2024 27th International Conference on Computer and Information Technology (ICCIT)
20-22 December 2024, Cox's Bazar, Bangladesh

Evaluating Deep Learning Models for Network Intrusion Detection: A Comparative Analysis

Mahbubul Islam*, Md. Muntasir Jahid Ayan*, Emrul Kais*, Rupak Kumar Das†, Md Motaharul Islam*
*Department of CSE, United International University (UIU), Dhaka 1212, Bangladesh
Email: {mislam201011@ms, mayan201439@bs, ekais201007@ms, motaharul@}cse.uiu.ac.bd
†College Of Information Sciences and Technology, Pennsylvania State University, University Park, PA 16802, USA
Email: rjd6099@psu.edu

***Abstract*—The exponential rise of the Internet of Things has opened up immense opportunities. It also increases network security risks. If attacks are not monitored or prevented at the early stage, they can create acute losses to enterprises, industry, and personnel. Deep Learning and Machine Learning based intelligent intrusion detection system are the answers to this challenge. This paper proposes an Intrusion Detection System (IDS) using popular DL and ML algorithms. We used the KDDcup99 dataset to compare supervised ML algorithms to DL algorithms. We evaluated the algorithm's performance by comparing metrics such as accuracy, precision, F1-score, as well as sensitivity (TPR), and the rate of false positives (FPR). Our experiment showed that DL-based algorithms are more accurately predicted than ML algorithms regarding all performance parameters used in this paper. BiLSTM performed highest in all measures compared to other algorithms and accuracy at 98%. These results illustrate DL algorithms' strong capability and potential over ML to detect network attacks.**

***Index Terms*—Intrusion detection, Machine learning, Network security, KDDcup99**

I. Introduction

Network security concern arise in a gradual manner with the increasing internet use. Intrusion is considered the most widely reported security breach on global and local internet traffic. The intrusion detection system (IDS) has been familiar to detect unwanted attack via internet, including preserving information security (IS) goals [1].

A good number of IDS system are currently used in global network, but a little number of them are artificial intelligence (AI) based. Integrating deep learning (DL) and machine learning (ML) based IDS is a huge exploration area in the field of computer networking and IS [2]. Internet heavily aids for business, education, entertainment and personal for last decades [3], [4]. Everyone is connected via the internet, which also stores a vast amount of data and sensitive personal information on a network that is open to attack. Any type of entry that occurs quickly has the potential to cause dangerous damage. Also, intrusion effect integrity, confidentiality availability of the data, information and network.So, the protecting of network and network device is very important [5], [6]. Many researchers are conducting IDS-based research using ML and DL techniques, but few of them are drawing comparisons between ML and DL. We have analyzed the gap and proposed an IDS system based on a comparative performance analysis of different ML and DL models, illustrating the best one. The framework diagram, Figure 1 illustrates the high level overview of ML and DL based IDS system.

We have used a benchmark dataset of intrusion detection named KDDcup99 [7]. We have selected 15 significant features that are closely related to intrusion nature detection. After that, all the ML algorithms, Multinomial Naive Bayes(NB), Linear Regression Classifier(LRC), Decision Tree(DT), Support Vector Machine (SVM) and DL algorithms Convolutional Neural Network (CNN), Long short-term memory (LSTM), A bidirectional LSTM (BiLSTM), Gated Recurrent Unit (GRU) based classifiers are applied upon the dataset. To assess the results, use indicators of performance metrics such as accuracy, precision, F1-score, as well as sensitivity (TPR), and the rate of false positives (FPR). [8]. As summarized, based on all the performance measure, it is clear that all the ML and DL algorithms performed significantly upon this dataset. According to the performance it is observed that, BiLSTM performed highest in the all measure Precision (98%), recall (98%), F1-score(98%). Our work complements the existing work and contributes towards providing the comparative performance analysis of DL and ML algorithms for computer network security and IDS. Below is a synopsis of the research's contributions:

- We analyzed the KDDCUP99 dataset to perform a comparative analysis of supervised ML techniques and deep learning algorithms.
- We have analyzed the performance of DL and ML algorithms based on various evaluation metrics, to measure the optimal algorithm for intrusion detection.
- We have highlighted best performed model based on each ML and DL category to understand the performance upon dataset.
- Finally, We have discussed significant challenges and future research direction of ML and DL based IDS for betterment to reduce research gap.

