Anomaly Based Intrusion Detection System: A Deep Learning Approach

Sourou Tossou
Department of Computer Science and Information Technology
University of the District of Columbia
Washington DC, USA
sourou.tossou@udc.edu

Miftahul Qorib
Department of Science and Mathematics
University of the District of Columbia
Washington DC, USA
miftahul.qorib@udc.edu

Thabet Kacem
Department of Computer Science and Information Technology
University of the District of Columbia
Washington DC, USA
thabet.kacem@udc.edu

***Abstract*—In recent years, computer networks have seen a considerable proliferation in terms of performance and total traffic volume. At the same time, cyber attacks have been on the rise ever since, which led to the emergence of Intrusion Detection Systems (IDSs) to deal with them. Conversely, artificial intelligence has been a popular technique that can be applied to a variety of purposes including detection of cyber attacks. However, most related work that leveraged artificial intelligence classifiers to address this problem used outdated datasets. In this paper, we implemented an anomaly-based intrusion detection system using deep learning algorithms with the goal of achieving higher performance while using a newer dataset. That is why we used the NSL-KDD dataset, which constitutes an improvement over the widely used KDD Cup 99 dataset, as it addresses some of its imperfections such as duplicated records and obsolete attack types. Then, we developed three deep learning classifiers that are Recurrent Neural Networks (RNNs), Multi-Layer Perceptron (MLP), and a hybrid Convolutional Neural Network-Long Short Term Memory (CNN-LSTM) model. Also, we compared the effectiveness of our proposed model with Machine Learning classifiers such as Support Machine Vector (SVM), K-Nearest Neighbor (KNN), and Gradient Boosting (GB). Finally, we validate our findings with a performance evaluation of our model, which showed encouraging results.**

***Index Terms*—Intrusion Detection System, Deep Learning, Anomaly Based Detection System, Network Security**

I. Introduction

The amount of data generated every day is truly tremendous. At our current rate, over 2.5 quintillion bytes of data are created every day, but with the growth of the Internet of Things (IoT), the pace will only keep accelerating [16]. However, with this exponential growth of data comes a critical challenge which is to guarantee data privacy and confidentiality. Consequently, security experts have been designing approaches to make computer networks more resilient to cyber attacks. In this context, IDS constitutes one of the most popular solutions. It has been materialized and discussed by many researchers who came up with various implementation techniques, varying from machine learning to deep learning classifiers [6].

There are two types of IDSs: Anomaly-based Intrusion Detection Systems (A-IDSs) and Signature-based Intrusion Detection Systems (S-IDSs). An A-IDS aims to detect the intrusions locally and at the network level. It operates by monitoring the system activity before making an assessment. Conversely, an S-IDS can only detect attacks for which a signature has been previously encountered. While A-IDS's classification is based on heuristics or rules, S-IDS is based on patterns or signatures to detect misuses that fall out of the normal system operation.

The popularity of A-IDSs stems from the fact that they address the main weakness of S-IDSs as it is not able to detect any attack without any prior known pattern or signature. An A-IDS may use machine learning to create an accurate model of the normal network activity. Then, if it detects a deviation from this model, it interprets it as an intrusion. A-IDSs that are designed using machine learning involve two phases: a training phase and a testing phase. In the training phase, normal operations are fed into the model to learn the normal behavior, then in the testing phase, programmers evaluate the trained model by feeding it anonymous data that the model classifies it as either an intrusion or as a normal operation [13].

When using a deep learning architecture, unsupervised features that are learned using machine learning capabilities separate normal data from abnormal one using a classifier. Classification tables are created by modeling the current data and there are several techniques to build them including data mining, expert systems, pattern processing, and statistical methods. Conversely, artificial intelligence techniques are being developed to deal with sophisticated attacks that are difficult to detect. The classification process uses techniques such as fuzzy logic, artificial neural networks, support vector machines, artificial immune system, and genetic algorithms [5].

