RANDOM FOREST AND MEMORY MODELS IN INTRUSION DETECTION

[1]Senthilkumar SP, [2]Suresh Kumar B*

[1]Assistant Professor, Department of Computer and Information Science, Annamalai University, Tamil Nadu, India,
[2]Assistant Professor, Department of Computer Science, Government Arts and Science College, Sriperumbudur, Chennai, Tamil Nadu, India.
*Corresponding Author's Name: Suresh Kumar B and Email: sureshaucis@gmail.com

ABSTRACT

With cyberattacks growing more complex and harder to predict, defending digital networks is becoming a tougher challenge than ever. This study looks at two ways to improve Intrusion Detection Systems (IDS): Random Forest (RF), a popular machine learning method known for being fast and straightforward, and Bidirectional Long Short-Term Memory (Bi-LSTM), a deep learning model designed to analyze sequences of data over time and pick up on subtle patterns. We worked with the CSE-CIC-IDS2018 dataset, carefully prepared to reflect real network traffic, to see how these models perform. Random Forest delivered solid and steady results with 96.8% accuracy, making it a quick and lightweight option perfect for setups where resources are limited. However, it had trouble spotting the more sophisticated, subtle attacks. Bi-LSTM on the other hand did even better by detecting complex intrusions with 98.02% accuracy and strong precision, recall, and F1-scores all close to 97%. Because it processes data both forwards and backwards, it gains a richer understanding of changing threat patterns. What this shows is that while Random Forest works well when speed and efficiency matter most, Bi-LSTM offers the flexibility and strength needed to keep up with evolving cyber threats. Choosing the right IDS means looking beyond just numbers. It is about finding what fits the shifting landscape of cybersecurity. By pairing careful data preparation with advanced AI techniques, this work helps pave the way toward smarter, safer digital networks.

***Keywords*:** Intrusion Detection System (IDS), Bi-LSTM, Random Forest, Network Traffic, CSE-CIC-IDS2018, Cybersecurity.

INTRODUCTION

The growing frequency and sophistication of threats have raised the requirement for more advanced cyber security solutions for the protection of digital infrastructure. Network Intrusion Detection System (NIDS) is essential in the detection as well as the counteraction of harmful activity in the network environment. Though the traditional machine learning (ML) solutions miss advanced threats as they cannot capture the temporality as well as deal with the imbalance in the dataset, the new developments in the area of deep learning (DL) have provided more effective solutions. RF as well as Bidirectional Long Short-Term Memory

(Bi-LSTM) models have been found as effective models for the enhancement of the accuracy of the intrusion detection. RF is highly interpretable with the added benefit of being resilient through the use of the ensemble approach, whereas the Bi- LSTM is proficient in the detection of sequential dependencies in the network traffic, thus being highly effective in the detection of complex as well as dynamic attack patterns. This study is a strict comparative analysis of RF and Bi-LSTM models using the CSE CIC IDS2018 dataset, a widely used benchmark for intrusion detection system evaluation. It is cleaned with strict data cleaning, class balancing, and feature transformation to maximize model accuracy as well as provide equal evaluation. The primary objective of the research is measuring the models for identifying diverse types of network intrusions. It employs conventional metrics such as accuracy, precision, recall, F1 score, and AUC ROC scores for evaluation. It indicates how Bi- LSTM is superior to RF in identifying the dynamic patterns of traffic in the network. It displays the potential of the application of deep learning models in the quest for increased precision, adaptability, as well as general resistance in the utilization of modern-day intrusion detection systems. The evolution of Network Intrusion Detection Systems (NIDS) has increasingly employed machine learning (ML) and deep learning (DL) methods in the fight against sophisticated cybersecurity threats. Researchers have had much success in recent years with applying artificial intelligence to secure computer networks. One of the most successful strategies has been the application of models that learn from a trend over the course of many months. Long Short Term Memory networks, as the models are called, are particularly effective at identifying threats that gradually build over time instead of occurring all of a sudden. To make them perform even better, other researchers have blended different types of models. Called ensemble techniques, the hybrids have been effective in increasing accuracy and yielding more consistent detection results (1). Despite all of this, though, conventional machine learning techniques are still wanting. They are not generally well-suited to handle the large amount of information which runs through contemporary networks, nor are they well-suited to identify the most critical patterns in the information (2). To avoid this problem, others have chosen to utilize more sophisticated techniques. As a case in point, one article proposed a method which preprocessed the information prior to analysis. Through the use of a new type of learning model which compresses information, and then utilizing the widely known classifiers Random Forest and Gaussian models, they could successfully attain much enhanced detection results with a commonly used dataset (3). Another group of researchers developed a network-based model that had an almost one-hundred percent accuracy level with false alarms kept at an extremely low percentage level. The model was

put in competition for the detection of botnet attacks in actual scenarios (4). More research utilized conventional models such as Logistic Regression, Decision Trees, and Random Forest successfully to differentiate between different types of attacks (5). Methods that are obtained from labeled sets of data are still applicable in the detection of general and targeted network threats (6). The issue of dealing with stale or biased data is still there, though. Some authors have established that most sets of data employed nowadays do not represent the numerous types of threats seen in current network landscapes (7). Some reviews in this area have proposed the employment of technologies such as convolutional networks and support vector machines. The reviews also propose questions of relevance concerning the performance of such techniques when used in a live setting (8,9). To enhance the performance of models, there are new techniques used to fine tune them. One group of studies employed nature-inspired techniques like the artificial bee approach and the path finding abilities of ants. They applied the techniques to fine-tune the models trained with the CSE CIC IDS2018 dataset and the improvements were visible (10). Another project employed a two step model with gradient boosting and an image-based network and achieved near perfect results across all measures (11). There was also a model constructed with artificial neural networks that filtered the most appropriate attributes and obtained nearly perfect outputs across all categories such as accuracy and precision (12). Various innovative strategies have continued to be developed. One such, which was called VAEMax, integrated two strong tools that identify known and unknown types of strikes. It worked particularly well even for newly emerging types of vicious traffic (13). Another group of researchers constructed a system that employed multiple intelligent subsystems together. The model learned from the past and had the capability of adapting with the emergence of new patterns, performing extremely well (14). Meanwhile, however, other models that performs well in one testing condition struggle when transferred to a different dataset. This problem has led to the need for developing the kind of models that can learn under many conditions (15). More recent models using new memory units have gone even further by achieving nearly perfect performance when tested against trusted datasets (16). A particular such model integrated the learning of images and sequences in an innovative manner and was robust even when faced with new and unforeseen threats (17). These technologies are not limited to cybersecurity at all. For example, in transportation, a vision based model identified potholes in roads. It worked quickly and with consistent accuracy, forming a useful resource for road maintenance (18). In finance, the studies investigated the way people in the countryside adopt new technology. They used the application of behaviour based models to determine what enables

or prevents that process (19). In industry, there have been groups working towards discovering how to combine digital simulations with development tools, as a means for developing systems which are not only more agile and robust under stress (20). Researchers in medical studies used pattern drawing to analyze precursors to nerve disorders. Learning algorithms enabled them to correctly ascertain indicators of illness from a basic drawing (21). In the educational sphere, predictive models of a computer type were created so as to make predictions as to a particular student's performance. The models analyzed the pattern of students' actions and made extremely accurate predictions (22). As for the field of information security, a new technology was developed which embeds concealed messages in photographs. It is invisible to the naked eye, with the quality of the photograph not affected, so it proves an efficient way to conceal information (23).

