National College of Ireland

Network Intrusion Detection System using CNN-LSTM Hybrid Network

MSc Research Project
MSc in Cybersecurity

Ajeeser Kokkali
Student ID: 20112491

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Ajeeser Kokkali

Student ID: 20112491

Programme: MSc. In Cybersecurity **Year:** 2022

Module: Research Project

Lecturer: Vikas Sahni

Submission Due Date: …26/04/2022……

Project Title: Network Intrusion Detection System using CNN-LSTM Hybrid Network

Word Count: ……4143…… **Page Count:** ……18……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Ajeeser Kokkali

Date: 24-04-2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Network Intrusion Detection System using CNN-LSTM Hybrid Network

Ajeeser Kokkali
Student ID: 20112491

ABSTRACT

Most service providers are concerned about the rise in computer networks and internet assaults. It has prompted the development and use of intrusion detection systems (IDSs) to aid in the prevention or mitigation of network intruder threats. Intrusion detection systems have played and continue to play a critical role in detecting network attacks and anomalies over the years. Many IDSs have been proposed by researchers all around the world to address the threat of network intruders. Most of the previously proposed IDSs, on the other hand, have a high proportion of false alarms. This research introduces a novel approach for enhanced intrusion detection that uses a hybrid algorithm of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). NSL-KDD, a credible intrusion detection dataset that covers all typical, updated intrusions and cyberattacks, is used to evaluate DL-IDS. This bidirectional approach demonstrated the accuracy of 98.39 percent. Precision, false positive, F1 score, and recall were used to evaluate the algorithm's performance, and it was determined to be promising for deployment on live network infrastructure.

1. INTRODUCTION

Due to the fast expansion of the internet, it is vital to detect breaches that pose a security risk to networks. By analyzing patterns of collected data, the Intrusion Detection System (IDS) was suggested in 1980 to provide robust security for equipment against malicious software attacks, such as denial of service (DoS) [1]. IDS can identify attacks and deny or halt unauthorized traffic when it functions like a Denial-of-Service attack (DoS). In general, intrusion detection can be thought of as the solution to a classification problem. One of the issues with certain existing IDS is that they have a low detection accuracy. Another issue is that they rely on known attack signatures, making them incapable of detecting novel attacks.

Traditional machine learning approaches have been widely employed to distinguish several sorts of assaults in attempt to overcome these drawbacks [2]. The majority of traditional machine learning algorithms, on the other hand, are shallow learning methods that focus on feature engineering and selection. Furthermore, they typically are unable to give an effective solution for the big intrusion data categorization problem that is generated by a large volume of network application traffic [3]. Because big data frequently necessitates high-dimensional learning, shallow learning is unsuitable for analysis and forecasting. Deep learning, on the other hand, has the ability to extract better representations for better model creation. As a result, deep learning-based IDS are being developed by researchers in this sector.

This paper first looked at state-of-the-art IDS technologies that use machine learning techniques for identification. Simple machine learning algorithms, on the other hand, have

significant drawbacks, and security threats are on the rise. Upgraded learning approaches are needed, especially for feature extraction and intrusion analysis. Hinton [4] explains that deep learning has had considerable success in a variety of domains, including natural language processing, picture processing, and weather prediction. The approaches used in DL have a nonlinear structure that allows for greater learning for composite data analysis. The fast advancement in parallel computing in recent years has also resulted in a significant hardware foundation for DL approaches.

How can new approaches like Convolutional Neural Network-LSTM layers be used to boost the detection rate of intrusion detection systems or not?

CNN's unique architecture improves the quality of data representations. CNN is primarily used in the fields of image recognition and sentence modelling, but it has not been used in intrusion detection. The main objective of this paper is to present a CNN + LSTM-based deep learning strategy for intrusion detection. Furthermore, most existing models have trouble recognizing various attack types, particularly User-to-Root (U2R) and Remote-to-Local (R2L) attacks. The existing models appear to have a reduced detection accuracy for these two sorts of attacks. To address the aforementioned challenges, this study proposes a unique approach for increased intrusion detection that employs a hybrid algorithm of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). Finally, on the NSL-KDD dataset, tests were performed to classify several types of network traffic and compare it to the CNN-only model, RNN-only model and CNN-LSTM hybrid model. With a precision of 98.40 percent, this bidirectional technique achieved the highest known accuracy.