This research is organized into following orders and sections: Section I contains the introductions to this research, Section II reviews of related scientific research, accordingly materials and methods illustrates in Section III, Section IV represents the result analysis, Section V highlights on future research scope, and Section VI summarize the conclusion of the research.

979-8-3315-1909-4/24/31.00

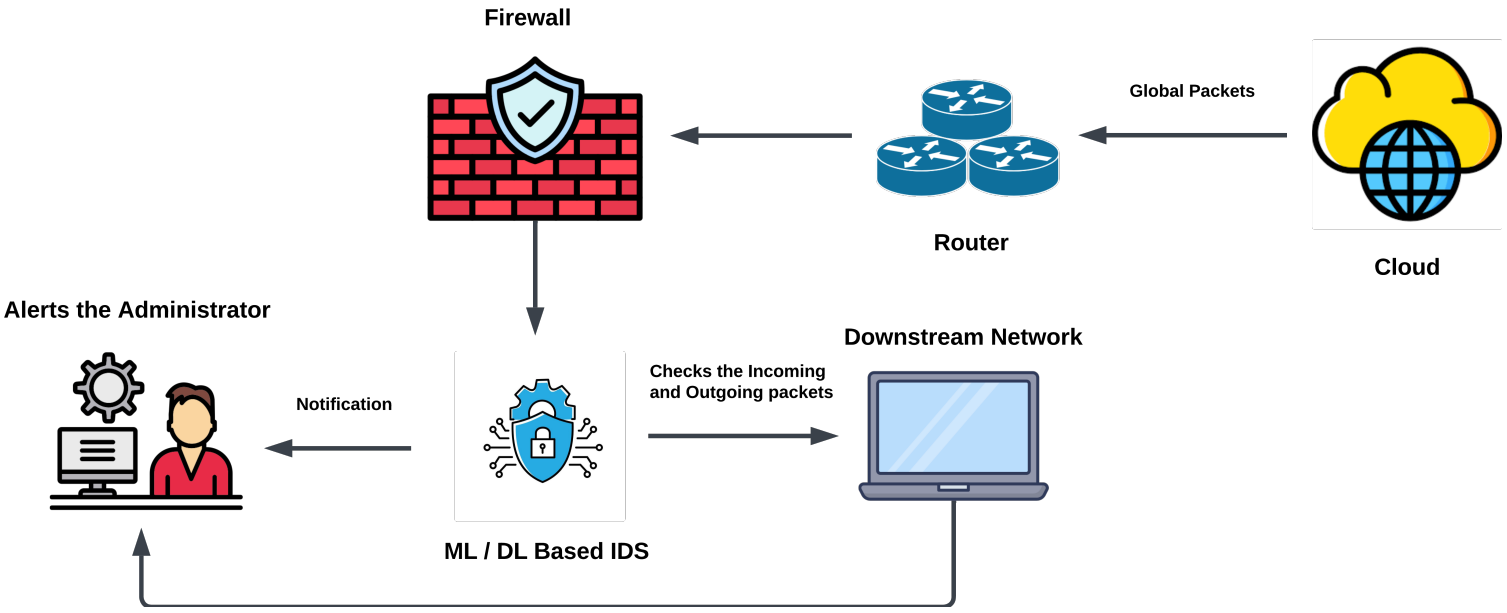


Fig. 1: Overview of ML/DL based IDS system

II. Literature Reviews

This section presents some relevant work on IDS based on DL and ML algorithms. A brief synopsis of the pertinent prior research and its contribution to this field is provided in Table I.

