyberattacks on enterprise networks, offering labeled data that supports machine learning and cybersecurity research. The dataset includes both normal and malicious traffic collected over five days, from

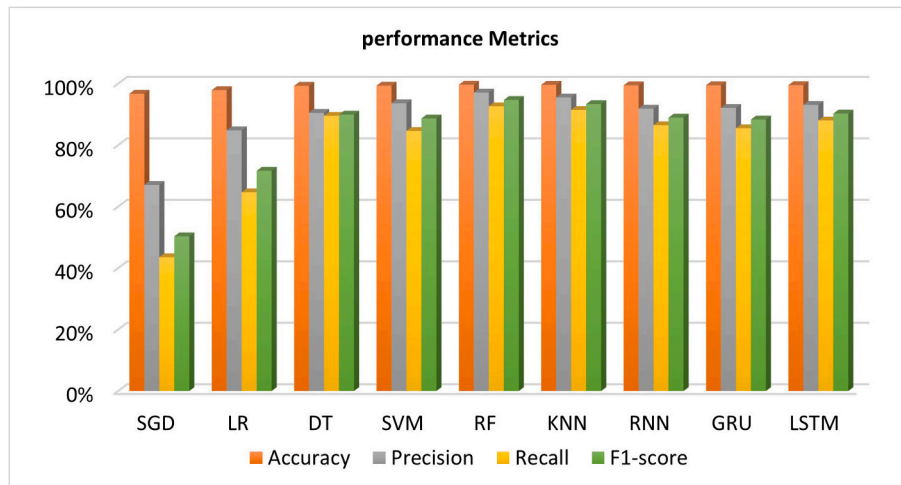


Fig. 13. The evaluation of the accuracy, precision, recall, and F1- measures for each approach in multi-class classification.

February 12 to February 16, 2018. It covers multiple attack families, including brute force attacks, Denial-of-Service (DoS) attacks, botnets, and infiltration attacks. Two files were downloaded: “Thursday-15-02-2018_” and “Friday-16-02-2018_”. Each file contains 1,048,575 instances and includes 80 features, one of which is the label column. The first file contains two types of DoS attacks: Slowloris and GoldenEye. The second file features two different DoS attacks: Hulk and SlowHTTPTest.

A total of 62 features, previously utilized in the processing of the CIC-DOS-2017 dataset, were selected also for this experiment. After pre-processing, which involved checking for infinite, null, and duplicate values, the dataset consisted of 1,154,831 benign flows and 196,518 attack flows, which were used for model training. The model’s performance was validated using a stratified k-fold procedure with k set to 10.

b) LR-HR DDOS –2024 dataset

LR-HR DDOS 2024 is a comprehensive dataset that includes both Low-Rate and High-Rate DoS and DDoS attacks in the context of Software-Defined Networking (SDN). This dataset reflects real-world conditions and incorporates various attack vectors and intensities. It has been systematically constructed to document details on attack generation, network setup, and parameters for efficient usage.

The downloaded dataset file contains 113,408 instances: 42,900 for normal traffic and 70,508 for malicious traffic. It features 24 attributes, including a label column where 0 indicates normal traffic and 1 indicates malicious traffic for binary classification. After pre-processing, which involves checking for infinite, null, and duplicate values, the dataset comprises 42,890 benign flows and 987 attack flows. For our experiment, we randomly selected 3,290 benign samples and used the 987 samples representing malicious attacks. This selection maintains a 70:30 ratio between the two types of traffic flows. The model’s performance was validated using a stratified k-fold procedure, with k set to 10.

To further assess the robustness of our proposed LSTM-FF model, we evaluated its performance on two additional datasets, as detailed earlier. The results of these evaluations are summarized in Table 8.