The rest of this article is structured as follows: background on IDS and related works are discussed in Section 2. The methodology for creating the proposed model is presented in Section 3, followed by detailed specification in Section 4, followed by an analysis of the implementation of the model using the deep learning algorithms in Section 5 and concluding with an evaluation and discussion of the experiment in Section 6.

2. RELATED WORK

The detection of intrusion in network systems has been a peculiar problem faced by most researchers. Since Denning introduced the first intrusion detection system in 1969, studies have applied multiple intrusion detection algorithms. Since traditional classification algorithms require manual feature extraction, deep learning techniques have proven to be an efficient way of combating intrusion. Deep learning (DL) in artificial intelligence (AI) mimics the human brain's functionality in the areas of data processing and the generation of patterns for effective decision making [5].

A new mix of Deep Learning termed Hybrid Deep Learning Network (HDLN) was built to catch code injection threats linked with the JavaScript code in the study [6]. The accuracy of this latter was judged on two levels: first, in relation to the number of hidden layers, filters, and neurons; the results showed that as the number of filters increases, accuracy increases; second, it was compared to other traditional classifiers; the marked accuracy was clearly the greatest. Finally, they stated that the accuracy of the previous work had been improved by this new model. The work put forward was commendable, except that the solution was restricted to injection attacks involving JavaScript code and did not address other types of assaults.

The authors introduce a new Deep Learning concept in their study [7], which combines Auto-Encoder with Deep Belief Network (DBN). The Auto-Encoder was tasked with reducing the dimensionality of data and identifying its key features, while the DBN was tasked with detecting the suspicious code. The new model recommendation was tested using the dataset KDD Cup 99, and the results were compared using simply a single DBN. The achievement has stated that the new process is far more accurate and consumes less time. The authors, however, did not explain why they chose to combine DBN and Auto-Encoder to create this hybrid.

Latah [8] suggested a five-stage hybrid classifier method to improve the detection rate against fraudulent traffic inside the network. The K-Nearest Neighbor method (KNN), Extreme Learning Machine (ELM) and Hierarchical Extreme Learning Machine are among the machine learning classifiers used in the model (H-ELM). The presented approach has an overall accuracy of 84.29 percent, with precision, recall, and F1-score percentages of 94.18, 77.18, and 84.83, respectively.

Kim et al. [9] used the KDD Cup'99 dataset to train the IDS and used the long short-term memory (LSTM) architecture to RNN. When compared to previous IDS classifiers, the LSTM-RNN IDS achieved a high level of accuracy while having a somewhat greater FAR. Despite the fact that their experiment yielded fantastic results, they only employed 1,630 records from NSL-KDD, which has 125,973 records. Meanwhile, they used the train set as the test set, implying that their experiment dataset may be biased.

Zhang et al. [10] developed deep hierarchical networks to detect network intrusions using current flow data from the CICIDS2017 and CTU datasets. The CNN + LSTM classification method had a 99.8% accuracy for CICIDS 2017 and a 98.7% accuracy for CTU data. The UNSW-NB15 data set obtained 97.49 percent accuracy using ML classification algorithms such as decision-making tree, SVM, RF, and Naïve.

The authors utilized a hierarchical strategy in another paper [11] by integrating two deep learning models, CNN and LSTM. The UNSW NB15 dataset was used for this project. On the obtained dataset, the authors usually use a multi-class classification algorithm. The author proposed a network detection system that could categorize NIDS detection assaults using deep learning models in another paper [12]. The authors suggested a technique for detecting assaults on networking systems and further classifying them using associated weights. The LSTM technique was used by the author to conduct consecutive operations on a stream of data. The author suggested using Auto encoders and the Principal Component Analysis approach in another work [13]. (PCA). This strategy could also be used to minimize the feature dimensions associated with the CICID2017 dataset. The resulting dimensions were then utilized to detect and classify harmful assaults in a networking system. This IDS-based innovative architecture classifies assaults based on previously recognized technological patterns. All of the networking nodes are contained within these packets.

3. RESEARCH METHODOLOGY

The IDS is the most essential security mechanism against complicated and large-scale network attacks, but its development is hampered by a lack of publicly available data. Many studies have used confidential data from a single organization or manually collected data to test IDS solutions, which has a negative impact on the reliability of their findings. In this paper, NSL-KDD dataset was used. This section provides an integrated strategy for improving network attack detection and response while lowering the proportion of false alarms. The proposed work is divided into the following categories, as shown in the diagram. The IDS is proposed, and it starts with the analysis of NSL-KDD data.