The ability of ML and DL models to extract characteristics as well as detect intricate patterns has made them attractive in the network anomaly detection field. Recent research [9]–[11] used the KDDCup'99 data to study DL approaches for detecting network intrusion. A non-symmetric deep autoencoder model using Support Vector Machine (SVM) proposed by Qazi et al. [10]; it evaluated various kinds of network assaults and minimized overfitting and computing complexity. CNN and CNN-LSTM combinations outperformed compared to other DL architectures on the KDD99 dataset, according to a comparison by Meliboev et al. [12] over a number of datasets. By employing SMOTE to up sample minority classes, they help lessen class imbalance. Researcher [13] proposed a hybrid multilayer DL model, a convolutional neural network for feature extraction paired with an LSTM, and a softmax classifier. Another research [14] established the IDS classification using classic ANNs by combining ANNs with Particle Swarm Optimization. Shokeen et al. [11] investigate by comparing several ML models for internet based attack detection, showed that voting, random forest, and decision tree methods have high accuracy rates. Although there are many outcomes in the field of IDS research, there is still opportunity for improvement in areas like lowering computational complexity, class imbalance, and the interpretability of network IDS, even though these studies show how effective DL techniques are in raising the efficiency of these systems.

III. Materials & Methods

This section describes the framework and methodology of the present work. The framework diagram Figure 2 illustrates the data processing steps, implementation of feature selection methods, deployment of ML and DL classifier. To explain the overall context, this research discusses the methods that were used as a combined solution for intrusion detection. We have used a benchmark dataset of intrusion detection named KDDcup99 [7]. The dataset was then divided into a training set (80%) and a testing set (20%). Before that, we used data preprocessing techniques, data cleaning for missing values, duplicate values and handling noise, outlier preparation, Minimax Scaler to scale the data, and performed label encoding on categorical features to process the dataset. Initially, the dataset contains 41 features, Using random forest, we have selected 15 significant features that are closely related to intrusion. Table III represents the selected classes, category of features and specific data type. After that, all the ML and DL based classifier are applied upon the dataset for binary classification to classify anomaly and normal. Accordingly, select some performance measure as accuracy, precision, F1-score, True positive rate, False positive rate to understand the outcome.

A. Dataset Description

We have used KDD'99 [7] which is the most widely used dataset for detecting anomalies. This data collection created by Stolfo et al. [16] based on DARPA'98 IDS evaluation program outcome. The KDDcup99 data includes two sets: the whole set approximately (18 M, 743 M) and the 10% subset (2.1 M, 75 M). In this research, we collect 10% KDD99 cup to evaluate the result. The 42nd feature in each set of data is the class feature, out of 41 attributes. In the dataset features

TABLE I: Summary of Related literature

Reference	Category	Dataset	Applied Model	Contribution
[15]	Single Classifier	10% KDD99	DT, NB	Implement improved hybrid model for detecting rare attack type
[10]	Hybrid	KDD99	SVM-based stacked non-symmetric deep auto-encoder	Less prone to overfitting, lower computing cost and the ability to assess a wide range of attacks
[12]	Single Classifier	KDD99, NSL-KDD	CNN, LSTM, RNN, GRU	Experiment with various DL models, Reduce imbalance by oversampling minority class with SMOTE
[13]	Hybrid	KDD99, NSL-KDD	CNN-LSTM combination	Improve traditional multilayer ANN by combining CNN-LSTM
[11]	Hybrid	KDD99	DT, LR, AdaBoost, KNN, NB and RF	Compared performance of various ML based single and ensemble classification algorithms,

