Artificial Intelligence for Cybersecurity

Journal Homepage: sciencesforce.com/aics

Artificial Intell. Cyb. Vol. 1 (2024) 1–11

Paper Type: Original Article

Hybrid Deep Learning-Based Model for Intrusion Detection

Ahmed Tolba [1,*], Nihal N. Mostafa [2], and Karam Sallam [3]

[1] Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt; a.tolba24@fci.zu.edu.eg.

[2] Department of Computer Science,Misr higher institute for computer and commerce, Egypt; nihal.nabil@fci.zu.edu.eg.

[3] School of IT and Systems, University of Canberra, ACT 2601, Australia; karam.sallam@canberra.edu.au.

Received: 13 Sep 2023 **Revised:** 07 Dec 2023 **Accepted:** 06 Jan 2024 **Published:** 11 Jan 2024

Abstract

There is an intensive need for intrusion detection systems (IDSs) due to incremental and frequent cyber-attacks. The first line of defense against online threats is an IDS. Researchers are using deep learning (DL) approaches to detect attackers and preserve user information. In this study, we introduce a hybrid DL-based model. The proposed model integrates LSTM and ResNet to eliminate the vanishing gradient problem and increase the accuracy of the classification model. The proposed model aims to classify between normal or an attack, with each attack either being a DoS, U2R, R2L, or a probe over the NSL-KDD dataset. The proposed model achieves 99.5% according to accuracy. The model was compared with other ML and DL models.

Keywords: Intrusion Detection System; Deep Learning; LSTM; ResNet; Cyber-attacks.

1 | Introduction

Recently, ubiquitous computing increase dependency new technologies from individuals to big companies and government agencies. Which led to an increase in the flow of transactions and information on the Internet. Which poses a challenge to intrusion detection system (IDS). Malicious attacks are getting more complex and difficult to detect unknown and obfuscated malware, as malware coder's goal is to avoid reverse engineering or detection by an IDS [1].

An IDs is a system that analyzes network traffic for suspicious activity and sends notifications when it is detected. It is a software application that analyzes a network or system for potentially dangerous activities or policy violations. Any malicious activity or violation is often reported to an administrator or collected centrally via a security information and event management (SIEM) system. A SIEM system combines data from numerous sources and employs alert filtering techniques to distinguish harmful behavior from false alarms.

There are two types of IDS host-based IDS(HIDS) and network-based IDS (NIDS). A HIDS is placed on a specific endpoint to defend it from both internal and external threats. Such an IDS may be able to monitor network traffic to and from the machine, observe active processes, and examine the system's logs. A HIDS's visibility is limited to its host machine, reducing the accessible context for decision-making, but it has

Corresponding Author: a.tolba24@fci.zu.edu.eg

https://doi.org/10.61356/j.aics.2024.1198

extensive insight into the host computer's internals. The NIDS is intended to monitor the entire protected network. It sees all traffic going over the network and makes decisions based on packet metadata and contents. This broader perspective provides greater context and the ability to detect widespread threats; nevertheless, these systems do not have visibility into the internals of the endpoints they defend [2].

There are some techniques that can detect assaults, such as Signature-based intrusion detection systems (SIDS) [3]. SIDS are match algorithms that are used to detect past intrusions. In a nutshell, when an intrusion signature matches the signature of a previous intrusion that is already in the signature database, an alarm signal is generated. Díaz-Verdejo [4] investigated the performance of three SIDS in web attacks, focusing on detection rates and precision. Results show that the maximum detection rate achieved by SIDS is insufficient for effective protection and is lower than expected for known attacks. The choice of predefined settings on each detector significantly influences its detection capability and false alarm rate. The study suggests that open-source SIDS with default configurations should be considered for web attack protection. Unfortunately, Traditional IDS are knowledge- or signature-based systems that aren't suitable for the quickly expanding network and are unable of handle assaults because to their rising volume, complexity, and deflation [1].

Furthermore, the firewall stops illegal access to the complete network. It has been demonstrated that the firewall and its variations are easily evaded by attackers, for example by using a false source address. Furthermore, several attacks such as DoS and DDoS went unnoticed. This led to the importance of developing and obtain IDS [5].

IDS is an operation of examining a computer system or network's activity for identifying unusual events, malicious activity or policy violations. Bad malware, such as worms and spyware, internet hackers who gain remote control over networks, and privileged individuals who abuse their privileges for illegal objectives. The primary purpose of the IDS is to identify potential events. Furthermore, IDS may be able to detect times where an attacker has properly exploited a system flaw. Most IDS may be setup with a collection of firewall regulations, such as settings, to help detect network traffic that violates the organization's safety or legal usage requirements. IDSs with this capacity can monitor data transfers and identify anomalies, such as sending a huge data to a computers or servers [6].