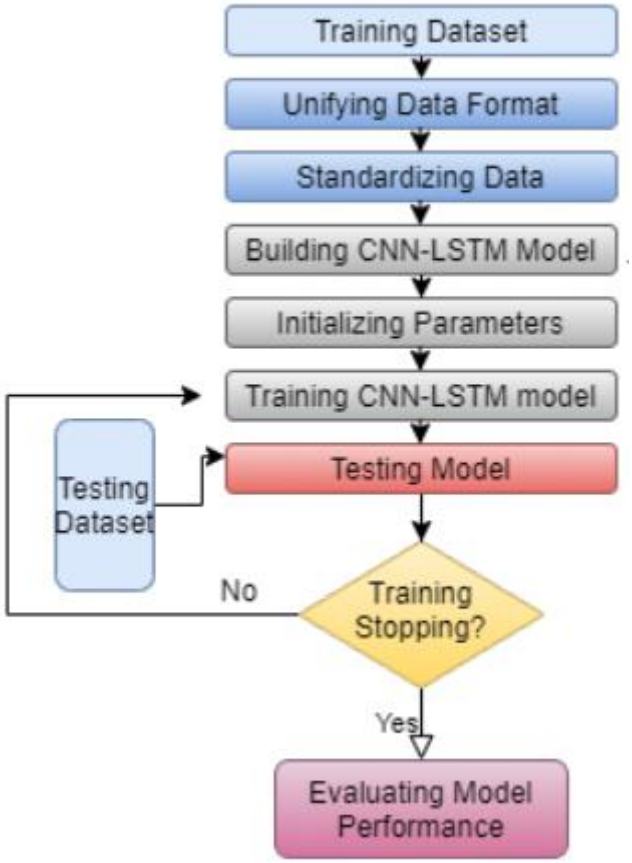

Figure 1. Proposed Methodology

1. NSL-KDD Dataset Loading: The NSL-KDD dataset is loaded into the Google Colaboratory environment.
2. Data Pre-Processing: All missing values in the dataset will be deleted at this point.
3. Feature Selection: The best features from datasets were chosen using the Anova Feature.
4. Dataset Split into Train and Test: The NSL-KDD dataset was split into two parts: train and test.
5. Classification Steps: To categorize the trained data set, the CNN-LSTM model was utilized.
6. Trained Data: The data that has been trained will be saved.
7. Test data prediction: - To obtain the prediction result, specify the trained model file.
8. Performance metric: At this point, performance metrics were used to obtain findings such as the confusion matrix, F1 Score, Recall, and accuracy.

The paradigm for detecting intrusion attacks centered on the Convolutional Neural Network algorithm with LSTM as output. The structure mainly consists of three phases:

Data Pre-processing: The dataset for CNN-LSTM, as well as the pre-processing of the data, is extracted from the obtained CSV file. The features of the dataset are processed using the jupyter notebook and the numerical python framework. All null values and reductant records were removed from the dataset as a result of this procedure. The arrays are then transformed into matrices, which are used as CNN's principal input.

Modelling: This is the first step in the implementation process; it is here that the CNN deep learning algorithm is created, and the various CNN layers are fused with LSTM layers. The tensorflow and keras frameworks are used to accomplish this. A graphics driver is used in conjunction with the notebook juypter framework to maximize efficiency and reduce processing time. The entire method is run on both the GPU and the CPU, utilizing all of the system's capabilities and producing the evaluation metrics. Precision, accuracy, FI-score, and recall are all important factors.

Visualization: To aid reading comprehension, the metrics obtained in the second stage are translated into graphic representations in the form of graphs and confusion matrices.

Software and Hardware used:

Dataset	NSL-KDD
Machine used	High Performance Computer (HPC) Technology
RAM	16GB DDR4
Software	Python V 3.9, Google Colaboratory
Function	Relu and Softmax activation
Training set	Keras, TensorFlow, Scikit learn

Multiple levels of concatenation are referred to as "deep learning." The input layer is the first layer, while the output layer is the last layer. In addition, between the input and output layers there are hidden layers. Each layer is made up of a number of units called neurons. The input is received by CNN, which analyses it before applying a ReLU activation function. The image data is also subjected to the same filter. This will ensure that the consistency of the input image data is preserved. The number of weights is determined by the number of filters. Because CNNs are sparsely connected, the number of weights is reduced, and computing speed improves quickly. Its output is transferred to the Pooling layer after convolution. The output of the convolution layer is fed into the LSTM algorithm. There will be three gates in each LSTM unit: input, forget, and output. The convolution layer's output will be used as an input to the LSTM. Apply the LSTM layer to the output after convolution and pooling, and it is transformed to a vector. After utilizing a loss function to calculate the error between the expected and predicted values. In general, fully connected layers are utilized at the end of a CNN, but adding fully connected layers did not improve the model. As a result, the fully connected layers are shortened. Following that, the LSTM calculates an attack label that is very close to the actual attack. The cost function is calculated, and the error is sent back to the network in order to reduce the loss. Adam optimizer is used to update the weights. These weights will be placed in the following period for training, and the procedure will continue until all of the epochs have been completed.