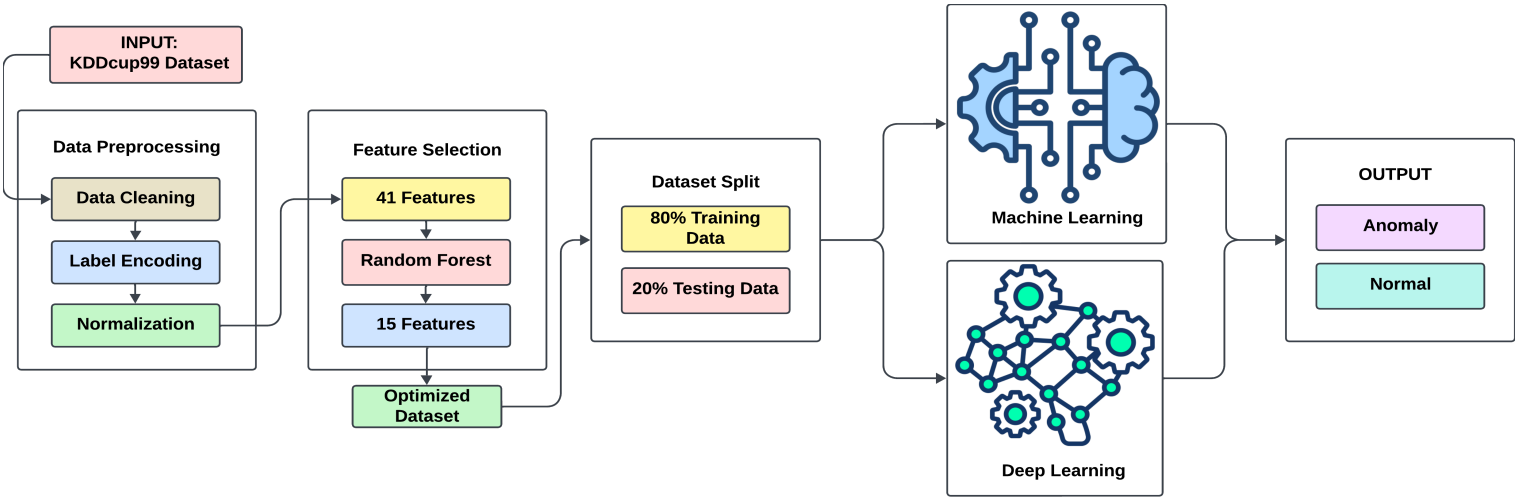


Fig. 2: Framework of the ML and DL based IDS system

derived based on some common attribute for network attack. Every network visit falls into one of the following categories: normal, probing, surveillance, other probing, Denial of Service (DOS), Unauthorized Remote Machine Access (R2L), and Unauthorized Access to Local Supervisor Privileges (U2R) [17]. An unapproved, dangerous event will be recorded based on a particular attribute [18].

Table II illustrates basic records of actual KDD99 data set in accordance with original record, distinct records and reduction rate be related with various types like attacks, normality and total count [18].

B. Algorithms Description

The DT classifier is a multistage decision-making model that can handle both numerical and nominal data.It makes quick choices because it uses a few layered simple conditional expressions. The major hyperparameter that affects DT is max depth [19], [20]. It can demonstrate how robust the tree's roots will be. Based on Bayesian theories, the Naive Bayes (NB) classifier is a set of corresponding algorithms. For measuring total number of output classes and the conditional likelihood for each class, conditional probability for each attribute will be determined. It is useful in the prediction of several different groups [21], [22].The logistic regression Classifier (LRC) model is used for solving classification problem. Logistic regression uses to classify data and show the connection between outcomes(dependent variable) and features like various types of categorical or nominal data (independent variable) and also predict based on independent variables which category or group dependent variable will go [23], [24]. In more DT, LRC, and NB, Support Vector Machines(SVM) are powerful methods that often utilized in classification tasks. SVM works by searching possibly the best hyperplane that differentiates data points. To find the best combination of parameters, SVM models are generally optimized using methods like grid search [25].