Then, the advancement of IDS is driven by key facts such as:

- The complexity of new networked systems makes them vulnerable to faults that can be exploited by attackers.
- Existing network systems have some serious security flaws, making them a target for attackers. While there are methods and efforts to identify and address shortcomings, it is often impossible to eliminate them.
- While certain intrusion prevention technologies exist, 100% protection is not attainable. IDS has been shown to be an effective tool for detecting and identifying intrusions. then, the prevention mechanism can be automatically updated.Most preventative techniques protect the system from outsider attacks. Authorized users in the organization often carry out assaults, making detection difficult. These kinds of attacks can cause more harm.
- New attack types are created to avoid prevention and detection methods. To improve security solutions, a dynamic framework with deep learning is recommended.

Thus, in this paper we aim to maintain a hybrid Long short-term memory (LSTM) and Residual Networks (ResNet) on NSL-KDD dataset to classify between normal or an attack, with each attack either being a DoS, U2R, R2L, or a Probe. A comparison between bidirectional LSTM and CNN (CNN-BILSTM), LSTM, gated recurrent unit (GRU), and decision tree (DT) has been made to evaluate our proposed model.

As a result, the following are the paper's primary contributions:

- A new hybrid DL model LSTM and ResNet is proposed.

- The model obtains the input data in LSTM layer. Which followed by a residual block involves two LSTM layers in a sequential manner, with a skip connection added to the output of the first LSTM layer. Then followed again with LSTM layer. This is done to prevent over-fitting and eliminate vanishing gradients problem.
- Also, more layers of LSTM can improve the classification accuracy of the proposed model.
- The proposed model is compared with both DL and ML models.
- Preserving weights from over-stack within early layers . The remainder of the paper is divided as follows. Section 2 provides the background needed for this study. Section 3 presents the methodology of this study. Section 4 presents proposed model. Section 5 presents experimental results. Section 6 illustrates the conclusion and future directions of this proposal.

2 |Literature Review

Hnamte et al. [7] proposed a flexible IDS for detecting cyber-attacks DL model. A novel two-stage DL technique, hybridizing LSTM and Auto-Encoders (AE) was introduced. The CICIDS2017 and CSE-CICDIS2018 datasets are used to determine optimal network parameters. Experimental results show the hybrid model works well in modern scenarios.

Nguyen and Kashef [8] introduced TS-IDS, a traffic-aware self-supervised learning system for IoT network intrusion detection, with the goal of capturing flow interactions between network entities. Our strategy improves performance by utilizing both node and edge features. Furthermore, we use auxiliary property-based self-supervised learning (SSL) to improve graph representation even in the absence of labeled data. We carried out experiments on two real-world datasets: NF-ToN-IoT and NF-BoT-IoT. We compared the proposed model to cutting-edge baseline models to show how powerful our proposed framework is.

Du et al. [9] presented a DL network in intrusion detection classification model (NIDS-CNNLSTM) for the Industrial Internet of Things (IIoT) wireless sensing scenario. The model uses LSTM, CNN, and binary classification and multi-classification scenarios. Its performance improves significantly compared to previous models, showing high detection and classification accuracy, low false alarm rates, and is suitable for large-scale and multi-scenario network data.

In FOG-cloud environment, Binbusayyis [10] introduced a hybrid architecture for DL intrusion detection in fog computing, combining DL models and RBFSVR to minimize data dimensionality. The integrated VGG19 and 2DCNN are used to train the dataset, transferring it to the fog layer for threat recognition. The technique outperforms other methods in detection rate, F-score, precision, recall, FAR, and accuracy.

Another contribution investigated the role of adversarial attacks in enhancing the robustness of ML and DL-based NIDS. Four powerful attack techniques, including Fast Gradient Sign Method (FGSM), Jacobian Saliency Map Attack (JSMA), Projected Gradient Descent (PGD), and Carlini & Wagner (C&W), are implemented in NIDS. The study also explores the use of heuristics defense strategies like Adversarial Training (AT), Gaussian Data Augmentation (GDA), and High Confidence (HC) to enhance NIDS's resilience under adversarial attack scenarios. The study provides a comprehensive background for researchers interested in AML and its implementation in computer network security [11].

Another study introduced a hybrid sampling algorithm using ADASYN and RENN for sample processing, improving feature-discriminative ability. An enhanced reptile search algorithm (IRSA) is proposed, using a sine cosine algorithm and Levy flight to optimize model weight. The model was trained on CIC-IDS 2017, UNSW-NB15, and WSN-DS datasets for binary and multiclass classification. The model demonstrated high detection rate, good accuracy, and low false alarm rate, with K4 achieving an accuracy score of 81.99, precision-recall of 82.69, detection rate of 82.12, F1-score of 80.33, and FAR of 2.3 [12].