4. DESIGN SPECIFICATION

The training and testing experiment was conducted on an Intel Core i7 processor with 16GB RAM, 1TB SSD storage, and Windows 11 64-bit OS. Python 3.9.7 was used to process the datasets, and Anaconda Navigator was utilized in conjunction with Python and Jupyter Notebook v6.4.8.

In prior research, alternative machine learning algorithms were demonstrated to be promising in predicting intrusion on NSL -KDD datasets. However, because shallow learning has a large false-positive rate, this paper focuses on deep learning methods, a branch of machine learning that improves and develops shallow learning. Deep learning allows multiple representations to be used to model complicated relationships and concepts. Well-known deep learning methods such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and CNN + LSTM are discussed in this study.

CNN, commonly known as ConvNet, is a deep learning technique that is mostly used for image categorization by assigning different attributes or objects in the image and allowing discrimination between them. The Visual Cortex influenced CNN's architecture, which matches the connectivity network of neurons in the human brain. Convolution, max-pooling, complete connection, and fully connected-Relu are some of the steps for categorization of the dataset. Convolution is important for feature extraction and data resizing after several steps [14].

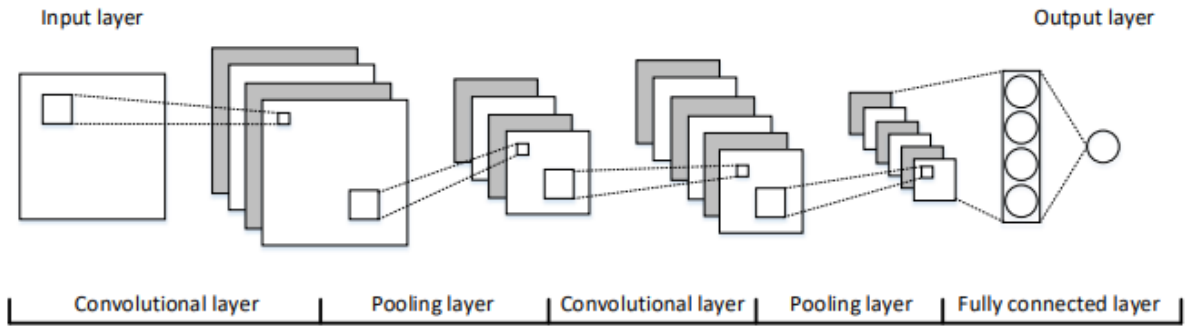

Figure 2: Architecture of a CNN

A recurrent neural network (RNN) is a type of neural network that tries to mimic time or any other sequence of events, such as language. Standard RNN has one flaw: the distance between words or sequences values grows as the distance between them grows, i.e., they are separated by a huge number of other words or values. The vanishing gradient problem (or exploding gradient problem) occurs when such dependencies are modelled [1].

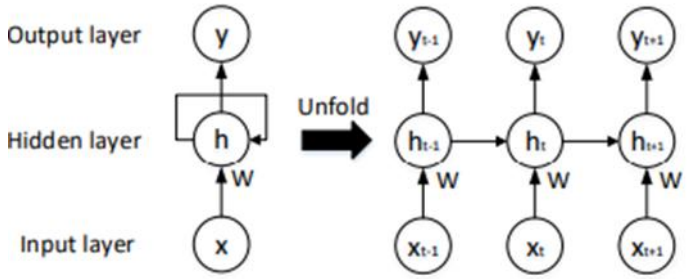

Figure 3: Structure of an RNN

CNN-LSTM system

Unlike standard Convolutional Neural Networks (CNN), RNNs aid in the creation of interaction between input sequences, resulting in a novel approach to feature hybrid. Researchers developed approaches for hybridizing features using LSTM, an RNN variation that can extract the long-term relationships of data characteristics in the sequence to increase recognition accuracy. In this research, a novel but related technique for extracting features from a dataset using multiple convolutional kernels. Furthermore, this strategy creates a complete end-to-end mapping of the relationship between features and attack types. This method is divided into two steps, the first of which is feature extraction using CNN and the second of which is feature fusion using LSTM.