TABLE II: Summary of KDD99 dataset

Type	Original records (Million)	Distinct records(Million)	Reduction rate(%)
Attacks	3.926	0.02618	0.93
Normal	0.9728	0.8128	0.16
Total	4.898	1.075	0.78

Deep Learning Models

The LSTM, Bidirectional LSTM, Bidirectional GRU, and CNN models are developed for binary classification tasks

TABLE III: Summary of selected features

Selected Feature	Category	Data type
protocol_type	Basic features of TCP connection	Discrete
service		Discrete
src_bytes		Continuous
dst_bytes		Continuous
logged_in	Content feature of TCP connection	Discrete
count	Time based statistical features of network flow	Continuous
srv_count		Continuous
dst_host_conut	Host based Statistical features	Continuous
dst_host_srv_conut		Continuous
dst_host_diff_srv_rate		Continuous
dst_host_error_rate		Continuous

aimed at detecting network intrusions using the KDDCup99 dataset.All models are compiled using Adam as an optimizer and binary cross-entropy as loss functions, with accuracy serving as the key metric.

LSTM, Bidirectional LSTM, and Bidirectional GRU, each model contains 50 units with 2 layers [26]–[28]. For capturing long-term dependencies in temporal data, LSTM is specifically designed, In both directions forward and backward BiLSTM processes input sequences that make it particularly effective task to improve the performance in contextual information. In each model set the dropout layers with a 0.2 rate help minimize overfitting, and a final dense layer activated by a sigmoid function handles binary classification. And the CNN model, structured with two convolutional layers featuring 64 and 128 filters, excels at extracting spatial features from sequential input, identifying localized patterns within the network traffic data. Max-pooling layers help reduce the dimensionality [29], while dropout layers with a 0.2 rate prevent overfitting. Finally, the fully connected layer uses ReLU as activation function and for binary classifications, it uses sigmoid layers.

C. Performance Evaluation

To measure the overall performance to comprise as using a measure of model accuracy, precision, F1 score. Also determine the confusion matrix based on TPR and FPR (True positive rate and False Positive rate), and also represent through ROC curve. Accuracy is measured as below:

$$Acc = \frac{Tpos + Tneg}{Tpos + Tneg + Fpos + Fneg}$$

True positive, true negative, false positive, and false negative are represented as Tpos, Tneg, Fpos, and Fneg, respectively. The F1-score represents the harmonic mean between recall and precision [30]. It calculates as follows, providing a balance between recall and precision:

$$F_1 - score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

Precision is the ratio of true positive predictions and the total number of positive predictions. Here's showing the calculation:

$$Precision = \frac{Tpos}{Tpos + Fpos}$$

TABLE IV: Summary Performance of ML based model

Model	Accuracy	Precision	F1_Score
SVM	0.53	0.53	0.69
MultinomialNB	0.81	0.77	0.84
LRC	0.93	0.93	0.94
DT	**0.97**	**0.93**	**0.93**

TABLE V: Summary Performance of DL based model

Model	Accuracy	Precision	F1_Score
CNN	0.97	0.98	0.97
LSTM	0.97	0.97	0.97
GRU	0.98	0.97	0.98
BiLSTM	**0.98**	**0.98**	**0.98**

True positive expressed as Tpos and, false positive expressed as Fpos.

IV. Experimental Analysis

All the selected ML and DL classifiers tested with 25192 instances which are fully randomized from KDDcup99 datasets. The classifier evaluated based on accuracy, precision, F1-score, and ROC curve. Result are acquired based on running code upon the ML classifiers using Python language and platform using google co-laboratory [31].

A 20-fold cross-validation applied for evaluation of the model. In 20-fold cross-validation, all the features divided into 15 particular features for better score performance. Comparative analysis and results of ML and DL model classifiers are shown respectively in Table IV and Table V. Among the ML model classifiers Decision Tree classifier performs best with 97% accuracy rate, 98% precision, and 97% F1 score and SVM outperforms lowest 53%, 53%, 69% rate with accuracy, precision, F1 score respectively.