3 | Methodology

In this section, we provide some preliminaries of the hybrid DL model proposed model.

3.1 | Long-Short Term Memory (LSTM)

LSTM [13] is a specific type of RNN that can learn long-term dependencies. The standard LSTM contains three gates, which regulate information and forward it on to the next unit. The forget value either forgets all or does not forget the information, based on the value of the forget gate. The input gate has two halves that regulate the new information required to add the next cell state. The initial component of the input gate is the sigmoid layer, which regulates the output value recorded in the cell state. The Tanh layer is the input gate's second component, and it generates a vector of new feature values that are stored in the cell state. The output gates provide the most recent cell state information. The statistics execute selectively through the gates' structure and are passed through to update and hold the historical statistics, as well as update the cell state. LSTM considers previous historical values, evaluates current unknown patterns by changing itself based on the entire pattern set, and predicts future events. The LSTM architecture is shown in Figure 1.

$$\text{LSTM Formulation} = \begin{aligned} f_t &= \sigma(W_f x_l + U_f h_{l-1} + b_f), \\ \check{c}_l &= \tanh(W_c x_t + U_{\check{c}} h_{l-1} + b_c), \\ i_l &= \sigma(W_i x_l + U_i h_{l-1} + b_{\check{i}}), \\ c_l &= f_t \odot c_{l-1} + i_l \odot \check{c}_t, \\ o_t &= \sigma(W_o x_l + U_o h_{l-1} + b_o), \\ h_l &= o_l \odot \tanh(c_l), \end{aligned} \quad (1)$$

where: $\boldsymbol{x_t}$ is the input at time t step , $\odot$ is the element wise dot product , $\boldsymbol{i_l}$, $\boldsymbol{o_t}$, $\boldsymbol{f_t}$ is the input gate , output gate and forget gate respectively , $\boldsymbol{c_l}$ is the cell state, W,U,b model parameter.

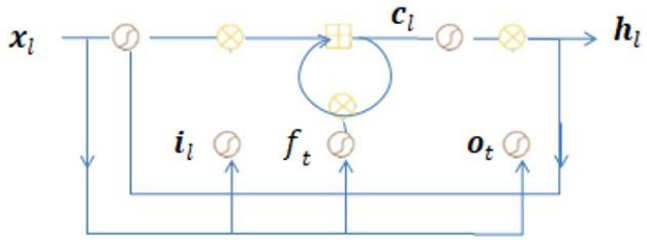

Figure 1. The LSTM architecture demonstration.

3.2 | Residual Network

ResNet [14] is a DL model which is designed to support hundreds or thousands of convolutional layers. It addresses the "vanishing gradient" problem by using "skip connections" to skip multiple layers and reuse activations from previous layers. This speeds up initial training and allows residual parts to explore more of the input image's feature space. Most ResNet models skip two or three layers at a time, with advanced models like HighwayNets learning "skip weights" dynamically.

ResNet architecture incorporates residual blocks, a key component. Older architectures like VGG16 used convolutional layers with batch normalization and nonlinear activation layers. However, increasing the number of layers can significantly improve CNN performance. ResNet architecture introduces an intermediate input to the output of convolution blocks. Residual block is shown in Figure 2.

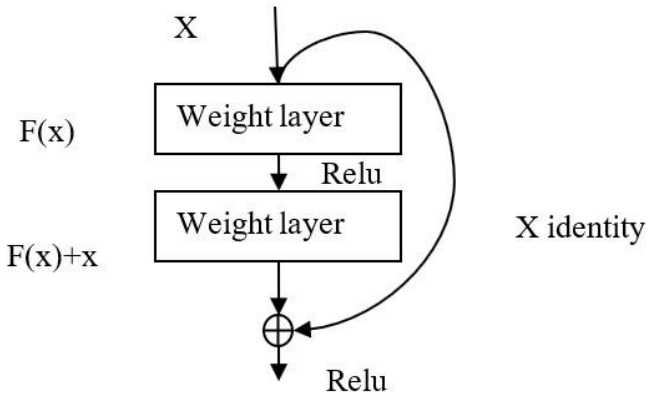

Figure 2. Residual network architecture.

4 | Proposed Model

Network security is an extremely important feature when exchanging data over the Internet. Cyber-attacks are becoming increasingly common in the IoT ecosystem due to reduced security measures. These existing models have many disadvantages, such as lower detection accuracy, lack of taxonomy, etc. This article introduces an innovative IDS model on the basis of a deep learning approach. The proposed model provides automated attack classification with high accuracy and fewer errors. LSTM is performed using residual blocks for attack detection. The systematic representation is shown in Figure 3.