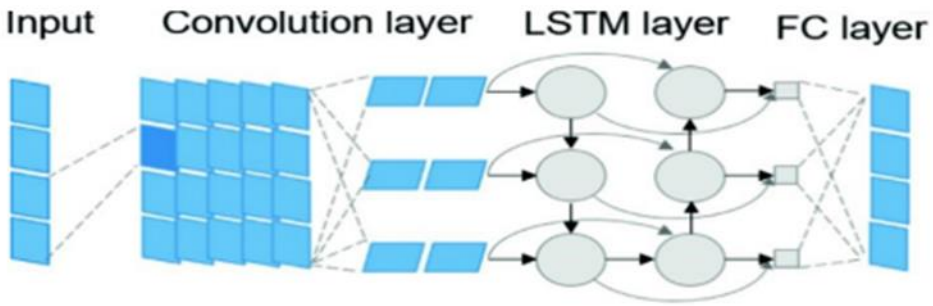

Figure 4: Hybridised architecture of CNN and LSTM

5. IMPLEMENTATION

After the models have been trained, they should be assessed on the test set that was left over. The confusion matrix was then used to calculate the performance measures. The elements of the confusion matrix are used to represent the expected and actual classifications. The classification process yields two classes: right and wrong. In order to compute the confusion matrix, we evaluated four basic scenarios:

- True Positive (TP) measures the proportion of genuine positives that are accurately detected.
- False negative (FN) refers to incorrect predictions. It identifies instances that are malicious yet are wrongly predicted as normal by the model.
- False positive (FP) refers to an inaccurate positive prediction when the detected assault is actually normal.
- True negative (TN) measures the proportion of actual negatives that are correctly identified attacks.

For a given classification device, the diagonal confusion matrix reflects the correct forecast, whereas nondiagonal components represent the incorrect forecast. Table below depicts this confusion matrix property. In addition, the following are some of the many evaluation tools that have been utilized in recent studies:
Precision refers to the exact number of attacks expected for all samples.

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

Recall: It's the proportion of samples correctly classified as Attacks to the total number of samples classified as Attacks. The phrase "Detection Rate" is sometimes used.

$$\text{Recall} = \text{Detection Rate} = \frac{TP}{TP + FN}.$$

False Alarm Rate: The ratio of incorrectly predicted Attack samples to all Normal samples is known as the false positive rate.

$$\text{False Alarm Rate} = \frac{FP}{FP + TN}.$$

True Negative Rate: The ratio of accurately diagnosed Normal samples to all samples labeled as Normal samples is what it's called.

$$\text{True Negative Rate} = \frac{TN}{TN + FP}.$$

Accuracy: It's the proportion of successfully classified examples to the total number of occurrences. Detection Accuracy refers to how well a dataset is balanced, and it can be used to evaluate a system's performance.

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

F-Measure: It is defined as the harmonic mean of the precision and recall variables when they are combined. In other words, this is a statistical technique for analysing a system's correctness, taking into account both the accuracy and recall of the system under investigation.

$$\text{F Measure} = 2\left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right).$$

Confusion Matrix

		Predicted class	
		Attack	**Normal**
Actual Class	**Attack**	True Positive	False Negative
	Normal	False Positive	True Negative

Table 1. Confusion Matrix

6. EVALUATION

The performance of the CNN-LSTM algorithm with CNN and RNN will be reviewed in this section. The accuracy of CNN-LSTM, CNN and RNN is measured and compared. Precision, false positive, F1 score, and recall are used to evaluate the performance of these algorithms.

Experiment 1: Performance of CNN algorithm on NSL-KDD dataset.
Train on 74258 samples, validate on 74259 samples.
Execution time: 50s.