In this paper, we developed an A-IDS approach that relies on three types of deep learning classifiers, namely, RNN, MLP, and CNN-LSTM. We decided to use the NSL-KDD dataset since it is newer compared to the KDD CUP 99 dataset, and it has more variety of features and up-to-date anomalies. We validate our findings with a performance evaluation of our model. The obtained results show potential of success.

The rest of the paper is organized as follows. Section II

This work was funded by NSF award 2011689

sheds the light on the related work. Section III describes the methodology we followed to design our solution. Section IV presents the results we obtained in the conducted experiments. Section V concludes the paper and discusses our future work.

II. Related Works

Abraham and Bindu [6] proposed a machine-learning approach for intrusion detection by reviewing the performance of the different machine-learning classification methods such as AdaBoost, Extra Trees, Gradient Boost, Linear Regression, MLP, and Random Forest on the DARPA dataset. They obtained a better performance with the Gradient Boost classifier.

In their study of the subject, Jisna et al. [11] proposed and evaluated a cloud-based deep learning intrusion detection system, combining a Stacked Contractive Auto-Encoder (SCAE) model and a support vector machine (SVM) model. This technique allowed them to build a model that simultaneously detects and classifies attacks. They tested their model on two well-known intrusion detection datasets, KDD Cup 99 and NSL-KDD. Their evaluation results showed that their hybrid model performed better than SVM but as good as an LSTM model, however, it needed longer training time.

Altunay et al. [5] proposed an analysis technique of the anomaly-based intrusion detection systems in the SCADA networks. Their approach consists of applying CNN, Auto-Encoders, Deep Convolutional Networks, LSTM, or combinations of these different methods. Their findings indicate that combining Deep Belief Networks (DBN) and Extreme Learning Machine (ELM) outperformed the other models. This was attributed to the DBN ability to shorten the training and testing periods, in addition to enabling the analysis of temporal attack models while the ELM helped to reveal the connections between the hidden and output nodes.

Albelwi [16] proposed an IDS approach based on Multi-Task Learning (MTL) for attack detection. His approach relies on deep neural networks (DNNs) and Deep Multi-Layer Perceptron (DMLP) models while the data from UNSW-NB15 and CICIDS2017 datasets was combined into one feature vector. Their findings indicate that his model outperformed some other models, such as neural networks and decision trees.

Li [7] proposed a CNN-BiLSTM intrusion detection model for complex system networks. The proposed model performed data re-sampling on some unbalanced data, reducing the class data variance. The proposed model leverages CNN's ability to extract deep local features to extract the sampled data effectively. Then, it uses a bidirectional LSTM network to learn and extract the correlation features between the continuous data in both positive and negative directions. The model comprehensively considers the time and space correlation of the intrusion data. It can also mine the unknown features and internal dependencies between the data, thus reducing the false alarm rate and improving the detection effect.

Dong et al. [19] proposed to use the KDD CUP 99 dataset as experimental data to apply the Auto-Encoder Network (AE Network) method and the AE-AlexNet classifier. Both strategies fall under the unsupervised multi-layer learning algorithm category, but the AE-AlexNet model has a higher detection rate of attacks.

Amutha et al. [14] proposed a hybrid system for network intrusion detection that combines a recurrent neural network (RNN) model and a Long short-term memory (LSTM) model on the UNSW-NB18 dataset. Their model is based on a detection engine that loads the trained model into the file system, inputs network data, runs the model for detection, and outputs the detection results. During its implementation, the Tanh function capacities helped them initiate the cell states while the sigmoid actuation capacities helped to activate nodes in the LSTM model, which allowed them to have a 95% of accuracy.

III. Methodology

We used the NSL-KDD dataset, which comprises 125971 records weighing 19 MB in its KDDTrain+ set and 22542 logs weighing 4 MB in its KDDTest+ batch. Unlike Jisna et al. who used all components of the NSL-KDD dataset in their approach, we decided to just focus on these two sets to avoid any redundancy. This dataset is often used in intrusion or divergence detection systems to evaluate different artificial intelligence approaches against cyber threats. It contains a set of specific data that can be trained and tested to ensure the best system accuracy on the various deep learning algorithms that we implemented [18]. In this section, we will first describe the methodology overview before describing the NSL-KDD dataset and explaining the technical details of our deep learning model.