7. The performance complexity

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) commonly used for sequential tasks. They are adapting to modeling long-term dependencies. However, they have limitations, such as computational complexity and memory requirements. The primary limitations are as follows:

• Computational complexity

Due to their recurrent nature, LSTM networks require more computational resources compared to feedforward neural networks. The hidden state propagation in LSTMs occurs across multiple time steps, leading to a linear increase in computational cost as the sequence length grows. Processing longer sequences can be particularly challenging and may demand substantial computational resources.

• Difficulty in capturing extremely long-term dependencies

Although LSTM networks are explicitly designed to capture temporal dependencies, they may encounter challenges in modeling dependencies. The issue is that LSTM may struggle to effectively store and transmit information as the time between essential events increases. This is known as the “vanishing gradient” problem, which occurs when gradients decrease exponentially as they move backward in time, making it difficult to learn from far-reaching dependencies.

• Sensitivity to hyperparameter

There are several hyperparameters in LSTM networks that require tuning, including learning rate, regularization parameters, and memory cell count. Determining the ideal collection of hyperparameters can be challenging and time-consuming. Inadequate selection of hyperparameters can result in substandard operation or make it harder to train the network efficiently.

• Adaptability

The model must be highly adaptable because network environments constantly change, and the behaviors of both normal and malicious activities may evolve. To build an effective IDS using LSTMs, it should be able to retrain or adjust its parameters to accommodate these changes. One of the problems with traditional

Table 8

Performance of the LSTM-FF model on the datasets.

	Accuracy	Precision	Recall	F1 score	Specificity	FAR	ROC-AUC
CIC-DOS2017	99.70 %	99.47 %	95.64 %	97.52 %	99.97 %	0.03 %	99.74 %
CSE-CIC-IDS2018	99.83 %	99.50 %	99.86 %	99.68 %	99.86 %	0.0017 %	99.98 %
LR-HR 2024	99.76 %	98.98 %	100 %	99.49 %	99.84 %	0.0030 %	99.99 %

LSTM networks is that they can't handle long-term dependencies. To fix this, researchers are working on better model architectures and training methods, like adding complex recurrent units like GRU and studying how attention works. Despite these advancements, traditional LSTM networks still have limitations. Ongoing research aims to enhance the interpretability. Efficiency and flexibility of LSTM models in cybersecurity.

8. Conclusion and recommendation

In this paper, we proposed the LSTM-Feedforward (LSTM-FF) deep learning-based anomaly detection scheme to identify LR-DoS attacks. Using the feature importance method from the Random Forest (RF) model, we demonstrated that LSTM-FF achieves strong detection performance with 62 selected features from the CIC-DOS2017 network flow dataset. To further evaluate its robustness and generalizability, we tested the model on two additional datasets, namely the CSE-CIC-IDS2018 dataset and the LR-HR DDoS-2024 dataset for SDN-based networks. The detection module leverages a Long Short-Term Memory (LSTM) network, a type of Recurrent Neural Network (RNN) designed to handle sequential data while retaining long-term dependencies. Additionally, we compared LSTM against other RNN variants, including Simple RNN and GRU. While all models performed well, LSTM demonstrated the best overall effectiveness in detecting LR-DoS attacks across multiple datasets.

To evaluate the effectiveness of the proposed LSTM-FF model, we compared it against six different machine learning approaches: SVM, LR, KNN, SGD, DT, and RF. All models were tested on the same dataset in two experiments—binary classification and multi-class classification. The results show that LSTM-FF outperforms all other methods in binary classification, with DT and KNN following closely behind. For multi-

class classification, RF and KNN achieve slightly better performance than LSTM-FF. However, LSTM-FF remains one of the top-performing models for LR-DoS detection, demonstrating strong accuracy and a notably low False Alarm Rate (FAR) of 0.03 %. Overall, the proposed model significantly enhances anomaly detection. One of the key challenges in using deep learning for IDS design is the high training time, and future work could explore Federated Learning as a potential solution to mitigate this issue.