DATASET PREPARATION

Dataset consolidation

One of the most critical obstacles to creating effective threat detection systems for computer networks is the quality of the training data. Frequently used public datasets all have serious flaws. They tend to have uneven distributions of types of attacks and ordinary traffic, and much of the data is stale. All of this can hinder the creation of well-performing models in realistic settings and hinder the advance of cybersecurity research (24). In order to resolve such concerns, a revised version of the CSE CIC IDS2018 data was made available as of the fifth of February of the year 2024. The revised version, which is Version 1, was made to serve as a robust and realistic basis for training and testing intrusion detection models. The data can be made available as a compressed file that is about seven hundred and five megabytes in size. It comprises eleven different files, each with organized records of network actions. Since it encompasses a large network of actions in a uniform structure, the data is an integral constituent to further the development in the fields of intrusion detection and cybersecurity (25).The preprocessing stage was initiated with organized extraction and compilation to maintain data integrity, uniformity, and compatibility with machine learning tools. The raw data, which were stored in a compressed format, were extracted in an organized form and stored within a directory. Various CSV files, each pertaining to various network traffic scenarios, were compiled into a single data set, with easier handling of data and lesser inconsistencies.

Challenges before preprocessing

Prior to preprocessing, the data set contained several problems like missing values, duplicated records, inappropriate data, imbalanced classes, and raw categorical features. Inconsistencies can adversely affect the performance of machine learning. The dataset now had 9,625,148 instances, including potential noise and redundant records, requiring extensive cleaning to ensure reliability and enable accurate analysis.

Data cleaning and deduplication

An extensive cleaning process was conducted to eliminate redundancies and inconsistencies. Duplicate cases were removed to prevent data leakage and artificial repetition. Missing values were handled using imputation or row removal where necessary. Irrelevant or corrupted cases were filtered out to maintain dataset integrity. The dataset was cleaned down to 5,183,021 cases after preprocessing, ensuring higher quality data for accurate analysis and model training.

Table 1: Initial data distribution of the CSE-CIC-IDS2018 dataset and data distribution after cleaning and deduplication

Attack Type	Initial Data Distribution - Sample Count	Data Distribution after Cleaning and Deduplication
Benign	6876913	3830384
DDoS attack-HOIC	686012	575364
DDoS attacks-LOIC-HTTP	576191	198861
DoS attacks-Hulk	461912	145199
Bot	286191	144535
FTP-BruteForce	193360	140610
SSH-Bruteforce	187589	94048
Infiltration	161934	41406
DoS attacks-SlowHTTPTest	139890	9908
DoS attacks-GoldenEye	41508	1730
DoS attacks-Slowloris	10990	555
DDoS attack-LOIC-UDP	1730	228
Brute Force -Web	611	84
Brute Force -XSS	230	55
SQL Injection	87	54
Total	9625148	51,83,021

The table above is the CSE-CIC-IDS2018 dataset split with 9,625,148 instances of benign and malicious traffic. With attacks including DDoS, DoS, brute force, botnet, and infiltration, it's a primary source for intrusion detection system testing and enhancement. The table also shows 5.18 million network traffic samples after cleaning, mostly benign (3.83M), with DDoS-LOIC-HTTP as the top attack.

Handling class imbalance

Ensuring datasets are balanced is paramount for minimizing bias in machine learning classifiers and predictors. Though commonly used datasets such as CSE CIC IDS 2017 and 2018 exist, many studies have not properly worked towards solving the problem of

imbalanced class distribution (26). Some studies have implemented resampling techniques such as the use of adaptive synthetic sampling, synthetic minority oversampling, and edited nearest neighbors, in combination with rescaled class weights, for enhancing detection of underrepresented threats (27). Some have proved that if particular classes overwhelm the training set, the performance of deep learning networks can be seriously impacted (28). It is also noted that no single remedy exists for the issue of class imbalance since every dataset is different (29). For better model generalizability and interpretability, attack categories of similar type were generalized into broad categories. Class imbalance was addressed using two-stage resampling. Initially, Random Under-Sampling (RUS) was used to keep the common classes at 100,000 instances per class. Secondly, the Synthetic Minority Over-Sampling Technique (SMOTE) was utilized to augment the minority class samples without repetition by generating synthesized samples. For instance, the Brute Force Attack category of just 837 samples was oversampled to 10,000 without overfitting but with enhanced representation. Balanced data helped the model to better learn general and uncommon attack patterns. Last but not least, label encoding was used to map attack categories to numeric labels in order to have the dataset prepared for machine learning algorithms. This method has also been recognized as valuable for improving the performance of models trained through supervised learning (30).

Table 2: Data Distribution after Random Under-Sampling (RUS)

Category	Grouped Category Counts	After RUS Count	After SMOTE Count	Numeric Label
NORMAL	3830384	100000	100000	0
DoS/DDoS Attack	972523	100000	100000	1
Botnet Activity	144535	100000	100000	2
Brute Force Attack	837	837	10000	3
Infiltration and Exploits	140694	100000	100000	4
SSH-Brute Force	94048	94048	100000	5

Table 2 shows the balanced dataset after applying RUS and SMOTE, ensuring fair representation across all attack categories.