A. Methodology Overview

Our methodology overview is depicted in Fig. 1. First, we downloaded the data from a NSL-KDD dataset repository. Since our dataset came with missing components, we performed some data cleaning and pre-processing by assigning the different feature names on the records. Next, we split our data into two sets: training and testing. Finally, we built three types of deep-learning algorithms before evaluating them through performance metrics. In the next subsections, we will shed the light on the technical details of our methodology.

B. NSL-KDD Dataset

The NSL-KDD dataset [9] was created in 2009 by Tavallaee, Bagheri, Lu, and Ghorbani, researchers at the University of New Brunswick, Canada. The dataset was designed as an improvement over the original KDD Cup 1999 dataset as it addressed some of its main shortcomings, such as the presence of redundant records and outdated attack types. The NSL-KDD dataset includes both normal and anomalous connections, with four different types of labeled attacks. The dataset has been widely used as a benchmark for intrusion detection research and has contributed to developing new and improved systems. The main differences that NSL-KDD has over the original KDD Cup 99, as highlighted in [4] and [9], can be summarized as follows:

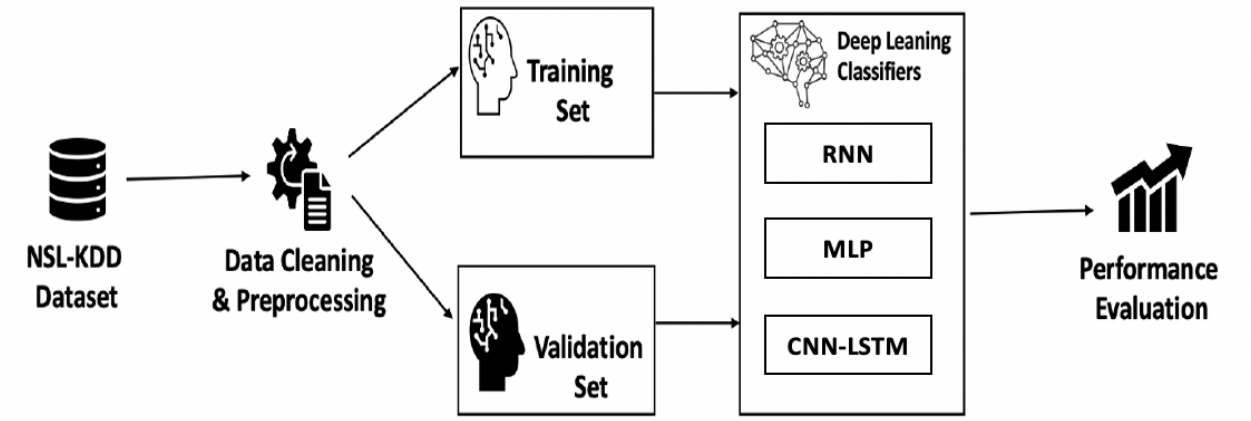

Fig. 1. Methodology Overview

- The classifier does not give biased results because the training set has no redundant data.
- The reduction ratio is lower because the test set has no repetitive data.
- The number of selected records from each difficulty-level group is inversely proportional to the percentage of records in the original KDD data set. As a result, the classification rates of distinct machine learning methods vary in the broader range, making it more efficient to evaluate different learning techniques accurately.
- The number of records in the train and test sets is reasonable, making it affordable to run the experiments on the complete set without randomly selecting a small portion. Consequently, the evaluation results of different research works would be consistent and comparable.

In both KDDTrain+ and KDDTest+ sets, there are 41 attributes that unfold different flow features (duration, protocol type, service, etc.) and an additional attribute label called "attack" that evaluates an attack type or a normal activity. The four attack categories are further grouped as Denial of Service Attack (DOS), Probing Attack (Probe), Remote to Local Attack (R2L), and User to Root Attack (U2R), affecting mostly the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP), as shown in Fig. 2. It also represents the percentage of the attacks compared to normal activities on the network (53% vs. 43%), which gives enough attack records to test different models.