CRedit authorship contribution statement

Suhaila ZeinElabideen Omer: Writing – original draft, Validation, Software, Methodology, Formal analysis, Data curation. **Fazirulhisyam Hashim:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Aduwati Sali:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Faisal Arif Ahmad:** Supervision, Conceptualization.

Declaration of competing interest

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

Acknowledgments

The authors would like to thank Universiti Putra Malaysia for the financial support under Journal Publication Fund (DPJ) and Geran Putra Berimpak with Project Code: UPM.RMC.800-3/3/3/1/2024/GPB/9784100.

Appendix A

Methodologies and Limitations of Current Research

Ref.	Dataset	Attack (H/L Rate)	Model Architecture	Evaluation Metrics	Classification	Limitations/Challenges
[25]	NSL-KDD	H	Different deep neural network structures (CNNs) Autoencoders (RNNs)	ROC curve, AUC, precision-recall curve, mean average precision (MAP)	Binary/Multi	- While the NSLKDD dataset is an improvement over KDDCUP99, it still has limitations and may not accurately represent real-world network traffic. - The performance of the models on unseen attacks could be affected by the variety of attack types present in the NSLKDDTest21 dataset that were not included in the training data.
[26]	NSL-KDD	H	Feedforward Deep Neural Networks (FFDNNs)	Accuracy, precision, recall, precision-recall curve	Binary/Multi	- The NSL-KDD dataset is not recent and has significant limitations. - The evaluation metrics used to assess performance rely on the accuracy and various classification measures, which may not fully capture the models' ability to detect rare or sophisticated attacks.
[27]	NSL-KDD And ISCX	H	Traditional RNN, LSTM, and GRU	Accuracy, confusion matrix curve, memory profiler	Binary/Multi	- The proposed Intrusion Detection System (IDS) framework struggles with high false positive rates and faces challenges in accurately detecting complex R2L and U2R attacks. - The model's reliance on deep learning techniques requires substantial computational resources, which may not be practical for all environments.
[28]	KDD99 and NSL-KDD	General DoS attacks	RNN with GRU units, (MLP), and a SoftMax layer	Accuracy, precision, detection rate (DR), false	Binary/Multi	- The detection of R2Land U2R attacks is challenging due to the smaller number of

(continued on next page)

(continued)

Ref.	Dataset	Attack (H/L Rate)	Model Architecture	Evaluation Metrics	Classification	Limitations/Challenges
				positive rate (FPR), confusion matrix		records available in datasets compared to other attack types, making feature extraction more difficult.
[29]	UNSW-NB15	Detection of various types of attacks	A Bidirectional Gated Recurrent Unit (BiGRU) with a Hierarchical Attention Mechanism	Accuracy, precision, recall, and false alarm rate (FAR)	Binary/Multi	- The KDD99 and NSL-KDD datasets used for evaluation may not fully represent real-world networks, which could affect performance. - Employing deep neural networks like GRUs and MLPs increases the complexity of the model, leading to challenges regarding training time and the requirement for computational resources. - The UNSW-NB15 dataset, which may not accurately reflect real-world network environments, was used to evaluate the model. - The model performs poorly in identifying certain types of attacks, particularly low-frequency attacks. - The hierarchical attention mechanism and the use of BiGRU introduce a significant level of complexity.
[30]	CIC-DDoS2019	H	GRU	Accuracy, precision, recall, and F1-score.	Binary/Multi	- The model was tested using the CICDDoS2019 dataset. However, the imbalanced distribution of attack types within this dataset reduces the model's effectiveness in detecting certain low-frequency attacks. - Additionally, employing GRU and other deep learning models adds significant complexity. This complexity can pose challenges in terms of the time and computational resources needed for both training and deployment
[31]	KDDCup and UNSW-NB15	H	Wide & Deep Transfer Learning Stacked GRU Framework	Accuracy, precision, recall, F1-score, and confusion matrix.	Binary/Multi	- Dataset Limitations: The KDD Cup 99 (10 %) and UNSW-NB15 datasets are older datasets and may not accurately represent today's real-world network environment. - Detection Effectiveness: The model is less effective at detecting low-frequency attacks due to the imbalanced distribution of attack types. - Model Complexity: The high complexity of the wide and deep transfer learning framework using stacked GRUs poses challenges in terms of computational resources, as well as training and deployment time.
[32]	UNSW-NB15, NSL-KDD, and InSDN	H	A hybrid deep learning approach combining CNN and Bidirectional Long Short-Term Memory (BiLSTM)	Accuracy, precision, recall, F1-score, and training time	Binary/Multi	- Redundant and imbalanced data in datasets can negatively impact model performance and flexibility. - The hybrid CNN-BiLSTM model requires a substantial number of training parameters, which can slow down the training process and increase computational costs. - Although the model shows promising results in theoretical evaluations, additional work is needed to improve and validate its practical application in real-time Software-Defined Networks.
[33]	KDD CUP 1999 and CSE-CIC-IDS 2018	H/L	A CNN is compared with an RNN	Accuracy, precision, recall, F1-score, and confusion matrix	Binary/Multi	- The study primarily utilizes the KDD CUP 1999 and CSE-CIC-IDS 2018 datasets. While these datasets are comprehensive, they may not fully represent the diversity and complexity of real-world network traffic and attacks. - The process of converting network traffic features into images for processing with convolutional neural networks (CNN) is innovative, but it may overlook some important aspects of the data. As a result, crucial features might be lost or not adequately represented in the image format.