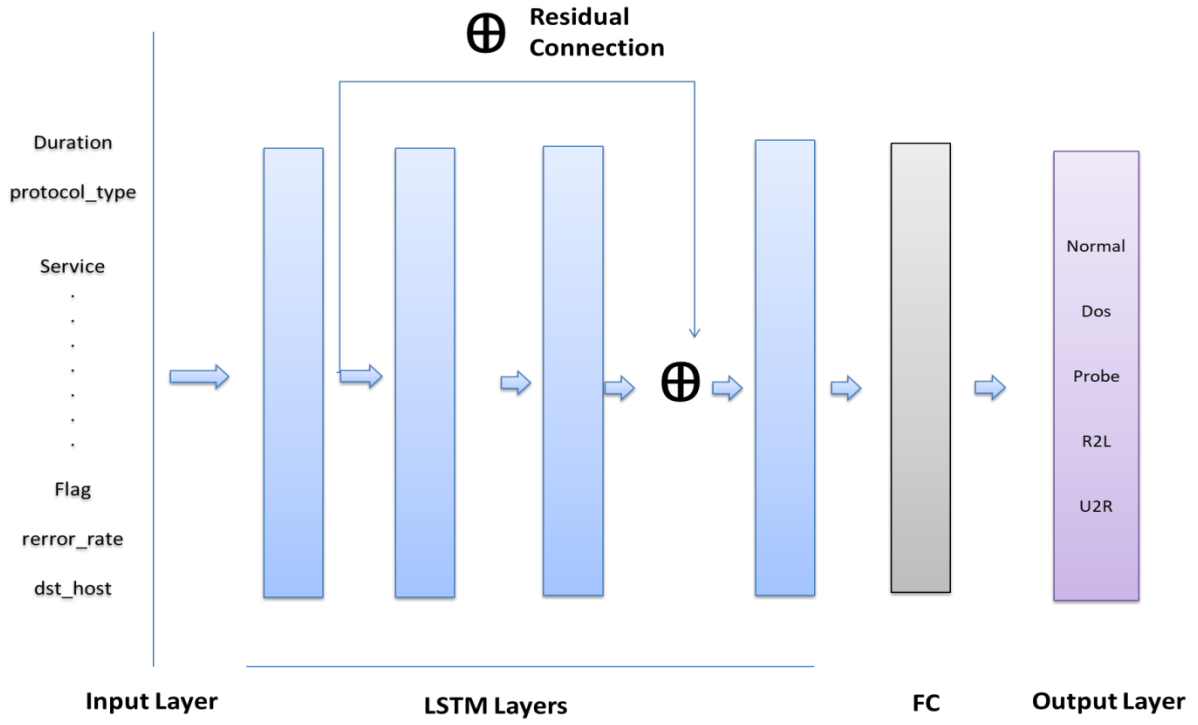

Figure 3. Proposed model architecture.

This proposed model is a hybrid of DL models which applies two LSTM layers in a sequential manner, with a skip connection added to the output of the first LSTM layer. This can avoid over-fitting and eliminate vanishing gradients problem. Additionally, it facilitates the training of deep networks by enabling the smooth flow of information through the network and preserving information from earlier layers.

The input data passed through a sequential process of LSTM layers with 150 unit, with the inclusion of a residual block in between. This is followed by the applying one LSTM layer, flattening, and dense layers to classify the data into five distinct classes. The softmax activation is employed for this classification.

Training Deep Learning Model

The model is built with parameters (criterion='gini', max_depth=3) . The proposed model compiled to determine the loss function, Adam optimizer with learning rate 0.01, and metrics for evaluating performance. The Categorical cross entropy loss function is used to optimize the initial weights of proposed model to increase classification accuracy. The loss function is mathematically defined as follows:

$$Minimize: loss = -\sum_{i=1}^{M} y_i \cdot log\, \breve{y}_i \quad (2)$$

Where y_i represent real values and $\breve{y}_i$ represent predicted values

Most DL models are trained with 30 epochs. In addition, in our experiments and applying mini-batch gradient descent technique to decreases the error calculated from the loss function (Categorical Cross Entropy). In each epoch, the data is divided into 158 batches so that the weights in each batch are updated. which means that in every epoch, the weights change 158 times, corresponding to the number of batches.