C. Deep Learning Classifiers

Our initial hypothesis was that using deep learning models such as CNN-LSTM, RNN, and MLP, we would be able to achieve better performance results in predicting cyber attacks on computer networks than the existing approaches that worked on the same NSL-KDD dataset.

*1) **RNN Classifier**:* RNNs learn the data through a sequential process. This sequential process is justified as they retain memory to store the previous outcome before processing the current sequence. It is known as recurrent because the prior output of each time step is used as input to the next time step. So, remembering the previous time step's output is important in the whole process. This allows the neural network to learn the long-term dependencies in the training data [9]. The sequential process for each time step is formulated in equation (1) and equation (2) as follows [10]:

$$S_t = f\left(W_{sx,x_t} + W_{ys,s_{t-1}} + b_s\right) \quad (1)$$

$$Y_t = g\left(W_{ys,s_{t-1}} + b_y\right) \quad (2)$$

Where f and g are the encoder and decoder functions respectively, $W_{sx,xt}$ represents the current inputs, $W_{ys,st-1}$ is the previous output at time step t, and b_s represents the bias.

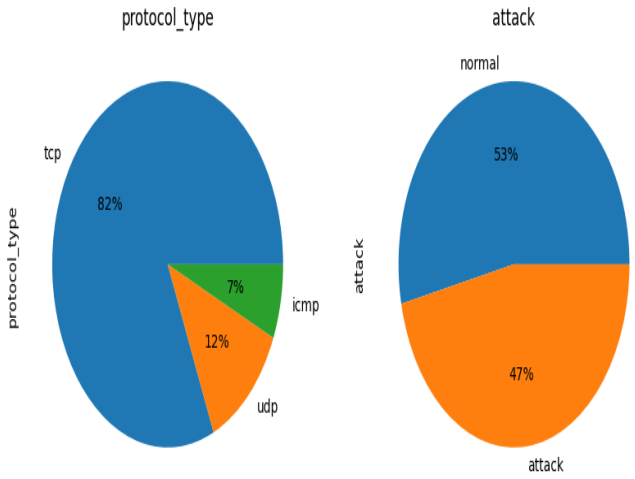

Fig. 2. Representation of the Attacks and Protocols

*2) **CNN-LSTM Classifier**:* CNN-LSTM model for intrusion detection consists of a set of CNN and LSTM layers that extract complex characteristics from the dataset and store complex irregular trends.

The CNN layer is well known for its capability of extracting local features from input layers and transforming them into

more complex ones. The LSTM layer represents the bottom layer of the proposed model, which stores the time information about the most dominant characteristics of the intrusion detection system extracted by the upper CNN layer.

The CNN-LSTM final layer is formed by fully connected layers, utilized to detect intrusions over certain periods of time. The output of the LSTM unit is flattened into a feature vector $h^l = h_1, h_2, ...h_l$, where l represents the number of units in LSTM. The equation (3) below represents the equation deployed at that level.

$$d_i^l = \sum_j w_{ji}^{l-1}(\sigma(h_i^{l-1}) + b_i^{l-1}) \tag{3}$$

Where σ is a non-linear activation function, w is the weight of the i^{th} node for layer $l-1$ and j^{th} node for layer l, and b_i^{l-1} represent the bias [1].

*3) **Multi-Layer Perceptron Classifier**:* MLP employs a supervised machine learning technique known as back-propagation. It is a feed-forward artificial neural network with several layers and a nonlinear activation function. The multilayer perceptron architecture includes a minimum of three layers: an input layer, an output layer, and a hidden layer. The input layer processes the received input signal, the output layer produces given outputs for the program, and the hidden layer is present between the input and output layers, where artificial neurons take in weighted inputs and generate a result through an activation function. It is done through the following formulas [2]:

$$y = \phi(\sum_{i=1}^{n} w_i \mathbf{X} + b) \tag{4}$$

$$y = \phi(W^T \mathbf{X} + b) \tag{5}$$

Where ϕ is the activation function, w is the weights, X is the input data, b is the bias, and y is the output.