(continued on next page)

(continued)

Ref.	Dataset	Attack (H/L Rate)	Model Architecture	Evaluation Metrics	Classification	Limitations/Challenges
[34]	NSL-KDD and UNSW-NB15	H/L	This model combines CNN and Bidirectional Long Short-Term Memory (BiLSTM)	Accuracy, precision, recall, F1-score	Binary/Multi	- The NSL-KDD and UNSW-NB15 datasets are not recent and have significant limitations. - The transformation of network traffic features into a format suitable for CNN and BiLSTM processing is innovative but may not capture all relevant aspects of the data. Important features might be lost or not adequately represented. - The scalability of the CNN model to handle large-scale network traffic in real-time is not addressed.
[35]	CIC-DoS dataset	L	Machine learning models: utilize six different models: J48, Random Tree, Rep Tree, Random Forest, MLP (Multilayer Perceptron), and SVM	Accuracy, precision, recall, F1-score, and false alarm rate (FAR).	Binary	- The paper does not outline a specific feature selection method, which could impact the performance and accuracy of the machine learning models employed. - The proposed architecture has been tested in a simulated environment; however, its performance and scalability in a real-world, large-scale network setting still need to be validated.
[36]	Real Traffic Dataset	L	Hybrid Deep Neural Networks: Combines CNN and GRU	Accuracy, precision, F1-score, detection rate (DR), false positive rate (FPR), and false negative rate (FNR)	Binary	- The proposed method is evaluated using traffic captured from a university website and LDoS attacks generated in a laboratory setting. However, this may not fully represent the diversity and complexity of real-world network traffic. - The study specifically focuses on certain LDoS attack programs: Slowloris, Slowhttptest, Pwnloris, Torshammer, and Httpbog. As a result, other types of LR-DoS attacks or variations may not be effectively detected by the proposed method.
[37]	InSDN, CSE-CIC-IDS2018, and UNSW-NB15	L	Hybrid Deep Learning Model: Combines CNN with machine learning algorithms (SVM, KNN, RF)	Precision, recall, F1-score, area under the curve (AUC), training time, and testing time	Binary/Multi	- The paper does not provide a detailed feature selection method, which could impact the performance and accuracy of the machine learning models utilized. - While the architecture achieves high accuracy, the false alarm rate remains significant, which could result in legitimate traffic being incorrectly classified as malicious.
[38]	CIC-DDOS2019 and CIC-IDS2018	H/L	Deep learning model Recurrent Units (GRU)	Accuracy, precision, recall, F1-measure, sensitivity, specificity	Binary	- The system is designed to analyze individual IP flows, but this approach may not detect more complex attack patterns that involve multiple flows. - The paper does not provide a specific method for feature selection, which could hinder the model's ability to generalize across different datasets.

