International Journal of
Software Computing and Testing

ISSN: 2456-2351
Volume 10, Issue 1, 2024
January–June
DOI (Journal): 10.37628/IJSCT

https://journalspub.com/journal/ijsct

Review IJSCT

Intrusion Detection System Using Gated Recurrent Neural Network

Purushotam Naidu K.[1],*, S. Deepika[2], N. Sri Ramya[3],
N. Sharmila[4], K. Vaishnavi[5], P. Bhavitha[6]

Abstract

In the realm of cybersecurity, the constant evolution of threats demands sophisticated intrusion detection systems (IDSs) capable of discerning intricate patterns in network traffic. This study proposes an IDS leveraging the capabilities of gated recurrent neural networks (GRNNs) to enhance the detection of anomalies and potential security breaches. The GRNN architecture, employing mechanisms like long short-term memory (LSTM) and gated recurrent unit (GRU), demonstrates efficacy in capturing long-range dependencies within sequential data, a critical attribute for analyzing network traffic. The proposed system undergoes a comprehensive process, including data collection, preprocessing, and training on a labeled dataset encompassing normal and malicious network behaviors. During the training phase, the GRNN refines its parameters to recognize patterns in network traffic. In the operational phase, the system continuously analyses incoming traffic, employing a predefined threshold to trigger alarms upon detecting potential intrusions. The benefits of employing GRNNs lie in their adaptability to changing traffic patterns and their capability to provide real-time intrusion detection. However, challenges include the need for substantial labeled training data and careful model optimization. The proposed IDS not only contributes to the arsenal of cybersecurity tools but also underscores the importance of leveraging advanced neural network architectures for effective and adaptive network security in the face of evolving threats.

Keywords: Neural networks, cybersecurity, GRU, IDS, real-time detection, XGBoost, GRNN, LSTM

INTRODUCTION

The escalating complexity of cyber threats in today's interconnected world necessitates robust and adaptive security measures. In safeguarding digital environments, intrusion detection systems (IDSs) play a crucial role by monitoring network activities and detecting anomalies that may signal security breaches. While conventional IDS methods have shown efficacy, the dynamic landscape of cyber threats necessitates more advanced approaches. Figure 1 illustrates two distinct types of IDS.

A Host-Based IDS (HIDS) operates on a specific endpoint, aiming to safeguard it from both internal and external threats. It possesses capabilities such as monitoring network traffic to and from the endpoint, observing active processes, and examining system logs. While its scope is confined to the host machine, reducing contextual information for decision-making, it offers extensive insight into the internal workings of the host computer.

A Network-Based IDS (NIDS) is crafted to oversee the entirety of a secured network. It possesses insight into all network traffic and makes

*__Author for Correspondence__
Purushotam Naidu K.
E-mail: purushotam.k30@gmail.com

[1]Assistant Professor, Department of Computer Science and Engineering (AI & ML), GVP College of Engineering for Women, Visakhapatnam, Andhra Pradesh, India
[2-6]Student, Department of Computer Science and Engineering (AI & ML), GVP College of Engineering for Women, Visakhapatnam, Andhra Pradesh, India

Received Date: March 28, 2024
Accepted Date: April 19, 2024
Published Date: May 29, 2024

Citation: Purushotam Naidu K., S. Deepika, N. Sri Ramya, N. Sharmila, K. Vaishnavi, P. Bhavitha. Intrusion Detection System Using Gated Recurrent Neural Network. International Journal of Software Computing and Testing. 2024; 10(1): 1–8p.

assessments using packet metadata and contents. This broader perspective offers enhanced context and the capacity to identify extensive threats. Nevertheless, these systems lack visibility into the internal workings of the endpoints they safeguard.

This article presents a novel method for intrusion detection, utilizing gated recurrent neural networks (GRNNs). GRNNs are part of the recurrent neural network (RNN) category and possess a unique ability to capture temporal relationships within sequential data. By incorporating GRNNs into the design of IDS, this research seeks to enhance the accuracy and efficiency of IDS.

The motivation for integrating GRNNs into IDS stems from their inherent capacity to model and analyze sequential data. Cyber threats often manifest as subtle changes in network behavior over time, posing a challenge for traditional IDS relying on static patterns. GRNNs, with their recurrent connections and memory capabilities, present a promising solution to effectively capture these temporal patterns.