METHODOLOGY

Model architecture and evaluation for intrusion detection

In choosing the models for intrusion detection, two major methods are investigated: Machine Learning (ML) and Deep Learning (DL). Under the ML method, the Random Forest (RF) algorithm, a strong and explainable algorithm, is utilized based on its ability to deal with structured data. In the DL method, Bidirectional Long Short-Term Memory (Bi-LSTM)

network is utilized because it can model sophisticated, non linear relations and sequential dependencies in network traffic effectively. RF and Bi-LSTM models are discussed in detail in the following sections. These two models are taken through a rigorous training and testing phase using supervised learning techniques. . Both models are trained independently on the CSE-CIC-IDS 2018 dataset and tested using a supervised learning framework. Once training is complete, predictions from each model are assessed based on principal metrics like accuracy, precision, recall, and F1-score. The output is evaluated based on a confusion matrix, finally labeling network traffic as normal or an attack. This method provides a thorough comparison between classical ML and deep learning methods for intrusion detection.

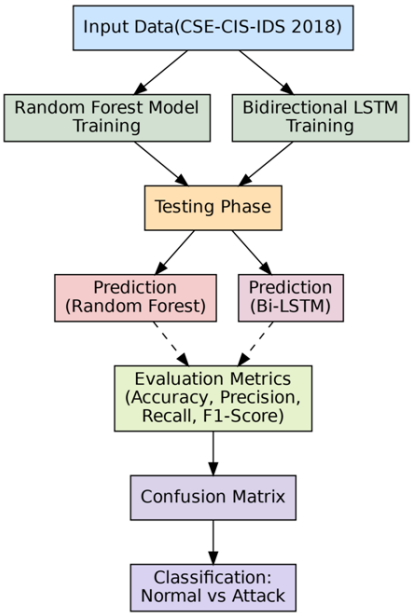

Figure 1: Framework of the Experimental Methodology

The diagram depicts the experimental process for intrusion detection with Random Forest (RF) and Bidirectional LSTM (Bi-LSTM) models

Random forest model

Structure and working

Random Forest is a method that combines multiple decision trees and uses randomness to improve accuracy while reducing the risk of overfitting (31). The RF model constructs multiple decision trees using random subsets of data and features, and aggregates their predictions through majority voting to ensure generalization and robustness. Each tree is independently trained on bootstrapped samples using random feature selection, increasing diversity and reducing the risk of overfitting. This ensemble approach ensures high performance even in large and complex datasets.

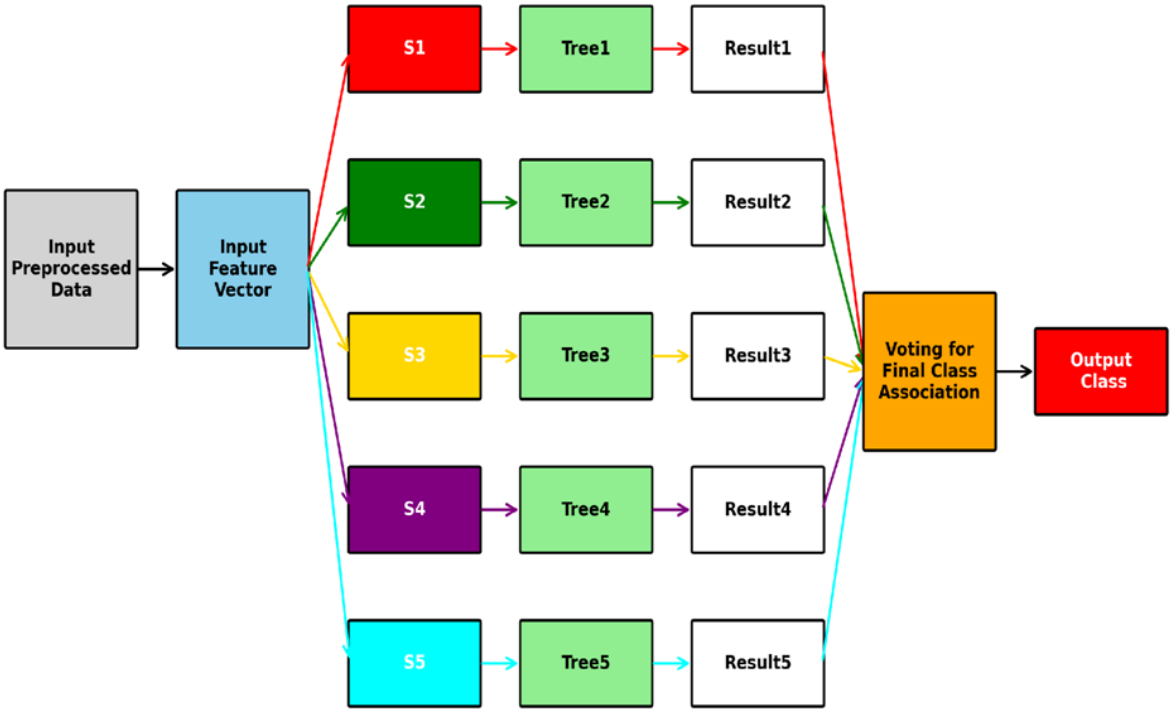

Figure 2: Architecture of Random Forest

The Random Forest model improves accuracy by combining multiple decision trees, each trained on random data subsets. Final classification is decided through majority voting, ensuring robustness and reducing overfitting.

Random forest- mathematical notation

Given a Dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$:

- $x_i \in R^m$ is a feature vector with m features
- $y_i \in \{1,2, \ldots \ldots K\}$ is the class label for K possible classes.

Ensemble model formation:

A Random Forest builds an ensemble of Ndecision trees $\{T_1, T_{2,\ldots\ldots}, T_{N,}\}$. . Each tree is trained on a bootstrap sample$D_i \subset D$, selected randomly with replacement.

Prediction function: For an inputx,, each decision tree T_i produces a prediction $h_i(x)$. The final prediction $H(x)$is obtained using majority voting:

$H(x) = \arg\max_k \sum_{i=1}^{n} \prod [h_i(x) = k]$ (1)

Where$\prod(.)$ is the indicator function (1 if the condition holds, 0 otherwise)

Splitting criterion:

At each node, a random subset of features $F \subseteq \{1,2,\ldots.,m\}$ is selected. The best feature fs chosen base on the Gini Index :

$$G = 1 - \sum_{k=1}^{k} p_k^2 \qquad (2)$$

Where $p_k$is the proportion of class k at the node.

Evaluation metrics:

Confusion Matrix: $C = [C_{ij}]$ where C_{ij} is the number of instances with true label i and predicted label j

$$\text{Precision:} P_k = \frac{TP_k}{TP_k + FP_k} \qquad (3)$$

$$\text{Recall} : R_k = \frac{TP_k}{TP_k + FN_k} \qquad (4)$$

$$\text{F1 Score} : F1_k = \frac{2.P_k.R_k}{P_k + R_k} \qquad (5)$$

Where TP_k, FP_k, and TP_k, represent true positives, false positives and fasle negatives, for class k.