References

- [1] M.Y. Aldarwbi, A.H. Lashkari, A.A. Ghorbani, The sound of intrusion: A novel network intrusion detection system, *Comput. Electr. Eng.* 104 (Dec. 2022), <https://doi.org/10.1016/j.compeleceng.2022.108455>.
- [2] S. Kumar, S. Gupta, and S. Arora, "Research Trends in Network-Based Intrusion Detection Systems: A Review," *IEEE Access*, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 157761–157779, 2021. doi: 10.1109/ACCESS.2021.3129775.
- [3] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets, and challenges, *Cybersecurity* 2 (1) (2019) Dec, <https://doi.org/10.1186/s42400-019-0038-7>.
- [4] Z. Ahmad, A.S. Khan, C.W. Shiang, J. Abdullah, F. Ahmad, Network intrusion detection system: a systematic study of machine learning and deep learning approaches, *Trans. Emerg. Telecommun. Technol.* 32 (1) (2021) e4150.
- [5] H. Liu, B. Lang, Machine learning and deep learning methods for intrusion detection systems: a survey, *Appl. Sci.* 9 (20) (2019) 4396, <https://doi.org/10.3390/app9204396>.
- [6] A.B. Nassif, M.A. Talib, Q. Nasir, F.M. Dakalbab, Machine learning for anomaly detection: a systematic review, *IEEE Access* 9 (2021) 78658–78700, <https://doi.org/10.1109/ACCESS.2021.3083060>.
- [7] L. Chen, Y. Zhang, Q. Zhao, G. Geng, and Z. Yan, "Detection of DNS DDoS Attacks with Random Forest Algorithm on Spark," in *Procedia Computer Science*, Elsevier B.V., 2018, pp. 310–315. doi: 10.1016/j.procs.2018.07.177.
- [8] P. Tao, Z. Sun, Z. Sun, An improved intrusion detection algorithm based on GA and SVM, *IEEE Access* 6 (2018) 13624–13631, <https://doi.org/10.1109/ACCESS.2018.2810198>.
- [9] X. Wang, C. Zhang, K. Zheng, Intrusion detection algorithm based on density, cluster centers, and nearest neighbors, *China Commun.* 13 (7) (2016) 24–31, <https://doi.org/10.1109/CC.2016.7559072>.
- [10] T. Wisanwanichthan, M. Thammawichai, A double-layered hybrid approach for network intrusion detection system using combined naive bayes and SVM, *IEEE Access* 9 (2021) 138432–138450, <https://doi.org/10.1109/ACCESS.2021.3118573>.
- [11] M. C. Belavagi and B. Muniyal, "Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection," in *Procedia Computer Science*, Elsevier B.V., 2016, pp. 117–123. doi: 10.1016/j.procs.2016.06.016.
- [12] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues ☆," vol. 189, p. 105124, 2020, doi: 10.1016/j.knosys.
- [13] N. Shone, T.N. Ngoc, V.D. Phai, Q. Shi, A deep learning approach to network intrusion detection, *IEEE Trans Emerg Top Comput Intell* 2 (1) (2018) 41–50, <https://doi.org/10.1109/TETCI.2017.2772792>.