The adaptability of GRNNs to dynamic and evolving attack strategies is another motivating factor. In the ever-changing landscape of cybersecurity, where attackers constantly innovate, the ability to learn from new patterns and improve detection capabilities is crucial.

LITERATURE SURVEY

Jadhav KP et al. [1] within the framework, various forms of RNNs are employed, including long short-term memory (LSTM), gated recurrent unit (GRU), and simple RNN. To assess the performance of the IDS framework, NSL-KDD and UNSW-NB15 benchmark datasets are considered. Existing IDSs struggle with low accuracy in detecting new attacks as the data complexity increases. To overcome this, the study implemented an XGBoost-based feature selection algorithm, significantly reducing the dataset’s dimensionality [2]. GRU and Support Vector Machines (SVM) for intrusion detection in network traffic data can lead to a powerful hybrid neural network architecture the 2013 Kyoto University honeypot systems’ network traffic data is used. This research introduces an advanced model for network intrusion detection, integrating a convolutional neural network (CNN) with a GRU. To address imbalances in positive and negative samples in the original dataset, a hybrid sampling algorithm combining Adaptive Synthetic Sampling (ADASYN) and Repeated Edited nearest neighbors (RENN) is employed. Feature selection involves a combination of the random forest algorithm and Pearson correlation analysis to mitigate feature redundancy. Spatial features are extracted using a CNN and further refined through a fusion of Average pooling and Max pooling, with an attention mechanism assigning varying weights to features to reduce overhead and enhance model performance. Additionally, GRU is utilized to extract long-distance dependent information features. The classification is accomplished through the SoftMax function. The study employs datasets such as UNSW_NB15, NSL-KDD, and CIC-IDS2017 [3].

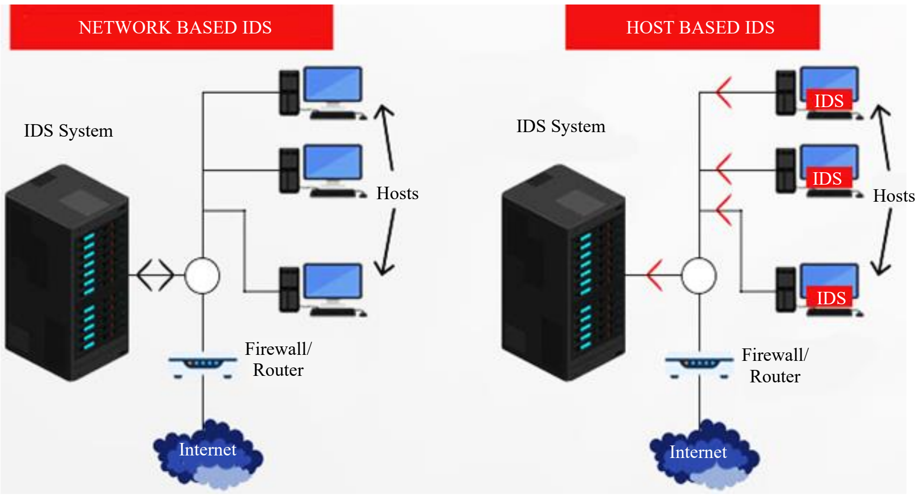

Figure 1. Types of IDS.