Overall accuracy:

$$Accuracy = \frac{1}{n} \sum_{i=1}^{n} \| [H(x_i) = y_i \qquad (6)$$

Where $H(x_i)$ is the predicted class for instance i

Training and evaluation

Random Forest has been shown to deliver consistently high accuracy and strong F1 scores when applied to the CSE CIC IDS2018 dataset (32). The RF classifier was initialized with 200 estimators and a fixed random state of 42 for reproducibility. The classifier was trained on reshaped two-dimensional data, and its performance was evaluated with a classification report and a confusion matrix plotted as a heatmap.

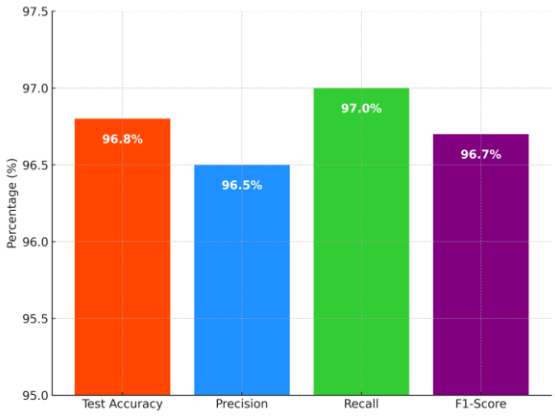

Figure 3: Random Forest - Metrics Comparison

The Random Forest classifier had 96.8% accuracy, with impressive precision, recall, and F1-score. These values certify the model as reliable in the separation of normal traffic from attack traffic for effective detection of intrusions. A balanced dataset was trained on the Random Forest model, with 200 estimators and a static random state of 42. Evaluation was done on its performance, utilizing a classification report and confusion matrix to provide credible accuracy and F1 scores.

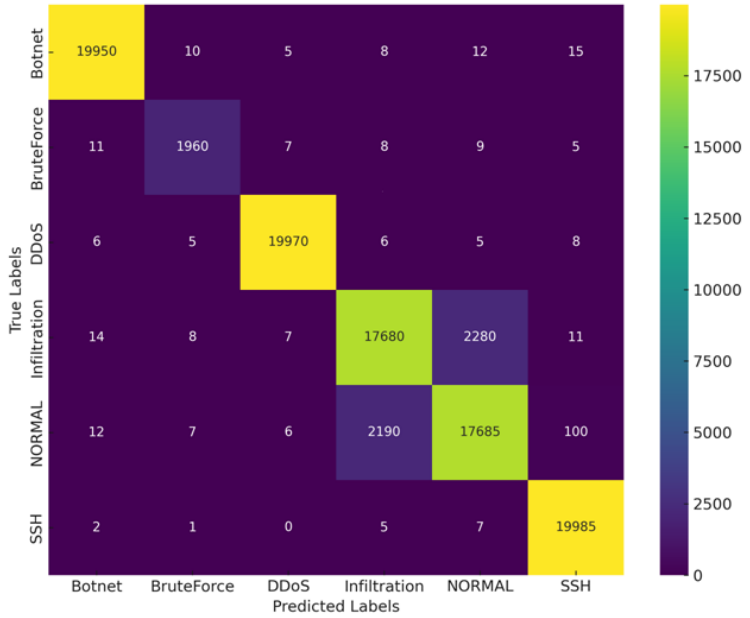

Figure 4: Random Forest - Confusion Matrix

The figure 4 shows how well the Random Forest model can classify, having high accuracy without much misclassifying. The classification report also corroborates the robust performance of Random Forest model, especially in identifying attack types with almost perfect precision, recall, and F1-scores. Though the model is performing outstandingly well when it comes to recognizing Botnet, DoS/DDoS, and SSH brute-force attacks, its precision and recall diminish somewhat for infiltration and normal traffic. Nevertheless, the overall high values confirm the robustness of the model in real-world intrusion detection. This means the model not only catches most of the threats accurately but also keeps false alarms to a minimum, which is crucial in practical security settings. While there's a little room for improvement with some traffic types, its strong overall performance makes it a reliable tool for protecting networks day-to-day.

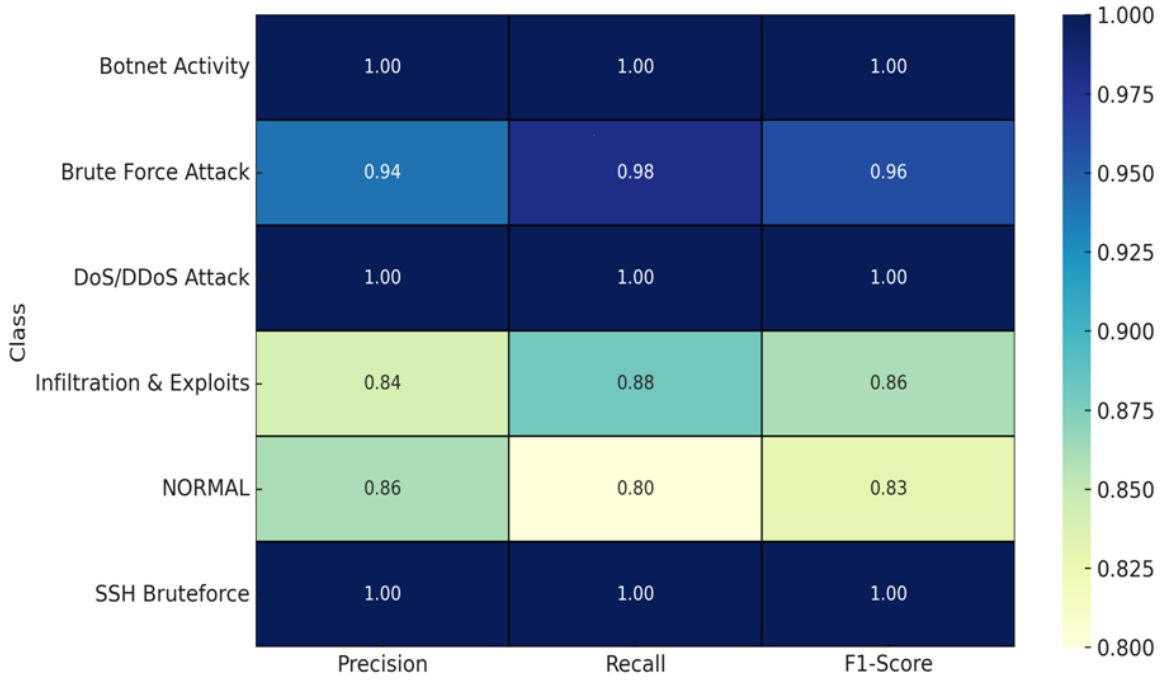

Figure 5: Random Forest - Classification Report

These results give us confidence that our Random Forest detector will be a dependable partner in day-to-day network defense, catching threats swiftly while sparing busy security teams from chasing down false alarms. In practice, this means smoother operations, quicker incident responses, and more time to focus on strategic improvements rather than firefighting routine alerts.