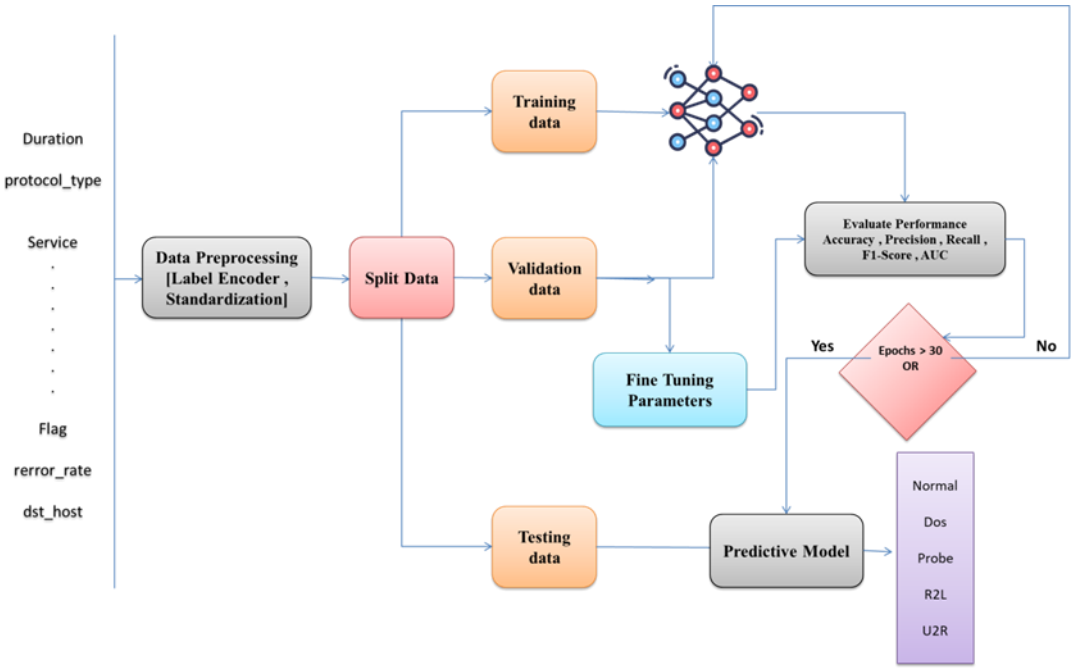

Figure 4. Deep learning pipeline for detect attacks.

The first step involves processing the NSLKDD dataset by converting categorical data into numerical format using a label binarizer. Subsequently, normalization is applied using the StandardScaler. Where The process of normalizing input data to achieve a consistent scale is crucial in enhancing the convergence speed and

reliability of the optimization algorithm. Subsequently, the dataset is divided into multiple subsets to facilitate training, and testing 80% , 20 % respectively , Where train data is divide also by 20% for validation. The training dataset is utilized to train DL models and subsequently assess their performance. In order to assess the model's performance and make necessary adjustments to its parameters, the validation dataset is employed. Finally, the use of the testing dataset is employed to evaluate the ultimate performance and transferability of the trained model on unfamiliar data. The overall DL pipeline in IDS is shown in Figure 4.

5 | Result and Discussion

This section investigates the performance of the proposed model using a widely-used dataset, NSLKDD [15]. In addition, it is compared to several DL models, such as CNN_BI-LSTM , LSTM , GRU and Machine learning algorithms such as Decision tree classifier. Those models are implemented in Python using the Kaggle platform and Keras API. The Adam optimizer was used to train the weights of those models for 30 epochs. The performance indicators used to evaluate the performance of those models are described as follows:

- Accuracy: The definition of this metric is the proportion of correctly predicted samples to all samples in a particular dataset. To compute this metric, use the equation that follows:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}}. \tag{3}$$

  where TP, FN, TN, and FP represent true positive, false negative, true negative, and false positive, respectively.

- Precision: Precision is a metric that quantifies how accurate positive predictions produced by a certain model are. The percentage of correctly detected positive instances to all anticipated positive instances is quantified by the statistic. This metric can be computed using the following equation:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}. \tag{4}$$

- Recall: This metric, which is also known as true positive rate or sensitivity, assesses how well the model can identify positive samples out of all the real positive samples. This metric can be computed using the following equation:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}. \tag{5}$$

- F1-score: The F1 score offers a fair evaluation of model performance by integrating recall and precision into a single metric. This metric can be computed using the following equation:

$$\text{F1} - \text{score} = 2 * \frac{Precision \cdot Recall}{Precision + Recal}. \tag{6}$$

  AUC: It shows how well the model can discriminate between positive and negative examples; a greater AUC denotes superior performance. This metric can be computed using the following equation:

$$\text{AUC} = \frac{1 + \frac{\text{TP}}{\text{TP+TN}} - \frac{\text{FP}}{\text{FP+TN}}}{2} \tag{7}$$

- Confusion Matrix: A confusion matrix is a visual aid that offers a brief summary of a machine learning and deep learning model's performance on a particular dataset. It is a way of displaying the number of accurate and inaccurate occurrences based on the model's predictions. It is common practice to assess the effectiveness of classification models using the previously mentioned matrices, such as F1Score, Accuracy, Permission, and Recall. When all of the true values are as enormous as they can be, the model performs at its peak.