The LA-GRU model is developed for constructing a combined IDS utilizing imbalanced learning techniques and GRU neural networks [4]. It outlines the implementation guidelines for enhanced versions of the Adam optimizer, GRU, and LA-SMOTE algorithms, demonstrating how GRU captures temporal data for anomaly detection in traffic samples. Experiments are conducted on the NSL-KDD dataset using TensorFlow as the simulation software. The hardware platform chosen for empirical tests is a desktop equipped with an Intel Core i7-8700K hexa-core processor, 128G SSD, 16G RAM, and running on the Windows 10 operating system [5]. *Deep learning (DL) GRU:* This groundbreaking research introduces deep learning methods for Internet of Things (IoT) security, specifically focusing on network data security. The study develops a lightweight IDS architecture for IoT networks, placing IDS classifiers at different TCP/IP layers to enhance accuracy and reduce false alarms. The utilization of RNNs, specifically LSTM and GRU algorithms, in deep learning proves advantageous for IoT devices due to their capacity to learn from previous time steps, requiring minimal human intervention [5, 6]. The KDD Cup 1999 dataset serves as a prevalent benchmark dataset for intrusion detection. This study employs deep learning's gated recurrent neural networks (LSTM and GRU) algorithms. The RNN-IDS model demonstrates robust modeling capabilities for intrusion detection, delivering high accuracy in both binary and multiclass classification tasks. A dataset is NSL-KDD. RNN-IDS on the test set KDDTest+ in the five-category classification experiments. The experimental results indicate that the model achieves an accuracy of 81.29% on the KDDTest+ test set and 64.67% on the KDDTest−21 test set [7]. IDS WIFI: the focus is on DL models for IDS. Deep learning employs hierarchical data processing stages to perform unsupervised feature learning and pattern classification. It independently learns feature representations from data, removing the necessity for manual feature engineering. There is a dataset so-called Aegean Wi-Fi Intrusion Dataset (AWID) dataset is used.

IDS in IoT using GRU: This research addresses the security challenges in the IoT by proposing a lightweight and distributed security solution [8]. Traditional security methods are impractical for low-capacity IoT devices that stay connected without human intervention. The study combines DL techniques, specifically gated recurrent neural networks (LSTM and GRU), with TCP/IP protocols to create a robust IoT IDS. The dataset utilized for intrusion detection is the DARPA/KDD Cup 1999 dataset. This innovative approach marks the first application of gated recurrent neural networks in IoT security, outperforming existing methods [9]. This research presents a sophisticated IDS employing a hybrid DL method known as a bidirectional RNN with both LSTM and GRU components. The research compares this model to traditional methods, showcasing its exceptional performance on the CICIDS2017 dataset. The study also highlights the individual performance of LSTM and GRU with RNN, emphasizing the superiority of DL techniques in IDS. The proposed model achieved a remarkable 99.13% accuracy in predicting network attacks, outperforming Naïve Bayes classifiers in accuracy and false positive rate. Notably, the model performed well using only 58% of the dataset attributes, demonstrating its efficiency and effectiveness in intrusion detection [10]. An IDS is introduced, utilizing an improved RNN to detect different types of intrusions. IDS using the enhanced RNN with other machine learning algorithms. Evaluation is conducted on two datasets: a smaller subset of the KDD-99 dataset comprising a thousand instances and the complete KDD-99 dataset. The aim is to identify the most effective method for intrusion detection among various machine learning algorithms, particularly focusing on RNN's enhanced version, in the context of different dataset sizes and complexities.

This paper presents the design of an IDS utilizing ensemble learning techniques within a Cloud Computing environment [11]. The model is trained and assessed using two datasets: CICIDS 2018 and SDN-based Distributed Denial-of-Service (DDoS) attack datasets. Commonly employed deep learning-based approaches such as RNN, ANN, DBN, DBM, and Autoencoder are utilized for input data processing. Additionally, the algorithm's performance is augmented through hybrid combinations of AlexNet, FractalNet, GoogLeNet, VGG, Dense CNN, and the K-means clustering algorithm [12]. DNN model which uses GRUs as the main memory unit, combined with multilayer perceptron (MLP) to identify network intrusions. KDD-99 and NSL-KDD datasets. The comprehensive detection rates reached 99.42% for KDD-99 and 99.31% for NSL-KDD, accompanied by notably low false positive rates of 0.05% and 0.84%, respectively. In particular, the detection rates for DoS attacks were 99.98% on KDD-99 and 99.55% on NSL-KDD. The next step could be to optimize the system so that it can be applied to real network environments and be implemented more efficiently [13]. The intrusion detection technology can be divided into three major categories: pattern-matching methods, traditional machine learning methods, and DL methods. We propose a principal component analysis (PCA) based novel hybrid method of wireless network intrusion detection using enhanced RNNs. We utilized two widely recognized public IDS datasets, specifically NSL-KDD and UNSW-NB15. LSTM, GRU requires only two gates to achieve its proper functionality: the update gate, and the reset gate. PCA is used for dimensionality reduction, and for classifying the anomalies novel hybrid method is used [14]. The NSL-KDD dataset serves as the input dataset, encompassing normal, probe, U2R, R2L, and DoS attack types for the proposed system (RNN). The precision is 52.09, the recall is 91.66 F, the measure is 64 and the accuracy is 92.55. The proposed NIDS system is improved by only 8% accuracy using an RNN. The proposed system also includes a comparison between an IDS utilizing alternative machine learning algorithms with a smaller subset of the KDD-99 dataset containing a thousand instances and the complete KDD-99 dataset.