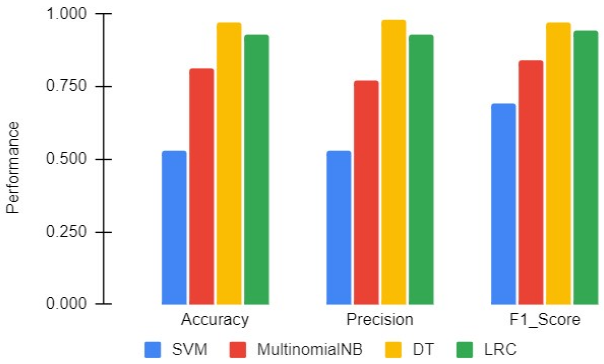


Fig. 3: Performance of ML based model

Subsequently, Multinomial Naive Bayes(NB) and Linear Regression Classifier(LRC) both perform quite good upon test dataset. MultinomialNB performs with 81% accuracy rate, 77% precision, and 84% F1 score while LRC performance

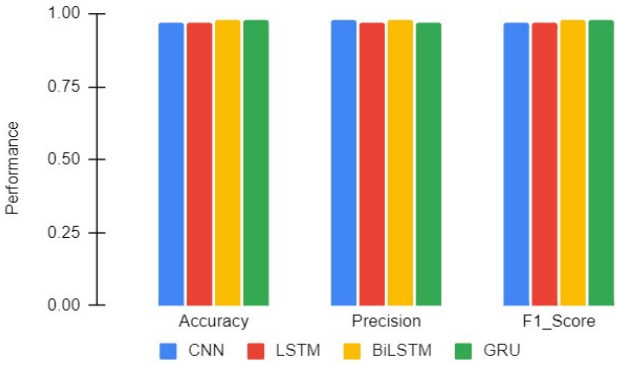


Fig. 4: Performance of DL based model

improved as lowest 93%, 93%, 94% rate with accuracy, precision, F1 score respectively. Figure 3 represents comparison among accuracy, precision, F1 score of ML based models and Figure 5 illustrates the ROC analysis curve of ML models.

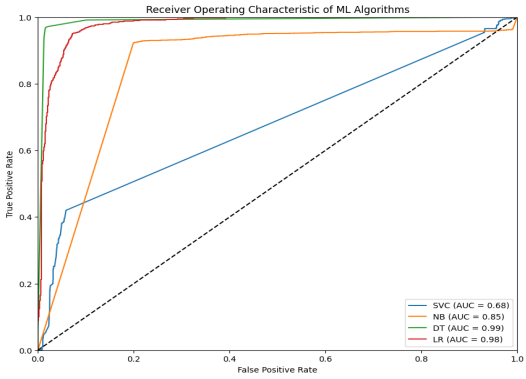
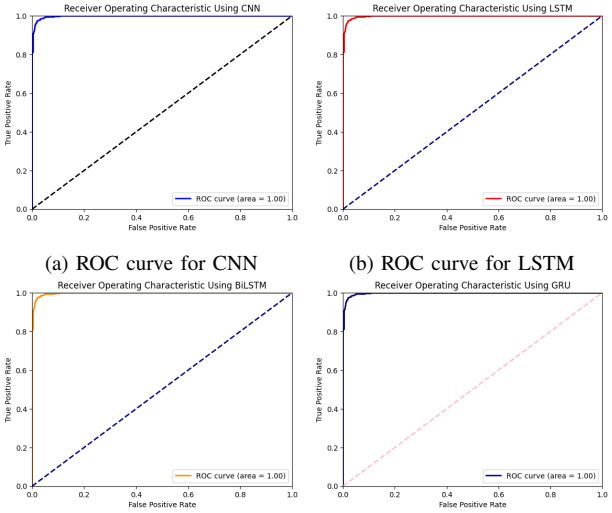


Fig. 5: Performance of ML based model

Conversely, Every DL model provides more than or equal 97% accuracy, whereas slight difference between precision and F1 score between CNN, LSTM, BiLSTM, and GRU four DL models. A comparative analysis showed on TableV based on accuracy, precision, F1 score measure using DL models. We observed that the BiLSTM and GRU both perform quite good upon test dataset. In Figure4 and Figure6 represent the comparison among the classifiers based on Precision, recall and F1-score, again BiLSTM performed highest in the all measure Precision (98%), recall (98%), F1-score(98%) while LSTM is the lowest in terms of Precision (97%), recall (97%), F1-score(97%).