- ROC Curve: The receiver operating characteristic (ROC) curve shows how well a model performs in classification. At various classification thresholds, it plots the specificity (1 - false positive rate) against the sensitivity (true positive rate). A greater ROC curve denotes better performance, and it is used to assess the model's ability to distinguish between positive and negative cases.

- TSNE: A dimensionality reduction method called t-SNE (t-Distributed Stochastic Neighbor Embedding) is frequently used to display high-dimensional data in a low-dimensional space. For exploratory data analysis, it is helpful. Its goal is to map data points so that related points are positioned in close proximity to one another. By grouping comparable classes after the prediction process, we may determine the model's effectiveness based on the increased distance between related classes.

As shown by the data shown in Table 1, The "proposed model" demonstrates exceptional performance in detect attacks , as seen by its highest accuracy, precision, recall, F1-Score, and AUC accuracy score of 0.995, 0.911, 0.869, 0.887, and 0.933 respectively, surpassing all other models.

Although the proposed model possesses the greatest number of parameters, it also demonstrates a commendable level of time consumption for both training and inference in comparison to alternative models, indicating effective utilization of resources.

The "Decision Tree" model has the lowest levels of accuracy, precision, recall, F1-Score, and AUC when compared to other models, suggesting comparatively inferior performance.

The "CNN-BILSTM" model exhibits the greatest computational time and parameter count compared to other models, indicating a higher demand on computational resources and time during both training and inference processes.

In Table 1 offers significant insights into the classification model's performance, demonstrating its accuracy in correctly categorizing examples into each class and identifying any potential misclassifications. In Figure 5 illustrate the performance evaluation of the proposed model is examined under the accuracy curve, loss curve, ROC curve and TSNE histograms.

According to the accuracy curve (a) and loss curve (b), The learning curve illustrates the evolution of the model's accuracy and loss during a 30 epochs dedicated to training and validation. The initial performance of the model on the training set was 92.84% accuracy and 0.2667 loss. Subsequently, the model achieved a validation accuracy of 98.93% and a validation loss of 0.0426. Throughout the training process, the model exhibited consistent enhancement, as evidenced by an increase in accuracy to 99.78% and a decrease in loss to 0.0082 by the conclusion of the 30 epochs. Simultaneously, the validation accuracy achieved a value of 99.66%, together with a validation loss of 0.0219. The shown learning curve demonstrates the model's aptitude for acquiring knowledge and making adaptations based on the training data, resulting in notable levels of accuracy and minimal loss on both the training and validation datasets.

The receiver operating characteristic (ROC) curve (c) for a 5-class classification model. It is seen that the macro average (AUC) attains a value of 93%, with all classes exhibiting a value over 96%, with the exception of the R2L class, which is limited by insufficient data. According to the TSNE plot (d), similar predicted values are grouped close to each other. This results are demonstrated in Figure 6. Table 2 presents the confusion matrix of the proposed model.

Table 1. Comparison between the proposed model and others in terms of various performance indicators.

Model Name	Consuming time	# Parameters	Accuracy	Precision	Recall	F1-Score	AUC
Proposed Model	26.646	706,355	0.995	0.911	0.869	0.887	0.933
LSTM	23.000	525,755	0.995	0.833	0.820	0.822	0.909
CNN-BILSTM	187.941	1,315,845	0.991	0.782	0.710	0.737	0.853
GRU	23.748	395,855	0.995	0.778	0.772	0.775	0.885
Decision Tree	1.699	-	0.958	0.566	0.558	0.562	0.771

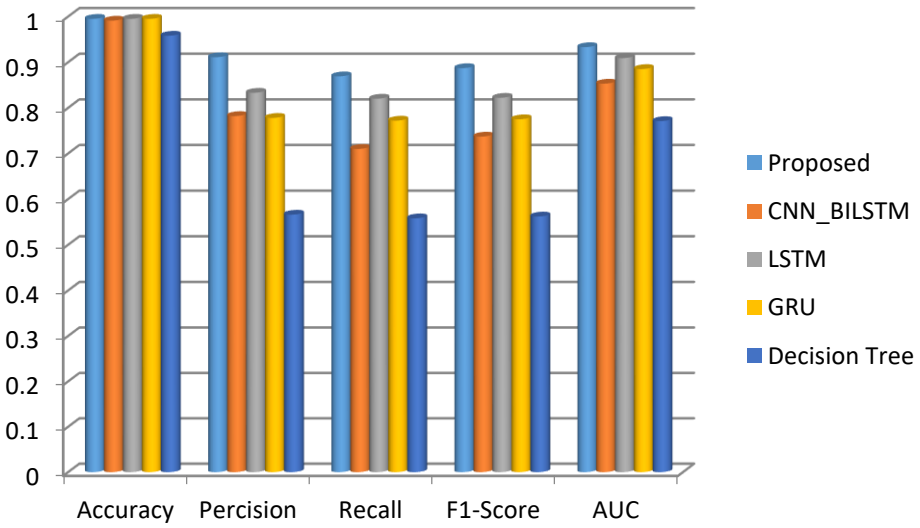

Figure 5. Comparison between the proposed model and others in terms of various performance indicators.