METHODOLOGY

In this research work a comprehensive set of libraries commonly used for data analysis, machine learning, and visualization in Python has been imported. Starting with foundational libraries such as NumPy and Pandas, which facilitate efficient handling and manipulation of data through arrays, matrices, and data frames, the code also incorporates libraries for data preprocessing, scaling, and visualization. Additionally, it imports machine learning frameworks like TensorFlow and XGBoost, along with various classifiers and regressors from the Scikit-learn library, enabling a diverse range of predictive modeling tasks. Notably, the code configures the environment to suppress warnings and ensures that all columns are displayed when presenting data frames. This comprehensive selection of libraries equips the user with a robust toolkit for data exploration, model building, and analysis tasks in Python as shown in Figure 2.

The model outlines a comprehensive data analysis and preprocessing workflow, followed by the construction and training of a neural network model for classification and regression tasks. The initial steps involve loading a dataset called NSL-KDD and using the Pandas library to read it into a DataFrame.

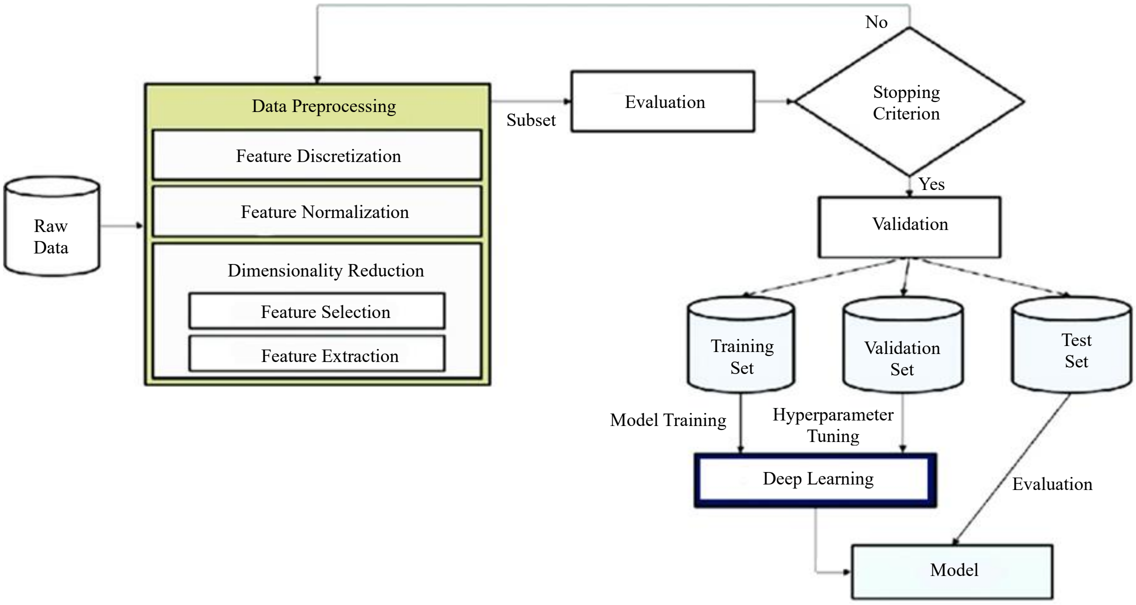

Figure 2. System architecture.

This DataFrame is then examined to understand its structure and contents, including an initial glimpse at its first few rows to gain insights into the data's format. Subsequently, the model proceeds with standardizing the column names by defining a list of expected column names. This ensures consistency in data representation by assigning the list to the DataFrame's column attribute. Data exploration techniques are then employed to understand the dataset's characteristics, such as using the `info()` method to summarize the DataFrame's structure and the `describe()` method to generate descriptive statistics for numerical features. Additionally, visual representation using a color gradient aids in identifying patterns and outliers within the data, while customizing the font family enhances the clarity of statistical insights.