Bidirectional long short-term memory (Bi-LSTM)

Structure and working

While traditional LSTM models process data in only one direction, constraining their ability to capture full context, Bi- LSTM models process sequences in two directions. The two-way processing boosts the ability of the model to capture effectively long range dependencies. The Bi-LSTM architecture consists of two LSTM models operating in two opposing directions. Their outputs are concatenated and passed through dense layers and, eventually, through a softmax output layer for prediction. The model makes use of input, forget, and output gates to maintain long term dependencies, and is optimized using Adam using weighted categorical cross entropy loss function. Adaptive optimization algorithms like Adam and RMSprop have been found to outperform stochastic gradient descent (SGD) in certain deep learning scenarios (33).

Bi-lSTM – Mathematical notation:

The model consists of the following layers:

Bidirectional lstm layer:

For a time step t and input x_t

Forward Pass :

$$\underset{h_t}{\rightarrow} = \text{LSTM}\left(x_t\ ,\ \underset{h_{t-1}}{\longrightarrow}\right) \qquad (7)$$

Backward Pass :

$$\underset{h_t}{\Leftarrow} = \text{LSTM}\left(x_t,\ \underset{h_{t+1}}{\Longleftarrow}\right) \qquad (8)$$

Final Output:

$$h_t = \left[\underset{h_t}{\leftarrow}; \underset{h_t}{\rightarrow}\right] \qquad (9)$$

The concatenation of forward and backward hidden states h_t captures both past and future context for better sequence modeling.

Loss function (categorical cross-entropy)

The model uses categorical cross-entropy loss, defined as:

$$L = -\sum_{i=1}^{n}\sum_{j=1}^{k} y_{ij}\log\left(\hat{y}_{ij}\right) \qquad (10)$$

where:

y_{ij} is the ground-truth label (one-hot encoded).

$\hat{y}_{ij}$ is the predicted probability from the softmax output.

n is the number of samples.

k is the number of classes.

Adam optimizer weight Update

The Adam optimizer updates parameters using the following equations:

First moment estimate (mean of gradients):

$$m_t = ß_1 m_{t-1} + (1 - ß_1)\, g_t \qquad (11)$$

Second moment estimate (uncentered variance of gradients):

$$v_t = ß_2 v_{t-1} + (1 - ß_2)\, {g^2}_t \qquad (12)$$

Bias correction for the moment estimates

$$\widehat{m}_t = \frac{m_t}{1 - ß_1^t}\ ,\quad \hat{v}_t = \frac{v_t}{1 - ß_2^t} \qquad (13)$$

Parameter update step:

$$\Theta_{t+1} = \Theta_t - \frac{\propto}{\sqrt{v_t} + \in}\ \widehat{m_t} \qquad (14)$$

where:

g_t is the gradient of the loss with respect to Θ (weights),

α is the learning rate.

$ß_1$, $ß_2$ are the decay rates

$\in$ is a small constant for numerical stability

Reduce learning rate on plateau (lr scheduler)

The learning rate is adjusted dynamically based on validation loss. If validation loss does not improve for a given number of epochs p (patience) , the learning rate $\propto$ is reduced by a factor f :

Where :

$\propto'$ is the new learning rate.

f is a reduction factor (e.g 0.5 in your case),

p is the patience parameter

Softmax activation (final layer)

The output layer uses a softmax activation function:

$$\hat{y}_t = \frac{e^{z_j}}{\sum_{i=1}^{k} e^{z_i}} \qquad (15)$$

where: z_j is the output of the last dense layer for class j,

k is the total number of classes,

$\hat{y}_t$ is the predicted probability for class j.

Training and evaluation of model

Training and evaluation begin by splitting the normalized and preprocessed dataset to training and testing, and by performing stratified splitting to maintain class balance. The data is reshaped to meet LSTM's input requirements, and class weights are computed to cater to any imbalance. The bidirectional LSTM network is augmented by attention, multiple LSTM, dropout, and batch normalization, and is finalized by including a last dense layer for soft-max classification. The network is compiled using Adam optimizer and weighted categorical cross-entropy loss, and is augmented by a learning rate scheduler to adaptively alter the learning rate. The network is trained using a validation split, and is thereafter tested on testing data by computing loss and accuracy. Predictions on testing data, and subsequent testing using classification report and confusion matrix, and monitoring using plots of accuracy and loss.

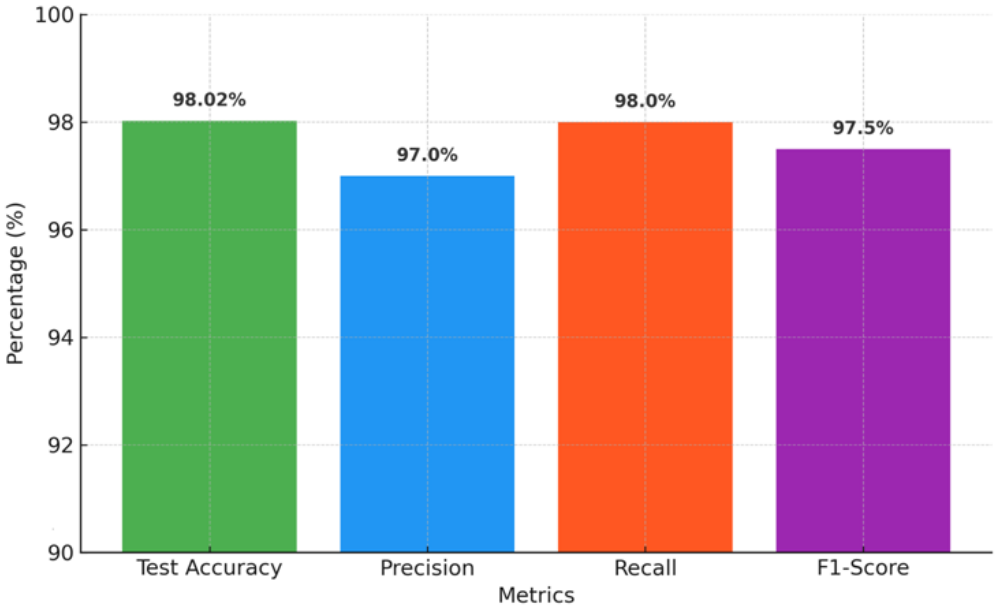

Figure 6: Bi-LSTM - Metrics Comparison

This is a graph comparing the performance measure of the Bi-LSTM model, which results in a whopping test accuracy of 98.02%. The model also registers high precision (97.0%), recall (98.0%), and an F1-score of 97.5%, demonstrating its robust capacity to identify intrusions correctly while sustaining an optimized performance across various measures.