Table 2. Confusion matrix of the proposed model.

		Estimated classes					
Actual classes		**normal**	**Dos**	**Prob**	**R2L**	**U2R**	**Recall (%)**
	normal	**9171**	0	0	0	10	1.00 %
	Dos	1	**2309**	1	0	46	0.98 %
	Probe	0	0	**205**	0	19	0.92 %
	R2L	0	0	0	**5**	6	0.45 %
	U2R	7	6	12	3	**13394**	1.00 %
Precision (%)		1.00%	1.00 %	0.94 %	0.62 %	0.99%	

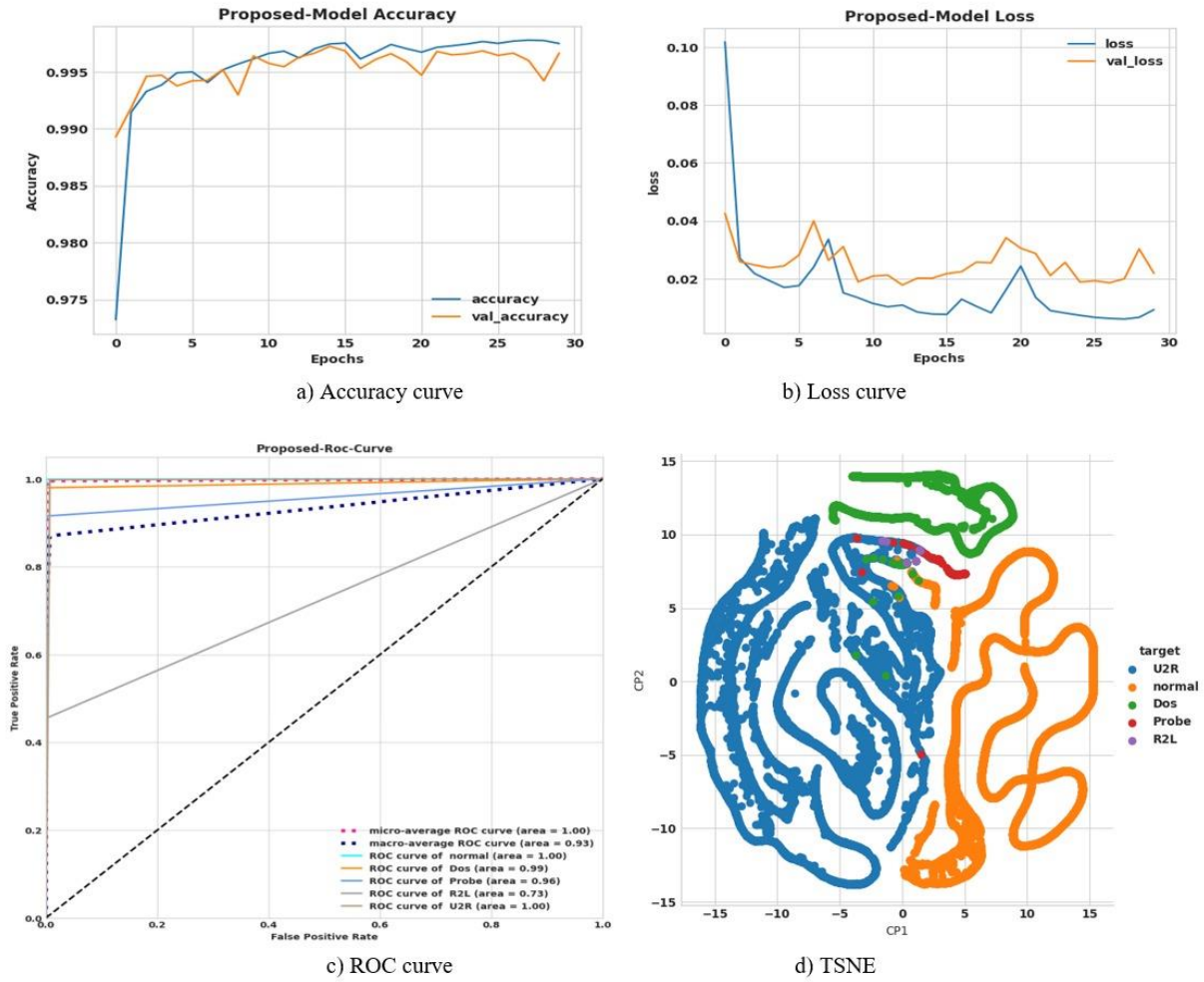

Figure 6. Performance evaluation of the proposed model under accuracy curve, loss curve, ROC curve, and TSNE events.