Data transformation is performed to modify values in the 'outcome' column, simplifying the classification task by replacing non-'normal' outcomes with the label 'attack'. This step lays the foundation for subsequent machine learning tasks focused on detecting and classifying network intrusions effectively. Following data preprocessing, the model implements essential functions and processes for data scaling and preprocessing, critical stages in machine learning pipelines as shown in Figure 2. The 'Scaling()' function is defined to standardize numerical features using RobustScaler from Scikit-learn, enhancing their robustness to outliers. Categorical columns are identified and removed, and numerical features are standardized using the 'Scaling()' function. Furthermore, the 'outcome' column is transformed into binary labels, facilitating binary classification tasks. Following that, PCA is employed for dimensionality reduction, reducing the dataset's dimensions while preserving important information. Subsequently, the dataset is divided into training and testing subsets to facilitate both classification and regression tasks. For classification, the features and corresponding labels undergo stratified splitting, ensuring balanced representation in both training and testing sets. Simultaneously, separate training and testing sets are prepared for regression tasks.

Next, a neural network model is constructed using the Keras API with TensorFlow backend, specifically designed for sequential data processing. The model architecture consists of GRU layers, known for capturing temporal dependencies within sequential data. The model is initialized with two GRU layers followed by a dense output layer for binary classification tasks. The assembled model is set up with an optimizer, loss function, and evaluation metrics. Adam optimizer is preferred due to its adaptive learning rate characteristics, the Binary Cross entropy loss function is employed for binary classification tasks, and 'accuracy' is opted for as the evaluation metric. Subsequently, the model is trained using the fit() method, utilizing the provided training data while keeping track of its performance on the validation dataset. The training process persists for a designated number of epochs, during which the model fine-tunes its parameters to minimize the loss function and enhance its performance.

RESULTS AND DISCUSSION

Open-source software library Scikit-learn (Sklearn) was used to construct learning models. In this work proposed model was compared with logistic regression, K-Nearest Neighbour (KNN), Naïve Bayes, SVM, random forest, decision tree, and XGBoost Classifiers. All these models were constructed in Python in the most efficient manner feasible to provide fair comparisons. Their performance was compared using identical software and hardware.

Logistic regression is a statistical technique employed in binary classification scenarios, where the dependent variable consists of two categorical outcomes. It estimates the probability of an input belonging to a particular category by adjusting a logistic function to match the observed data. The training accuracy of the model which is built using the logistic regression algorithm is 87.97%. The test accuracy of the model which is built using the logistic regression algorithm is 87.62%.

A supervised learning technique such as the Naïve Bayes algorithm is employed to solve classification issues. GaussianNB is used here. This algorithm is based on Bayes theorem. The classification of the test data set can be quickly and easily predicted. Additionally, it does very well in the area of multiclass prediction. The training accuracy of the model which is built using the Naïve

Bayes algorithm is 91.8%. The test accuracy of the model which is built using the Naïve Bayes algorithm is 91.6%.

KNN relies on a supervised learning approach. Predicting or classifying a new unknown variable requires knowing how many of its nearest neighbors are known. KNN determines the shortest paths to the unknown data by calculating the distances between each location in the neighborhood and the unknown data. The term "distance-based algorithm" has become common because of this. The training accuracy of the model which is built using the KNN algorithm is 99.05%. The test accuracy of the model which is built using the KNN algorithm is 98.93%.

XGBoost is a machine learning ensemble technique that employs gradient boosting decision trees. It implements the gradient boosting concept with a more regulated model formalization to manage overfitting. The XGBoost model achieved a training loss of 0.94 and a test loss of 1.001.

Random forests, functioning as an ensemble learning technique, have the capability to classify and make predictions. There are many independent decision trees in this system, but they work together as a unit.

The predictions of our model rely on the class predictions generated by each tree within the random forest. The training accuracy of the model which is built using random forest is 99.994% and the test accuracy is 99.87%. Measuring the effect of PCA Training Accuracy is 99.94% and the test accuracy is 99.82%.