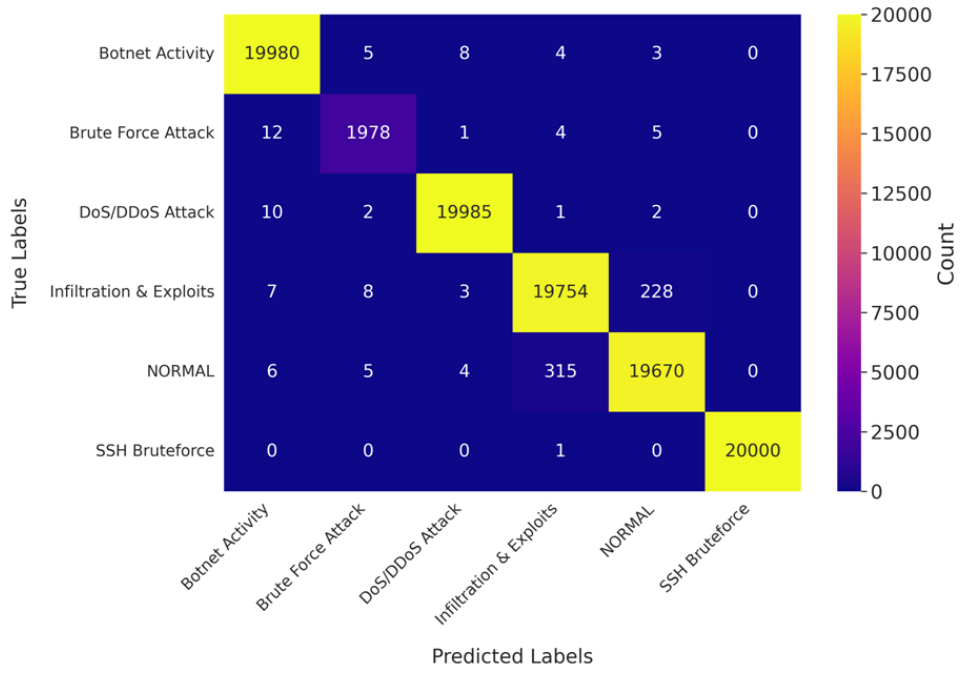

Figure 7: Bi-LSTM - Confusion Matrix

Figure 7 illustrates the Bi-LSTM confusion matrix, showcasing its high accuracy in detecting various intrusion types. The model correctly classifies most attacks with minimal errors.

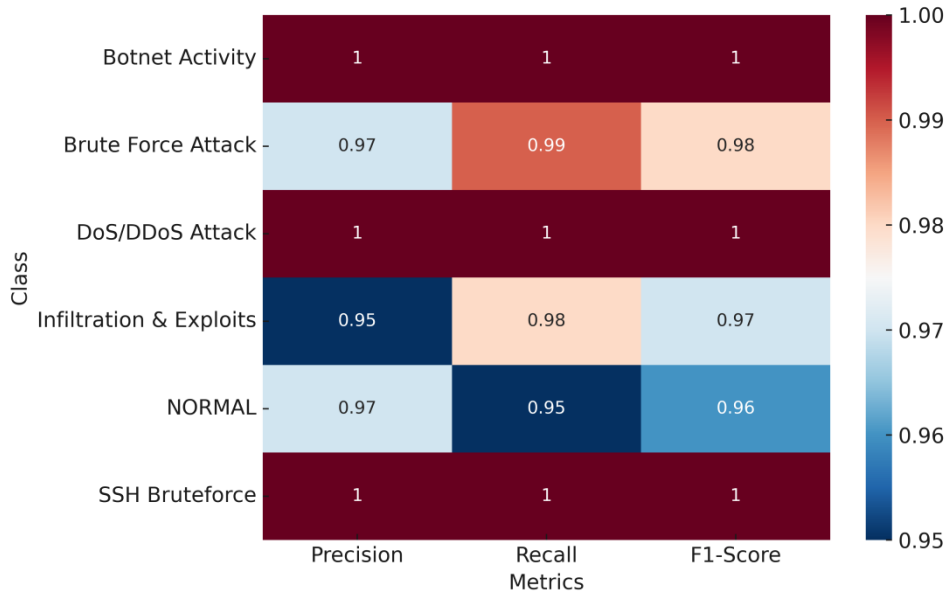

Fig. 8. Bi-LSTM – Classification Report

Figure 8 presents the Bi-LSTM classification report, showcasing impressive precision, recall, and F1-scores across various intrusion types. The model's outstanding performance in detecting each class highlights its robustness and reliability for real-world intrusion detection tasks.

RESULTS AND DISCUSSION

Comparative model performance overview

The results indicate Bi-LSTM to possess improved classification for all forms of cyber attacks. Registering a staggering 98.02 percent accuracy and a mere 0.0365 loss, Bi-LSTM surpasses RF in precision, recall, and F1 score in several attack types. The Area Under the Receiver Operating Characteristic Curve (AUC ROC) value efficiently measures the capacity of the models to discriminate between legitimate and malicious network traffic. The AUC ROC scores in various attack categories are encapsulated in the following table. Bi- LSTM demonstrates enhanced discrimination capabilities across all attack categories. It particularly excels in detecting complex attack patterns, such as Brute Force and Infiltration, where Random Forest exhibits comparatively lower classification effectiveness.

Table 3: Comparative Performance Metrics and AUC-ROC Scores of Random Forest and Bi-LSTM Models

Metric	Random Forest	Bi-LSTM
Test Accuracy	96.8%	98.02%

Precision	96.5%	97.0%
Recall	97.0%	98.0%
F1 Score	96.7%	97.5%
AUC-ROC (Botnet Activity)	0.997	0.999
AUC-ROC (Brute Force Attack)	0.972	0.985
AUC-ROC (DoS/DDoS Attack)	0.995	0.998
AUC-ROC (Infiltration Exploits)	0.961	0.976
AUC-ROC (NORMAL)	0.968	0.980
AUC-ROC (SSH Bruteforce)	1.000	1.000

The above table provides the extensive evaluation metrics for both models.