IV. Experimental Results

A. Machine Learning Classifiers

To evaluate the efficiency of our model, we also implemented some machine learning classifiers and compare them with our deep-learning classifiers. These machine learning classifiers are described as follows.

*1) **Support Machine Vector Classifier**:* The Support Machine Vector Classifier [8] is a set of related supervised learning methods used for classification and regression. It is a classification and regression prediction method which uses machine learning theory to optimize the predictive accuracy while avoiding over-fitting of the data. The SVM [17] creates the best line/decision boundary that splits n-dimensional space into classes, making it easy to insert new data points in the right group for the future. The hyperplane is known as the most optimal boundary. The SVM method chooses extreme vectors, making hyperplane creation easier.

*2) **K-Nearest Neighbor Classifier**:* KNN is a supervised learning algorithm that does not need any training data points for modeling. KNN attempts to predict the proper class for the testing dataset by measuring the distance between the testing dataset and all the training points. Then, it chooses the K number of points closer to the testing data. We calculate the distance using Euclidean formula. The KNN method determines the probability of the testing data belonging to 'K' training data classes. The class with the most significant likelihood is selected [3]. The Euclidean distance function is:

$$D(x, y) = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{6}$$

*3) **Gradient Boosting Classifier**:* The Gradient Boosting classifier utilizes a sequence of decision trees by progressively having trained trees. It can be used for both classification and regression. Each stage involves the creation of a new decision tree based on the preceding decision tree's faults, which aids in reducing errors. Proper techniques for assessing and training a dataset with a gradient-boosting classifier take more time and storage. The equations below show the process [12]:

$$\mathbf{f_0}(\mathbf{x}) = \arg\min_{\gamma} \sum_{i=1}^{n} \mathrm{L}(\mathrm{y_i}, \gamma) \tag{7}$$

$$\mathbf{L} = \frac{1}{n} \sum_{i=0}^{n} (\mathrm{y_i} - \gamma_i)^2 \tag{8}$$

Where L is our loss function, γ is our predicted value, y_i is the observed value, and $argmin$ means we have to find a predicted value/gamma for which the loss function is minimum.

B. Results

Using the combination of the KDDTrain+ and KDDTest+ within the NSL-KDD dataset, we divided the dataset into two parts, 80% for training and 20% for testing the different intrusion detection models that we built through a Python script.

To show the effectiveness of our models, we decided to compare them using the different performance evaluators, such as the Accuracy rate, the Sensitivity rate, and the Precision rate, as we can see in table I. The accuracy rate is the number of correct predictions divided by the total number of instances in the dataset [15] using the following formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{9}$$

The sensitivity measures how well a machine learning model can detect positive instances [15] by using this formula:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

Finally, the precision, which is also known as positive predictive value, is the ratio of the number of true positives to the total number of positives detected by the model [15].

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

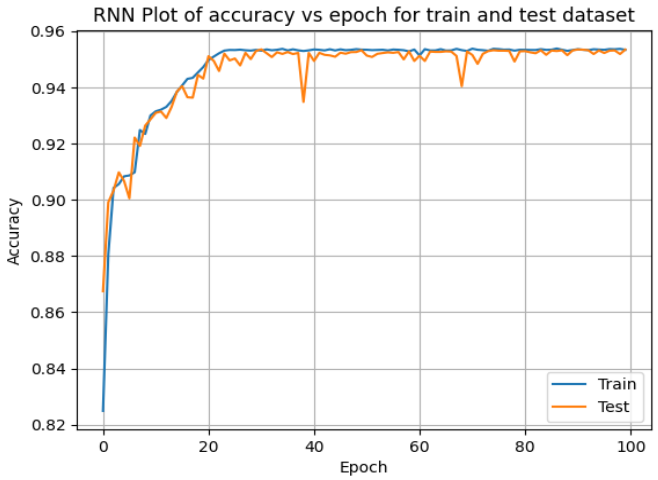

Fig. 3. RNN Accuracy Plot.