The decision tree is a supervised learning system. It solves regression and classification problems. Decision Node and Leaf Node are nodes in a decision tree. Leaves are decision results, while decision nodes make multi-branch decisions. Visualize all possibilities for addressing an issue or making a decision. Training accuracy is 99.99% and test accuracy is 99.86% using a decision tree.

SVM classifies data points by finding an N-dimensional hyperplane. Features determine hyperplane size. For example, two input features yield a straight hyperplane. Model training accuracy is 97.48% and testing accuracy is 97.28%. Here, LinearSVC (LBasedImp1) is used.

In the realm of machine learning, the term CNN typically stands for Convolutional Neural Networks, which are a category of DL algorithms frequently employed for tasks involving image recognition and classification. Model training accuracy is 97.01% and testing accuracy is 96.41%.

Gated Recurrent Neural Network

The training accuracy of the GRNN is 93.66%. The test accuracy of the GRNN is 93.41%. Training accuracy for the GRU model is higher than logistic regression and Naïve Bayes models. Its accuracy is less than KNN, SVM, decision tree, Random Tree, and CNN models. Testing accuracy for the GRNN model is higher than logistic regression, and Naïve Bayes models.

Table 1. Models accuracy in %.

Model	Training accuracy	Test accuracy
Logistic regression	87.97	87.62
K-nearest neighbors	99.05	98.93
Naïve Bayes	91.80	91.60
SVM	97.48	97.28
Decision tree	99.99	99.86
Random forest	99.99	99.87
PCA_Random Forest	99.99	99.82
CNN	97.07	96.41
GRNN	93.66	93.41

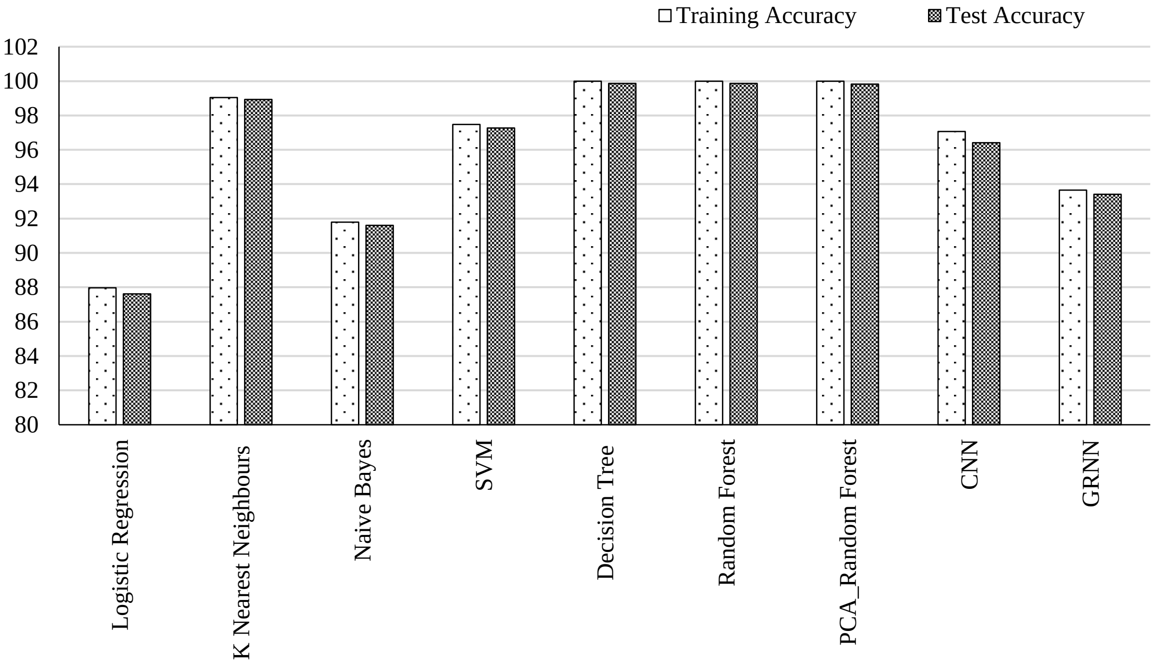

Figure 3. Algorithms accuracy.