Evaluation using AUC -ROC scores

Figure 9 highlights Bi-LSTM's superior AUC-ROC scores across all attack categories, demonstrating its strong ability to distinguish between normal and malicious traffic. It outperforms Random Forest, especially in detecting complex attacks like Brute Force and Infiltration.

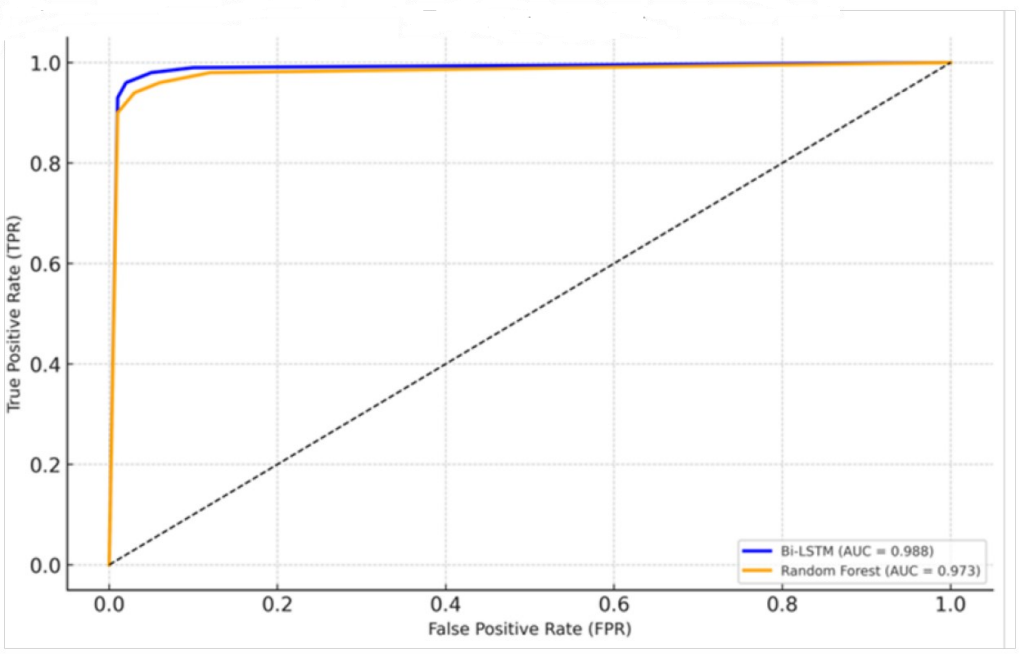

Figure 9: AUC-ROC Score Comparison of Bi-LSTM and Random Forest across Attack Categories

CONCLUSION

This research is compared with Bi-LSTM and RF in the discovery of intrusions over the CSE-CICIDS2018 dataset, whereby Bi-LSTM has higher values of accuracy, recall, and AUC-ROC. Bi-LSTM is capable of having 98.02% accuracy, effectively capturing network traffic's

sequential dependencies. RF is, however, also a good contender, having 96.8% accuracy, and is capable of having better computational cost and training time. Additionally, False Positive Rate (FPR) and training time pinpoint vital trade-offs. RF trains in 45 Seconds and is therefore ideal for use in real-time, while Bi-LSTM trains for 12 Minutes but is better in attack detection. Bi-LSTM also provides lower FPR (2.1%) compared to RF (3.5%), keeping false alarms to a minimum. These insights suggest how accurately detecting needs to be balanced against efficiency, depending on an Intrusion Detection System's (IDS) demands for implementation. Future research must focus on optimizing deep learning IDS for real time application while keeping computational cost low and maintaining high accuracy. This study offers insight into the comparative strengths of RF and Bi-LSTM for real-time detection, guiding future IDS implementations.

FUTURE WORK

Future research must aim at maximizing IDS performance in several critical aspects. Firstly, training Bi-LSTM models can be optimized through lean models such as Transformers, or through quantization and knowledge distillation methods to ensure performance while utilizing fewer resources. Secondly, FPR reduction can be obtained by integrating fast RF training with the capability of Bi-LSTM to learn deep features through hybrid models and anomaly detection to minimize false alarms. Real-time IDS optimization is also required, achievable through GPU acceleration or edge computing for faster inference, and streaming models for real-time detection of attacks without batch delay. Dataset diversification, especially with novel datasets other than CSE-CIC-IDS2018, will improve generalization to new cyber threats. Last but not least, enhancing interpretability with Explainable AI (XAI) and feature importance analysis will make it easier to trust and obtain better insights into attack patterns. Remedying these aspects will render future IDS models more efficient, dynamic, and effective in defense against real cyber threats.

Acknowledgement

Nil.

Funding

The current research has not received any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author Contributions

Dr. S.P. Senthilkumar was responsible for the conception, design, methodology implementation, and manuscript drafting. He also performed the experiments and coordinated the study.

Dr. B. Suresh Kumar contributed to data preprocessing, analysis, manuscript review, and editing. Both authors approved the final manuscript and are accountable for its content.

Ethics Approval

Not applicable.

Data Availability

The dataset used in this study (CSE-CIC-IDS2018) is publicly available at: https://www.unb.ca/cic/datasets/ids-2018.html

Abbreviations

RF: Random Forest, Bi-LSTM: Bidirectional Long Short-Term Memory, IDS: Intrusion Detection System, DL: Deep Learning, ML: Machine Learning, AUC-ROC: Area Under the Receiver Operating Characteristic Curve, FPR: False Positive Rate.