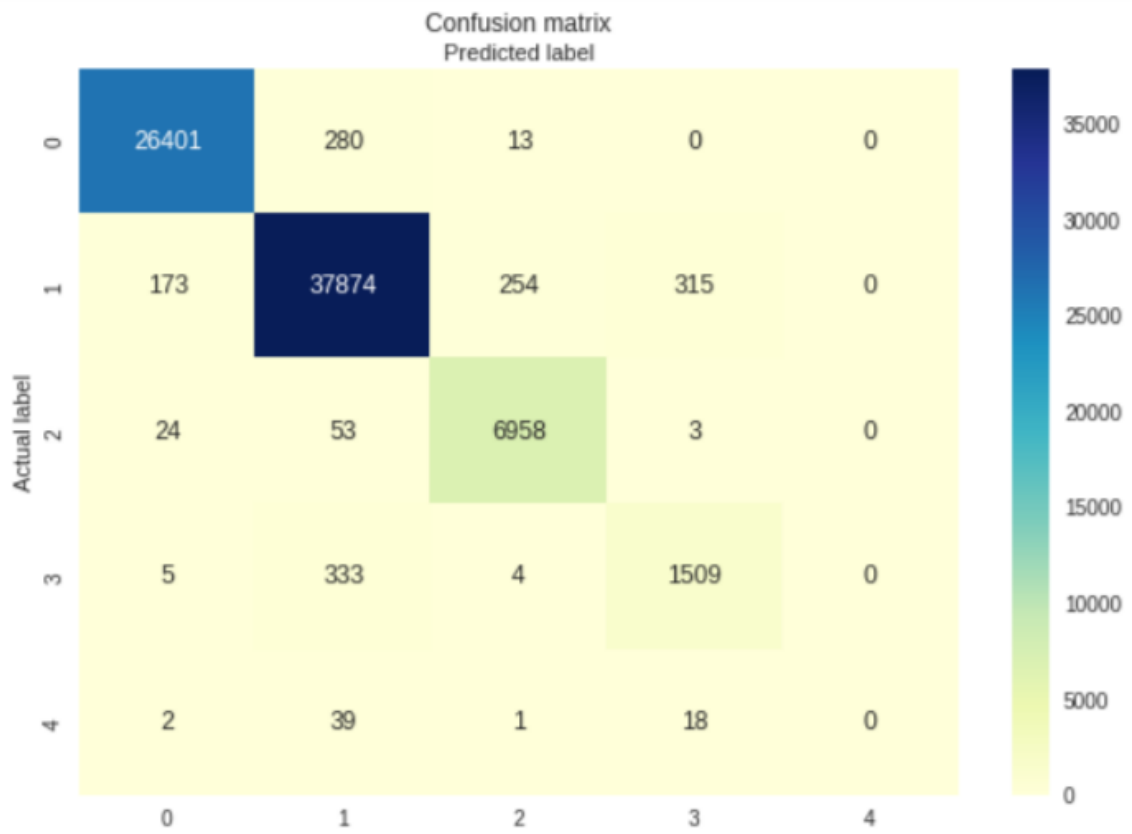

Figure 5: Confusion metrics of CNN

Figure 6: Training and Validation accuracy

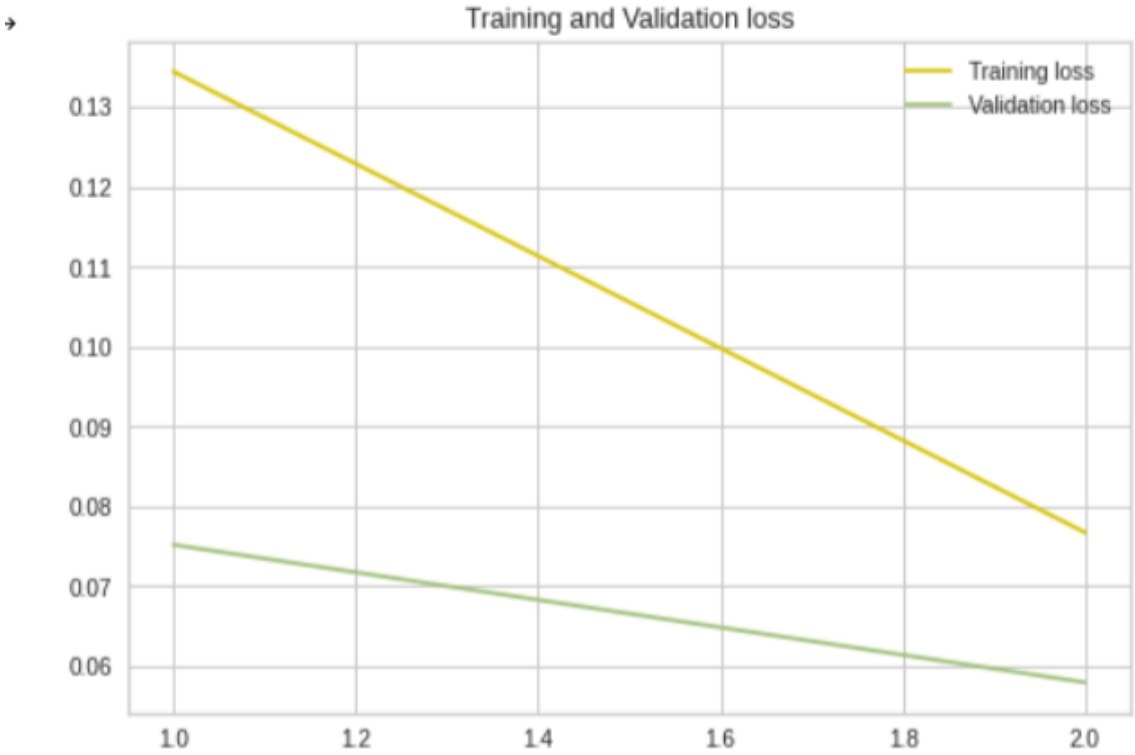

Figure 7: Training and Validation loss

	precision	recall	f1-score	support
DoS	0.99	0.99	0.99	26694
Normal	0.98	0.98	0.98	38616
Probe	0.96	0.99	0.98	7038
R2L	0.82	0.82	0.82	1851
U2R	0.00	0.00	0.00	60
accuracy			0.98	74259
macro avg	0.75	0.75	0.75	74259
weighted avg	0.98	0.98	0.98	74259

Table 2: Performance metrics of CNN

Experiment 2: Performance of CNN-LSTM algorithm on NSL-KDD dataset.
Train on 74259 samples, validate on 74258 samples.
Execution time: 228s.