6 | Conclusion and Future Work

This paper introduces a novel model that features a robust approach to classifying network intrusions. Traditional intrusion detection cannot enhance the strength and flexibility of IDS. A hybrid model between LSTM and residual network has been presented in this study. This integration aims to eliminate the gradient vanishing problem and avoid overfitting. The model enables the smooth flow of information through the network and preserving information from earlier layers. The proposed algorithm is evaluated on KSL-KDD dataset to classify between normal or an attack, with each attack either being a DoS, U2R, R2L, or a probe. The model achieves superior results in accuracy, precision, recall, F1-Score, and AUC accuracy score of 0.995, 0.911, 0.869, 0.887, and 0.933 respectively.

Despite the significant advancements made in IDS through DL methods, several challenges persist. Most intrusion detection datasets face challenges due to the existence of new types of attacks. Therefore, emerging new datasets that should include common attacks, correspond to current network environments, be representative, balanced, and have less redundancy and noise is needed to enhance research studies. Also, more unsupervised learning methods are needed to be implemented on IDS where the real data is not labeled.

Acknowledgments

The author is grateful to the editorial and reviewers, as well as the correspondent author, who offered assistance in the form of advice, assessment, and checking during the study period.

Author Contributaion

All authors contributed equally to this work.

Funding

This research has no funding source.

Data Availability

The datasets generated during and/or analyzed during the current study are not publicly available due to the privacy-preserving nature of the data but are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there is no conflict of interest in the research.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Silivery, A.K., et al., A model for multi-attack classification to improve intrusion detection performance using deep learning approaches. Measurement: Sensors, 2023. 30: p. 100924.
[2] Saranya, T., et al., Performance analysis of machine learning algorithms in intrusion detection system: A review. Procedia Computer Science, 2020. 171: p. 1251-1260.
[3] Khraisat, A., I. Gondal, and P. Vamplew. An anomaly intrusion detection system using C5 decision tree classifier. in Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2018 Workshops, BDASC, BDM, ML4Cyber, PAISI, DaMEMO, Melbourne, VIC, Australia, June 3, 2018, Revised Selected Papers 22. 2018. Springer.
[4] Díaz-Verdejo, J., et al., On the detection capabilities of signature-based intrusion detection systems in the context of web attacks. Applied Sciences, 2022. 12(2): p. 852.
[5] Chopra, A., Security issues of firewall. Int. J. P2P Netw. Trends Technol, 2016. 22(1): p. 4-9.
[6] Quadar, N., et al., Intrusion Detection Systems in Automotive Ethernet Networks: Challenges, Opportunities and Future Research Trends. IEEE Internet of Things Magazine, 2024. 7(2): p. 62-68.
[7] Hnamte, V., et al., A novel two-stage deep learning model for network intrusion detection: LSTM-AE. IEEE Access, 2023.
[8] Nguyen, H. and R. Kashef, TS-IDS: Traffic-aware self-supervised learning for IoT Network Intrusion Detection. Knowledge-Based Systems, 2023. 279: p. 110966.
[9] Du, J., et al., NIDS-CNNLSTM: Network intrusion detection classification model based on deep learning. IEEE Access, 2023. 11: p. 24808-24821.
[10] Binbusayyis, A., Hybrid VGG19 and 2D-CNN for intrusion detection in the FOG-cloud environment. Expert Systems with Applications, 2024. 238: p. 121758.
[11] Roshan, K., A. Zafar, and S.B.U. Haque, Untargeted white-box adversarial attack with heuristic defence methods in real-time deep learning based network intrusion detection system. Computer Communications, 2024. 218: p. 97-113.
[12] Biyyapu, N., et al., Designing a modified feature aggregation model with hybrid sampling techniques for network intrusion detection. Cluster Computing, 2024: p. 1-19.
[13] Hochreiter, S. and J. Schmidhuber, Long short-term memory. Neural computation, 1997. 9(8): p. 1735-1780.
[14] Targ, S., D. Almeida, and K. Lyman, Resnet in resnet: Generalizing residual architectures. arXiv preprint arXiv:1603.08029, 2016.
[15] NSL KDD dataset.