- [14] Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, May 2018, doi: 10.1109/ACCESS.2018.2836950.
- [15] G. Nguyen, et al., Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey, *Artif Intell Rev* 52 (1) (2019) 77–124, <https://doi.org/10.1007/s10462-018-09679-z>.
- [16] C. Yin, Y. Zhu, J. Fei, X. He, A deep learning approach for intrusion detection using recurrent neural networks, *IEEE Access* 5 (2017) 21954–21961, <https://doi.org/10.1109/ACCESS.2017.2762418>.
- [17] S.M. Sohi, J.P. Seifert, F. Ganji, RNNIDS: enhancing network intrusion detection systems through deep learning, *Comput Secur* 102 (2021), <https://doi.org/10.1016/j.cose.2020.102151>.
- [18] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [19] M. Rezapour, S. Nazneen, K. Ksaibati, Application of deep learning techniques in predicting motorcycle crash severity, *Eng. Rep.* 2 (7) (2020), <https://doi.org/10.1002/eng2.12175>.
- [20] K. Bandara, C. Bergmeir, S. Smyl, Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach, *Expert Syst Appl* 140 (2020), <https://doi.org/10.1016/j.eswa.2019.112896>.
- [21] W. De Mulder, S. Bethard, and M. F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling," *Computer Speech and Language*, vol. 30, no. 1. Academic Press, pp. 61–98, 2015. doi: 10.1016/j.csl.2014.09.005.
- [22] S. Hochreiter, J. Schmidhuber, Long Short-Term memory, *Neural Comput* 9 (8) (1997) 1735–1780, <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [23] Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: Lstm cells and network architectures, *Neural Computa* 31 (7) (2019) 1235–1270, <https://doi.org/10.1162/neco.a.01199>.
- [24] Y. Imrana, Y. Xiang, L. Ali, Z. Abdul-Rauf, A bidirectional LSTM deep learning approach for intrusion detection, *Expert Syst Appl* 185 (2021), <https://doi.org/10.1016/j.eswa.2021.115524>.
- [25] S. Naseer, et al., Enhanced network anomaly detection based on deep neural networks, *IEEE Access* 6 (2018) 48231–48246, <https://doi.org/10.1109/ACCESS.2018.2863036>.
- [26] S.M. Kasongo, Y. Sun, A deep learning method with filter-based feature engineering for wireless intrusion detection system, *IEEE Access* 7 (2019) 38597–38607, <https://doi.org/10.1109/ACCESS.2019.2905633>.
- [27] T.T.H. Le, Y. Kim, H. Kim, Network intrusion detection based on a novel feature selection model and various recurrent neural networks, *Appl. Sci.* 9 (7) (2019) 1392, <https://doi.org/10.3390/app9071392>.
- [28] C. Xu, J. Shen, X. Du, F. Zhang, An intrusion detection system using a deep neural network with gated recurrent units, *IEEE Access* 6 (2018) 48697–48707, <https://doi.org/10.1109/ACCESS.2018.2867564>.
- [29] C. Liu, Y. Liu, Y. Yan, J. Wang, An intrusion detection model with hierarchical attention mechanism, *IEEE Access* 8 (2020) 67542–67554, <https://doi.org/10.1109/ACCESS.2020.2983568>.
- [30] S.U. Rehman, M. Khaliq, S.I. Imtiaz, A. Rasool, M. Shafiq, A.R. Javed, Z. Jalil, A. K. Bashir, Diddos: an approach for detection and identification of Distributed Denial of Service (DDoS) cyberattacks using gated recurrent units (GRU), *Future Gener. Comput. Syst.* 118 (2021) 453–466, <https://doi.org/10.1016/j.future.2021.01.022>.
- [31] N.B. Singh, M.M. Singh, A. Sarkar, J.K. Mandal, A novel wide & deep transfer learning stacked GRU framework for network intrusion detection, *Journal of Information Security and Applications* 61 (2021), <https://doi.org/10.1016/j.jisa.2021.102899>.