LSTM usually processes in one direction. It has access to the past information only in each time steps. On the other hand, BiLSTM processes in both directions forward and backward, it can capture both past and future data points. GRU is a simplified version of LSTM, it takes less computational power and quicker to run. In context of efficiency and training time GRU performs better, In our dataset BiLSTM outperformed GRU and LSTM, in capturing both past and future dependencies

As summarized, based on all the performance measures, it



(a) ROC curve for CNN (b) ROC curve for LSTM

(c) ROC curve for BiLSTM (d) ROC curve for GRU

Fig. 6: Comparing ROC curve of selected DL model

is clear that all the ML and DL models performed significantly well upon this dataset. The performance analysis showed significant value count (accuracy, Precision, F1-Score and ROC curve) for each ML and DL models that gives a realization for dataset and features. Though there is a subtle difference between them, all models significantly perform well on the dataset. The above illustration observed both BiLSTM and GRU working better compared to the other models. Finally, based on all the performance measures, it was evident that the BiLSTM is the best classifier for the dataset used.

The results show that All DL models provide better Accuracy, Precision, F1 score in this analysis. From these results and analysis, DL models performs better than ML models and provides better accuracy than ML models.

V. Future Work

Our Future work will focus on multiple datasets testing, additionally in big data. We can investigate more robust data preprocessing techniques before applying more sophisticated algorithms and in-depth learning approach. It can help focus to detect on rare class attack and zero-day attack in initial attempts by using DL, select more normalization function based on features distribution and in depth conceptual model of the particular form of attack. Also, there is an advance research scope to test recent data-driven technology like Explainable Artificial Intelligence (XAI) to explain more the technology used. XAI can interpret the models through examination of the relative performance of DL and supervised ML models.

VI. Conclusion

We performed a comprehensive study on how certain algorithms are represented on the KDD99dateset. The comparison of algorithms was evaluated using measures of accuracy, precision, F1-score, as well as sensitivity (TPR), and the rate of false positives (FPR). Nevertheless, even with an unevenly

distributed data set and only specific features employed, we were still able to use all those techniques and produce a meaningful result that will be useful to the diligent researcher. Following competitive study, we discovered that, out of all the performance indicators, DL algorithms, particularly BiLSTM and GRU, perform best on the KDD99 dataset. These makeshift algorithms could be able to create effective network interference monitoring devices that an organization can manage for security reasons. Finally, we conclude that a centralized and distributed IDS system based On deep learning framework allows increased reliability, efficiency, and availability.

REFERENCES

[1] T. Rupa Devi and S. Badugu, "A review on network intrusion detection system using machine learning," in *International Conference on E-Business and Telecommunications*, pp. 598–607, Springer, 2019.

[2] J. Cannady, J. Harrell, *et al.*, "A comparative analysis of current intrusion detection technologies," in *Proceedings of the Fourth Technology for Information Security Conference*, vol. 96, 1996.

[3] M. Islam, *Contextual Gamification Platform based Big5 Personality Trait and Preference Selection for User Recommendation*. PhD thesis, UIU, 2024.

[4] M. F. A. Sayeedi, J. F. Deepti, A. M. I. M. Osmani, T. Rahman, S. S. Islam, and M. M. Islam, "A comparative analysis for optimizing machine learning model deployment in iot devices," *Applied Sciences*, vol. 14, no. 13, p. 5459, 2024.

[5] R. Patel, A. Thakkar, and A. Ganatra, "A survey and comparative analysis of data mining techniques for network intrusion detection systems," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2, no. 1, pp. 265–260, 2012.

[6] M. Islam, H. M. M. Jamil, S. A. Pranto, R. K. Das, A. Amin, and A. Khan, "Future industrial applications: Exploring lpwan-driven iot protocols," *Sensors*, vol. 24, no. 8, p. 2509, 2024.