References

1. Sulaiman VS, Abdulazeez AM. A comparative analysis of intrusion detection systems: Leveraging classification algorithms and feature selection techniques. J Appl Sci Technol Trends. 2024;5(1):34–45.
2. Idrissi KH, Kartit A. Network intrusion detection using combined deep learning models: Literature survey and future research directions. IAENG Int J Comput Sci. 2024;51(8).
3. Elhanashi A, Gasmi K, Begni A, Dini P, Zheng Q, Saponara S. Machine learning techniques for anomaly-based detection system on CSE-CIC-IDS2018 dataset. In: Int Conf Appl Electron Pervading Ind Environ Soc. Springer Nature Switzerland; 2022.
4. Kanimozhi V, Jacob TP. Artificial intelligence-based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. In: 2019 Int Conf Commun Signal Process (ICCSP). IEEE; 2019. p. 33–6.
5. Göcs L, Johanyák ZC. Identifying relevant features of CSE-CIC-IDS2018 dataset for the development of an intrusion detection system. Intell Data Anal. 2024;28(6):1527–53.
6. Ibrahimi K, Jouhari M, Jakout Z. Enhancing intrusion detection systems using machine learning classifiers on the CSE-CIC-IDS2018 dataset. In: 2024 11th Int Conf Wirel Netw Mob Commun (WINCOM). IEEE; 2024. p. 1–6.
7. Momand A, Jan SU, Ramzan N. A systematic and comprehensive survey of recent advances in intrusion detection systems using machine learning: Deep learning, datasets, and attack taxonomy. J Sensors. 2023;2023(1):6048087.
8. Issa M, Aljanabi M, Muhialdeen H. Systematic literature review on intrusion detection systems: Research trends, algorithms, methods, datasets, and limitations. J Intell Syst. 2024;33(1):20230248.
9. Ogundokun RO, Basil U, Babatunde AN, Abdullahi AT, Adenike AR, Adebiyi AA. Intrusion detection systems based on machine learning approaches: A systematic review. In: 2023 Int Conf Sci Eng Bus Sustain Dev Goals (SEB-SDG). IEEE; 2023. p. 1–4.
10. Najafi Mohsenabad H, Tut MA. Optimizing cybersecurity attack detection in computer networks: A comparative analysis of bio-inspired optimization algorithms using the CSE-CIC-IDS 2018 dataset. Appl Sci. 2024;14(3):1044.
11. Zhang H, Zhang B, Huang L, Zhang Z, Huang H. An efficient two-stage network intrusion detection system in the Internet of Things. Information. 2023;14(2):77.

12. Khan M, Haroon M. Artificial neural network-based intrusion detection in cloud computing using CSE-CIC-IDS2018 datasets. In: 2023 3rd Asian Conf Innov Technol. IEEE; 2023.
13. Qiu Z, Zhou D, Zhai Y, Liu B, He L, Cao J. VAEMax: Open-set intrusion detection based on OpenMax and variational autoencoder. In: 2024 5th Inf Commun Technol Conf. IEEE; 2024. p. 98–105.
14. Soltani M, Khajavi K, Jafari Siavoshani M, Jahangir AH. A multi-agent adaptive deep learning framework for online intrusion detection. Cybersecurity. 2024;7(1):9.
15. Cantone M, Marrocco C, Bria A. On the cross-dataset generalization of machine learning for network intrusion detection. arXiv Preprint. 2024;arXiv:2402.10974.
16. Mahdi MS. Intrusion detection systems based on RNN and GRU models using CSE-CIC-IDS2018 dataset in AWS cloud. J Al-Qadisiyah Comput Sci Math. 2024;16(4):141–60.
17. Li L, Lu Y, Yang G, Yan X. End-to-end network intrusion detection based on contrastive learning. Sensors. 2024;24(7):2122.
18. Chandra S, Dubey K, Agarwal G, Chakraborty N. Pothole detection in drivable area using deep learning. Int J Comput Appl [online]. 2023;185(34):17–22.
19. Lavanya B, Rajkumar AD. Adoption of Digital Innovations in Rural Banking of Vellore District: Based on UTAUT Model. Int Res J Multidiscip Scope. 2024;5(1):263–71.
20. Saravanakumar R, Arularasan AN, Divakar Harekal, Praveen Kumar R, Kaliyamoorthi P, Pushpalatha KS, Vidhya RG. Integration of DevOps and Digital Twins for resilient Cyber-Physical Systems in Industry 4.0 and Industry 5.0. Int Res J Multidiscip Scope. 2024;5(3):571–82.
21. Afifi MT, Azab AM, Hassan KFA. Performance evaluation of machine learning and deep learning techniques for building effective intrusion detection system. SN Comput Sci. 2023;4(3):1–18.
22. Imandoust N, Farajollahi E, Sadjadi S. Anomaly detection in IoT networks using ensemble machine learning techniques based on the CICIDS2018 dataset. Procedia Comput Sci. 2023;213:458–68.
23. Aljamla N, Jawawdeh DN, Abdulrazzaq MAR. A comparative analysis of different deep learning algorithms for intelligent intrusion detection systems. In: 2023 IEEE 3rd Int Maghreb Meet Conf Sci Tech Autom Control Comput Eng (MI-STA). IEEE; 2023. p. 1–6.
24. Omotosho JA, Asani FA. Comparative analysis of recurrent neural network and long short-term memory for cyber intrusion detection. SN Comput Sci. 2023;4(4):1–12.
25. BirjandTavakol A, Sabeti MF, Sadeghian H, Hagh MS, Khanaliloo B. Comparison of different machine learning algorithms for detecting network intrusion using CSE-CIC-IDS2018 dataset. J Intell Fuzzy Syst. 2023;45(3):3541–51.
26. Abdulkareem SE, Alsadoon MA, Abed ST, Ali AMT. Feature selection and classification based IDS using a hybrid model of MVO and SVM. Indones J Electr Eng Comput Sci. 2023;32(1):78–87.
27. Chandran M, Balasubramanian S. A comparative study on deep learning based intrusion detection system. J Eng Res. 2023;11(1):92–101.
28. Senthilkumar MP, Balasubramanian SK. Enhanced intrusion detection using deep learning techniques. J Electr Eng Autom. 2023;5(1):32–9.
29. Kiani H, Fazel K. Network intrusion detection using ensemble learning on CIC-IDS2018 and CSE-CIC-IDS2018 datasets. Comput Sci Inf Syst. 2024;21(1):255–75.
30. Abubakar HM, Hayajneh T, Abdulsalaam R, Jamil IA. Lightweight anomaly-based network intrusion detection system using deep learning in cloud computing. Multimed Tools Appl. 2024;83:11713–35.
31. Ansari J, Khan R, Gulzar M, AlMajed A. Comparative analysis of classification algorithms for network intrusion detection using feature subset selection with CSE-CIC-IDS2018 dataset. Sustainability. 2024;16(3):1337.
32. Jamali A, Moosavi NS, Dehghan F. Network intrusion detection using optimized convolutional neural network
33. with dropout regularization technique. Multimed Tools Appl. 2024;83:9161–77.
34. Singh A, Sharma M, Deepak D. CIC-CNNIDS2017: Convolution neural network-based dataset for anomaly-based intrusion detection system. Multimed Tools Appl. 2024;83:10455–72.