- [32] R. Ben Said, Z. Sabir, I. Askerzade, CNN-BiLSTM: a hybrid deep learning approach for network intrusion detection system in software-defined networking with hybrid feature selection, *IEEE Access* 11 (2023) 138732–138747, <https://doi.org/10.1109/ACCESS.2023.3340142>.
- [33] J. Kim, J. Kim, H. Kim, M. Shim, E. Choi, CNN-based network intrusion detection against denial-of-service attacks, *Electronics (Switzerland)* 9 (6) (2020) 1–21, <https://doi.org/10.3390/electronics9060916>.
- [34] K. Jiang, W. Wang, A. Wang, H. Wu, Network intrusion detection combined hybrid sampling with deep hierarchical network, *IEEE Access* 8 (2020) 32464–32476, <https://doi.org/10.1109/ACCESS.2020.2973730>.
- [35] J.A. Perez-Diaz, I.A. Valdovinos, K.K.R. Choo, D. Zhu, A flexible SDN-based architecture for identifying and mitigating Low-Rate DDoS attacks using machine learning, *IEEE Access* 8 (2020) 155859–155872, <https://doi.org/10.1109/ACCESS.2020.3019330>.
- [36] C. Xu, J. Shen, X. Du, Low-rate DoS attack detection method based on hybrid deep neural networks, *Journal of Information Security and Applications* 60 (2021), <https://doi.org/10.1016/j.jisa.2021.102879>.
- [37] M.S. ElSayed, N.A. Le-Khac, M.A. Albahar, A. Jurcut, A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique, *J. Netw. Comput. Appl.* 191 (2021), <https://doi.org/10.1016/j.jnca.2021.103160>.
- [38] M.V.O. Assis, L.F. Carvalho, J. Lloret, M.L. Proença, A GRU deep learning system against attacks in software-defined networks, *J. Netw. Comput. Appl.* 177 (2021), <https://doi.org/10.1016/j.jnca.2020.102942>.
- [39] H.H. Jazi, H. Gonzalez, N. Stakhanova, A.A. Ghorbani, Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling, *Comput. Netw.* 121 (Jul. 2017) 25–36, <https://doi.org/10.1016/j.comnet.2017.03.018>.
- [40] A. A. Adeyemo and K. A. Ganiyu, "A STUDY OF DENIAL OF SERVICE ATTACK WITH ITS TOOLS AND POSSIBLE MITIGATION TECHNIQUES," 2019.
- [41] I. PES Institute of Technology (Bangalore), IEEE Communications Society, IEEE Photonics Society. Bangalore Chapter, IEEE Robotics and Automation Society. Bangalore Chapter, and Institute of Electrical and Electronics Engineers, 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) : 19-22 Sept. 2018.
- [42] T. Apostolovic, N. Stankovic, K. Milenkovic and Z. Stanisavljevic, "DDoSSim - System for Visual Representation of the Selected Distributed Denial of Service Attacks," 2018 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2018, pp. 118-122, doi: 10.1109/ZINC.2018.8448570.
- [43] M. M. Najafabadi, T. M. Khoshgoftaar, A. Napolitano, and C. Wheelus, "RUDY Attack: Detection at the Network Level and Its Important Features." [Online]. Available: <http://www.gartner.com/newsroom/id/2344217>.
- [44] O. Yevsieieva and S. M. Helalat, "Analysis of the impact of the slow HTTP DOS and DDOS attacks on the cloud environment," in 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), IEEE, Oct. 2017, pp. 519–523. doi: 10.1109/INFOCOMMST.2017.8246453.
- [45] Institute of Electrical and Electronics Engineers and IEEE Communications Society, Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) : 22-24 March 2017, Chennai, India.
- [46] I. Sharafaldin, A. Habibi Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization. *International Conference on Information Systems Security and Privacy*, 2018.
- [47] Ahmed, A. (2024). *LR-HR DDoS-2024 Dataset for SDN-Based Networks* [Data set]. GitHub. <https://github.com/Abdussalaam/LR-HR-DDoS-2024-Dataset-for-SDN-Based-Networks>.