[7] K. Cup, "http://kdd. ics. uci. edu/databases/kddcup99/kddcup99. html," *The UCI KDD Archive*, 1999.

[8] Z. Azam, M. M. Islam, and M. N. Huda, "Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree," *IEEE Access*, 2023.

[9] J. J. Tanimu, M. Hamada, P. Robert, and A. Mahendran, "Network intrusion detection system using deep learning method with kdd cup'99 dataset," in *2022 IEEE 15th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*, pp. 251–255, IEEE, 2022.

[10] M. Imran, N. Haider, M. Shoaib, I. Razzak, *et al.*, "An intelligent and efficient network intrusion detection system using deep learning," *Computers and Electrical Engineering*, vol. 99, p. 107764, 2022.

[11] A. Shokeen, N. Yadav, and V. Sisaudia, "Performance analysis of different machine learning algorithms for intrusion detection on kdd-cup-99 dataset," in *AIP Conference Proceedings*, vol. 3072, AIP Publishing, 2024.

[12] A. Meliboev, J. Alikhanov, and W. Kim, "Performance evaluation of deep learning based network intrusion detection system across multiple balanced and imbalanced datasets," *Electronics*, vol. 11, no. 4, p. 515, 2022.

[13] M. B. Umair, Z. Iqbal, M. A. Faraz, M. A. Khan, Y.-D. Zhang, N. Razmjooy, and S. Kadry, "A network intrusion detection system using hybrid multilayer deep learning model," *Big data*, 2022.

[14] S. Norwahidayah, N. Farahah, A. Amirah, N. Liyana, N. Suhana, *et al.*, "Performances of artificial neural network (ann) and particle swarm optimization (pso) using kdd cup '99 dataset in intrusion detection system (ids)," in *Journal of Physics: Conference Series*, vol. 1874, p. 012061, IOP Publishing, 2021.

[15] M. Sarnovsky and J. Paralic, "Hierarchical intrusion detection using machine learning and knowledge model," *Symmetry*, vol. 12, no. 2, p. 203, 2020.

[16] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the jam project," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, vol. 2, pp. 130–144, IEEE, 2000.

[17] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert systems with applications*, vol. 39, no. 1, pp. 424–430, 2012.

[18] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, pp. 1–6, Ieee, 2009.

[19] X. Du, H. Xu, and F. Zhu, "Understanding the effect of hyperparameter optimization on machine learning models for structure design problems," *Computer-Aided Design*, vol. 135, p. 103013, 2021.

[20] A. Pathak, S. Pathak, *et al.*, "Study on decision tree and knn algorithm for intrusion detection system," *International Journal of Engineering Research & Technology*, vol. 9, no. 5, pp. 376–381, 2020.

[21] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," in *2017 IEEE 15th international symposium on intelligent systems and informatics (SISY)*, pp. 000277–000282, IEEE, 2017.

[22] B. Sharmila and R. Nagapadma, "Intrusion detection system using naive bayes algorithm," in *2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, pp. 1–4, IEEE, 2019.

[23] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," *Procedia Computer Science*, vol. 89, pp. 117–123, 2016.

[24] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," *Procedia Computer Science*, vol. 171, pp. 1251–1260, 2020.

[25] H. Wang and D. Hu, "Comparison of svm and ls-svm for regression," in *2005 International conference on neural networks and brain*, vol. 1, pp. 279–283, IEEE, 2005.

[26] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[27] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[28] S. Mangal, P. Joshi, and R. Modak, "Lstm vs. gru vs. bidirectional rnn for script generation," *arXiv preprint arXiv:1908.04332*, 2019.

[29] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[30] J. Miao and W. Zhu, "Precision–recall curve (prc) classification trees," *Evolutionary intelligence*, vol. 15, no. 3, pp. 1545–1569, 2022.

[31] M. J. Ayan, "Google colab link," 2024.