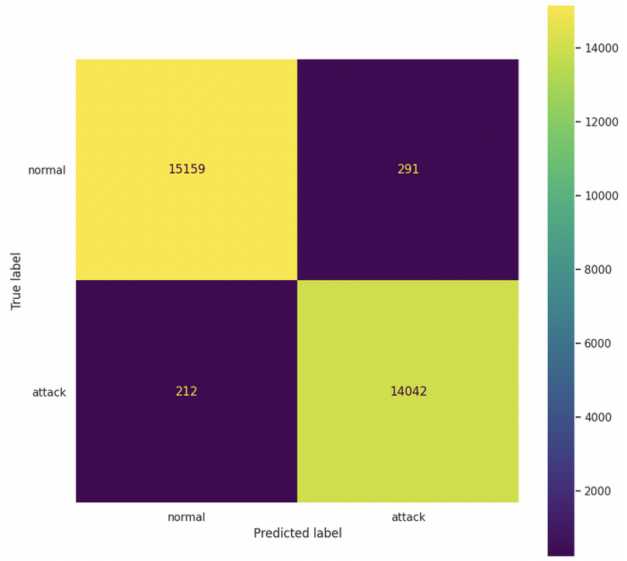

Fig. 4. MLP Confusion Matrix.

RNNs can be used for anomaly detection, where they learn the normal behavior of a system and flag deviations from that behavior as potential intrusions. Also, some intrusions or attacks may exhibit long-term dependencies, where actions taken at one time step affect subsequent steps. RNNs can capture such dependencies better than models that do not consider the sequence of events. Fig. 3 shows the resulting performance of the RNN classifier in our intrusion detection system. As it can be observed, we have achieved an accuracy rate of 96% and a data loss of 4%. During its implementation, three hidden unit layers were used, and 100 epochs were executed while training and testing, and each epoch took approximately 3-4 seconds of the proposed method. The experimental result of our RNN model shows how the algorithm learns the normal behavior of a system and flags deviations from that behavior as potential intrusions.

Since the NSL-KDD is a tabular dataset, we chose the MLP model because of its versatility. It can handle a wide range of data types and automatically learn relevant feature representations from the data, so we don't need to perform extensive manual feature engineering. As Fig. 4 portrays, the confusion matrix of the MLP model allowed us to record 15159 for the True Positives, 291 for the False Positives, 212 for the True Negatives, and 14042 for the False Negatives. By applying the different formulas for accuracy, sensitivity, and precision, we were able to get the accuracy rate of the MLP model evaluated at 98.3%.

Our best performance came from the CNN-LSTM algorithm. The NSL-KDD dataset contains both spatial (static features) and temporal (sequential or dynamic features) information. We then decided to use the CNN-LSTM model because it is designed to capture spatial and temporal patterns effectively. CNNs excel at spatial feature extraction, while LSTMs are well-suited for modeling sequential data. This combination can be beneficial for intrusion detection, where you need to consider both static and dynamic aspects of the network traffic. As shown in Fig. 5, our CNN-LSTM has outperformed the other algorithms with an accuracy rate of 99.3% and a data loss of only 0.7%. In order to implement this model, we used a kernel size of 5, 64 filters, a pool size of 4, and 100 epochs, where each epoch ran for approximately 12 seconds of the proposed method.

In this paper, we wanted to prove that Deep Learning algorithms are more reliable and more robust than simple Machine Learning in A-IDSs while using a newer dataset. Our hypothesis was validated via the evaluation results. As shown in Table I, the accuracy of the ML models that we built was 93.83% for SVM, 98.32% for KNN, and 98.81% for GB. These results prove that none of these models has a better accuracy rate than our hybrid CNN-LSTM model.