The GRNN model train accuracy and test accuracy are compared with various other machine learning algorithms in Table 1 and Figure 3.

CONCLUSION

The implementation of an IDS utilizing GRNNs presents a promising avenue for enhancing network security. The comprehensive process, encompassing data collection, preprocessing, and training on a labeled dataset, demonstrates the system's commitment to addressing both normal and malicious network behaviors. The architecture of the GRNN, incorporating elements like LSTM and GRU, proves to be effective in capturing intricate patterns and long-range dependencies inherent in sequential data. In this work, a deep learning GRU model was proposed for the IDS. In this model, 7 different algorithms such as logistic regression, Naïve Bayes, random forest, decision tree, PCA_Random Forest, KNN, and SVM algorithms were used to achieve the best results. The proposed model accuracy was compared with seven different machine algorithms and the results show that the proposed model has not given more accuracy compared to some models like KNN, SVM, decision tree, random forest, PCA_Random Forest, and CNN.

REFERENCES

1. Jadhav KP, Arjariya T, Gangwar M. Hybrid-Ids: an approach for intrusion detection system with hybrid feature extraction technique using supervised machine learning. Int J Intell Syst Appl Eng. 2023;11:591-597.
2. Jadhav KP, Arjariya T, Gangwar M. Intrusion detection system using recurrent neural network-long short-term memory. Int J Intell Syst Appl Eng. 2023;11:563-573.
3. Udas PB, Karim ME, Roy KS. SPIDER: A shallow PCA based network intrusion detection system with enhanced recurrent neural networks. J King Saud Univ Comput Inf Sci. 2022;34:10246-10272. DOI: 10.1016/j.jksuci.2022.10.019.
4. Aldallal A. Toward efficient intrusion detection system using hybrid deep learning approach. Symmetry. 2022;14(9):1916. DOI: 10.3390/sym14091916.
5. Gautam S, Henry A, Zuhair M, Rashid M, Javed AR, Maddikunta PK. A composite approach of intrusion detection systems: hybrid RNN and correlation-based feature optimization. Electronics. 2022;11:3529. DOI: 10.3390/electronics11213529.

6. Yin C, Zhu Y, Fei J, He X. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access. 2017;5:21954-21961. DOI: 10.1109/ACCESS.2017.2762418.
7. Almasoudy FH, Al-Yaseen WL, Idrees AK. Differential evolution wrapper feature selection for intrusion detection system. Procedia Comput Sci. 2020;167:1230-1239. DOI: 10.1016/j.procs.2020.03.438.
8. Ge Y, Li J, Tian Y. Internet of Things Intrusion Detection System Based on D-GRU. 2022 4th International Conference on Applied Machine Learning (ICAML); 2022 Jun 17-19; Changsha, China. IEEE; 2022. p. 1-6. doi: 10.1109/ICAML57167.2022.00066.
9. Zhang M, Fernández-Torres MÁ, Camps-Valls G. Hybrid Recurrent Neural Network for Drought Monitoring. NeurIPS 2022 Workshop on Tackling Climate Change with Machine Learning; 2022. Available from: https://www.climatechange.ai/papers/neurips2022/51.
10. Bingu R, Jothilakshmi S. Design of Intrusion Detection System using Ensemble Learning Technique in Cloud Computing Environment. Int J Adv Comput Sci Appl (IJACSA). 2023;14(5). doi: 10.14569/IJACSA.2023.0140580.
11. Xu C, Shen J, Du X, Zhang F. An intrusion detection system using a deep neural network with gated recurrent units. IEEE Access. 2018;6:48697-48707. doi: 10.1109/ACCESS.2018.2867564.
12. Roy KS, Ahmed T, Udas PB, Karim ME, Majumdar S. Malhystack: A hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis. Intell Syst Appl. 2023;20:200283. DOI: 10.1016/j.iswa.2023.200283.
13. Gunjal SP, Aher SM. Network intrusion detection using recurrent neural network algorithm. Int Res J Eng Technol. 2020;07:2143-2148.
14. Kasongo SM. A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. Comput Commun. 2023;199:113-125. DOI: 10.1016/j.comcom.2022.12.010.