The resulting performance evaluation of our model is encouraging, which proves that our hypothesis was correct. Yet, other researchers might have used different datasets or models

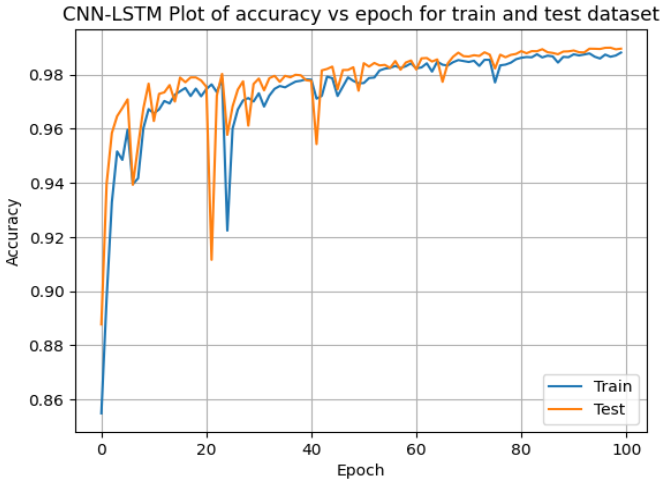

Fig. 5. CNN-LSTM Accuracy Plot.

TABLE I
TABLE OF COMPARISON OF MODELS.

Models	Accuracy	Sensitivity	Precision
SVM	93.83%	92.96%	94.11%
RNN	95.6%	95.6%	95.1%
KNN	98.32%	97.93%	98.55%
MLP	98.30%	98.51%	97.96%
GB	98.81%	98.65%	98.81%
CNN_LSTM	99.3%	99.0%	99.5%

to test their intrusion detection system, but their results are either lower or slightly similar to ours. Therefore, considering the high performance of our detection system, especially the CNN-LSTM model, we believe that our approach holds the promise of constituting a reliable option to mitigate cyber threats targeting computer networks.

V. CONCLUSION

As intrusion detection systems continue to play a vital role in preventing cyber attacks on computer networks, their effectiveness depends directly on the decision engines that are being used. While signature-based detection systems could become quickly obsolete when they encounter an unknown attack, anomaly-based intrusion detection systems constitute a more viable option as they leverage heuristics. Conversely, artificial intelligence, which is being applied to various fields nowadays, holds a promising potential in revolutionizing such systems thanks to the power that machine learning and deep learning classifiers hold. There have been several related proposed solutions that consider older datasets, such as the KDD CUP 99, which has numerous issues that may put their results in question.

Being driven by the goal of achieving better performance while using a newer dataset, this paper proposed an anomaly-based intrusion detection system based on deep learning classifiers, namely RNN, MLP, and CNN-LSTM. We used the NSL-KDD dataset, which is more reliable than the traditionally used datasets. Our findings were supported by our evaluation results thus validating our initial hypothesis. In our future work, we plan to use live datasets and more advanced models to secure networks in real-time.

REFERENCES

[1] A. Agga, A. Abbou, M. Labbadi and Y. el Houm, "Short-Term Load Forecasting: Based on Hybrid CNN-LSTM Neural Network," 2021 6th International Conference on Power and Renewable Energy (ICPRE), Shanghai, China, 2021, pp. 886-891, doi: 10.1109/ICPRE52634.2021.9635488.

[2] A. Ahmad et al., "Vehicle Recognition using Multi-Layer Perceptron and SMOTE Technique," 2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH), Riyadh, Saudi Arabia, 2022, pp. 190-193, doi: 10.1109/SMARTTECH54121.2022.00049.

[3] B. Nuralamsyah, S. R. Anggraeni, L. Awwabi, N. A. Ranggianto, H. Studiawan and A. M. Shiddiqi, "Performance Analysis Between EOTI-K-Means++, EOTI, and KNN for Brute Force Detection System," 2022 10th International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 2022, pp. 53-58, doi: 10.1109/ICoICT55009.2022.9914878.

[4] G. Karatas, O. Demir and O. Koray Sahingoz, "Deep Learning in Intrusion Detection Systems," 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), Ankara, Turkey, 2018, pp. 113-116, doi: 10.1109/IBIGDELFT.2018.8625278.

[5] H. C. Altunay, Z. Albayrak, A. N. Özalp and M. Çakmak, "Analysis of Anomaly Detection Approaches Performed Through Deep Learning Methods in SCADA Systems," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2021, pp. 1-6, doi: 10.1109/HORA52670.2021.9461273.

[6] J. A. Abraham and V. R. Bindu, "Intrusion Detection and Prevention in Networks Using Machine Learning and Deep Learning Approaches: A Review," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 2021, pp. 1-4, doi: 10.1109/ICAECA52838.2021.9675595.

[7] J. Li, "Network Intrusion Detection Algorithm and Simulation of Complex System in Internet Environment," 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2022, pp. 520-523, doi: 10.1109/ICIRCA54612.2022.9985720.

[8] M. Pradhan, S. Mohanty and A. O. Seemona, "Machine Learning-Based Intrusion Detection System for the Internet of Vehicles," 2022 5th International Conference on Computational Intelligence and Networks (CINE), Bhubaneswar, India, 2022, pp. 1-6, doi: 10.1109/CINE56307.2022.10037357.

[9] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 2009, pp. 1-6, doi: 10.1109/CISDA.2009.5356528.

[10] M. W. Nadeem, H. G. Goh, Y. Aun and V. Ponnusamy, "A Recurrent Neural Network based Method for Low-Rate DDoS Attack Detection in SDN," 2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS), IPOH, Malaysia, 2022, pp. 13-18, doi: 10.1109/AiDAS56890.2022.9918802.

[11] P. Jisna, T. Jarin and P. N. Praveen, "Advanced Intrusion Detection Using Deep Learning-LSTM Network On Cloud Environment," 2021 Fourth International Conference on Microelectronics, Signals & Systems (ICMSS), Kollam, India, 2021, pp. 1-6, doi: 10.1109/ICMSS53060.2021.9673607.

[12] P. V. Reddy and S. M. Kumar, "A Method for Determining the Accuracy of Stock Prices using Gradient Boosting and the Support Vector Machines Algorithm," 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2022, pp. 1596-1599, doi: 10.1109/ICOSEC54921.2022.9952143.

[13] R. Hamad, L. Yang, W. Woo and B. Wei, "Joint Learning of Temporal Models to Handle Imbalanced Data for Human Activity Recognition.," Applied Science, pp. 10(15), 5293; https://doi.org/10.3390/app10155293, 30 30 2020.

[14] S. Amutha, K. R, S. R and K. M, "Secure network intrusion detection system using NID-RNN based Deep Learning," 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2022, pp. 1-5, doi: 10.1109/ACCAI53970.2022.9752526.

[15] S. A. Afshine Amidi, "Recurrent Neural Networks cheatsheet," [Online]. Available: https://stanford.edu/ shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks. [Accessed 23 04 2023].

[16] S. A. Albelwi, "An Intrusion Detection System for Identifying Simultaneous Attacks using Multi-Task Learning and Deep Learning," 2022 2nd International Conference on Computing and Information Technology (ICCIT), Tabuk, Saudi Arabia, 2022, pp. 349-353, doi: 10.1109/ICCIT52419.2022.9711630.

[17] S. P. Mohanty, U. Choppali and E. Kougianos, "Everything you wanted to know about smart cities: The Internet of things is the backbone," in IEEE Consumer Electronics Magazine, vol. 5, no. 3, pp. 60-70, July 2016, doi: 10.1109/MCE.2016.2556879.

[18] S. Tamy, H. Belhadaoui, M. A. Rabbah, N. Rabbah and M. Rifi, "An Evaluation of Machine Learning Algorithms To Detect Attacks in Scada Network," 2019 7th Mediterranean Congress of Telecommunications (CMT), Fez, Morocco, 2019, pp. 1-5, doi: 10.1109/CMT.2019.8931327.

[19] Y. Dong, R. Wang and J. He, "Real-Time Network Intrusion Detection System Based on Deep Learning," 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2019, pp. 1-4, doi: 10.1109/ICSESS47205.2019.9040718.