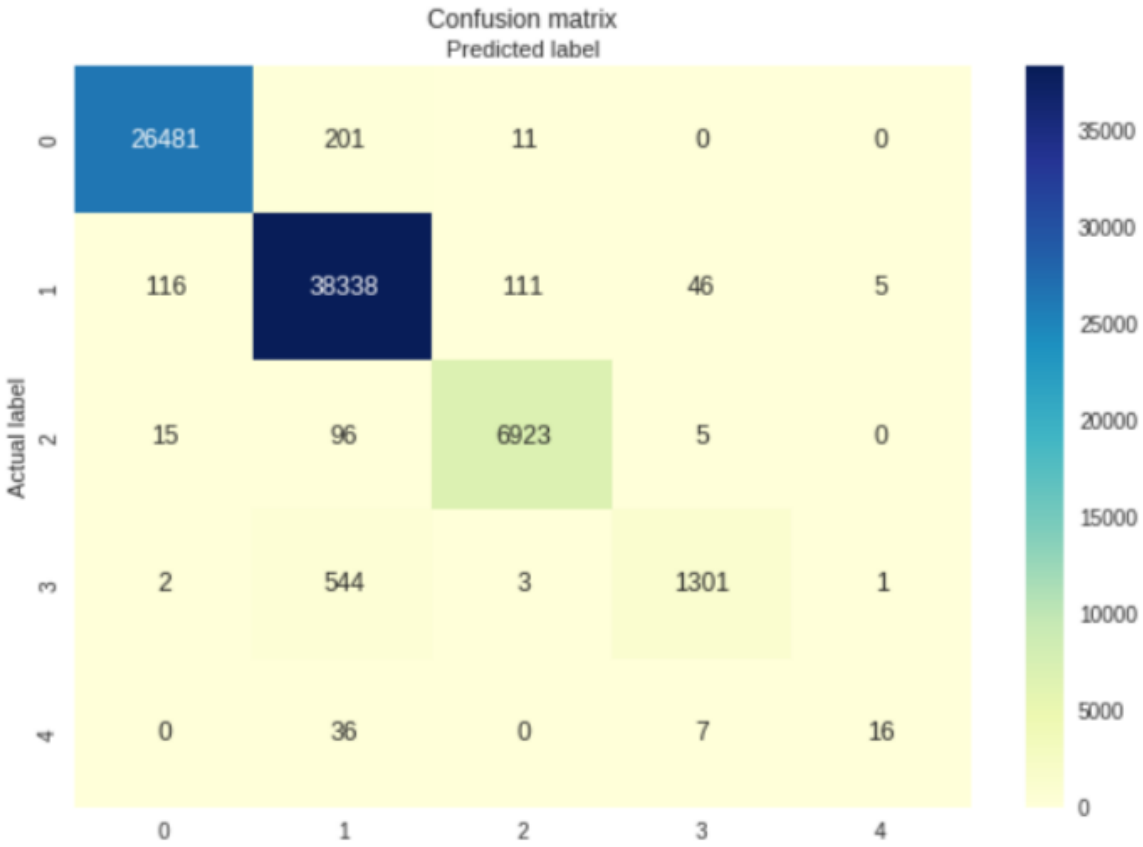

Figure 8: Confusion metrics of CNN-LSTM

Figure 9: Training and Validation accuracy

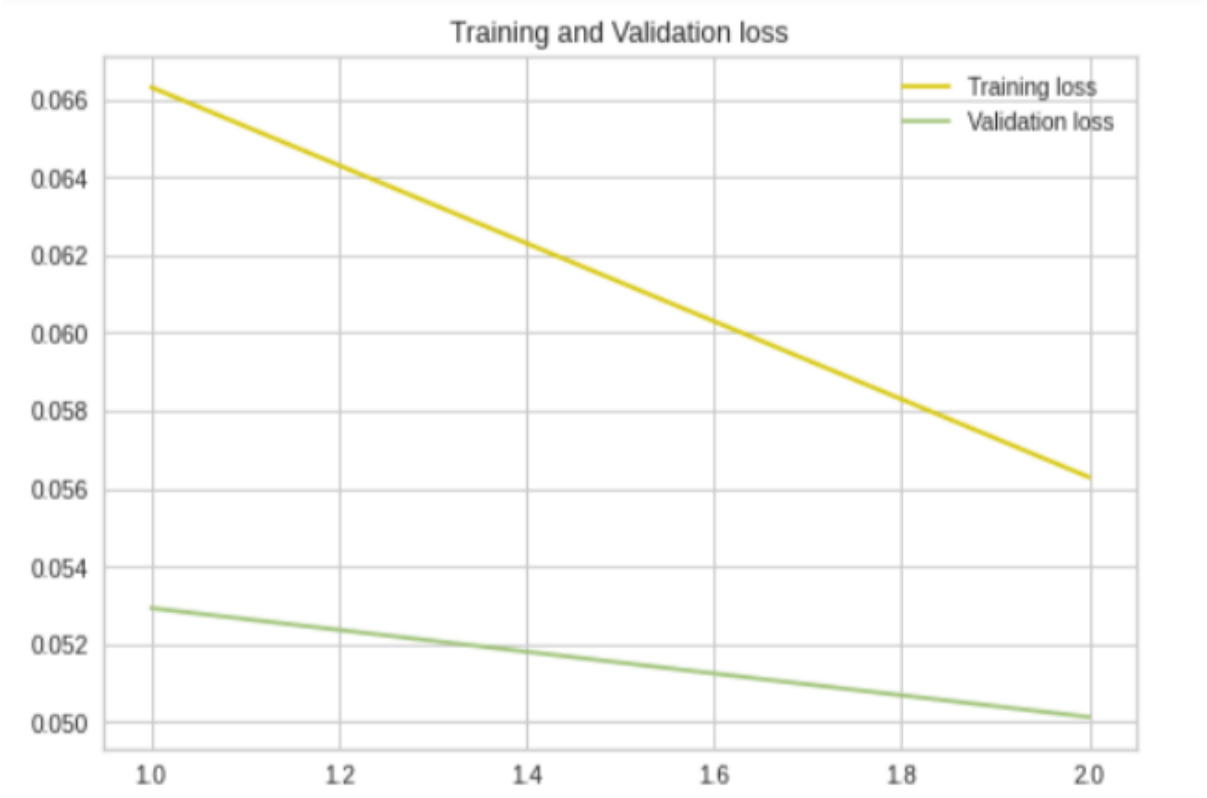

Figure 10: Training and Validation loss

	precision	recall	f1-score	support
DoS	1.00	0.99	0.99	26693
Normal	0.98	0.99	0.99	38616
Probe	0.98	0.98	0.98	7039
R2L	0.96	0.70	0.81	1851
U2R	0.73	0.27	0.40	59
accuracy			0.98	74258
macro avg	0.93	0.79	0.83	74258
weighted avg	0.98	0.98	0.98	74258

Table 3: Performance metrics of CNN-LSTM

Experiment 3: Performance of RNN algorithm on NSL-KDD dataset.
Train on 74259 samples, validate on 74258 samples.
Execution time: 165s.

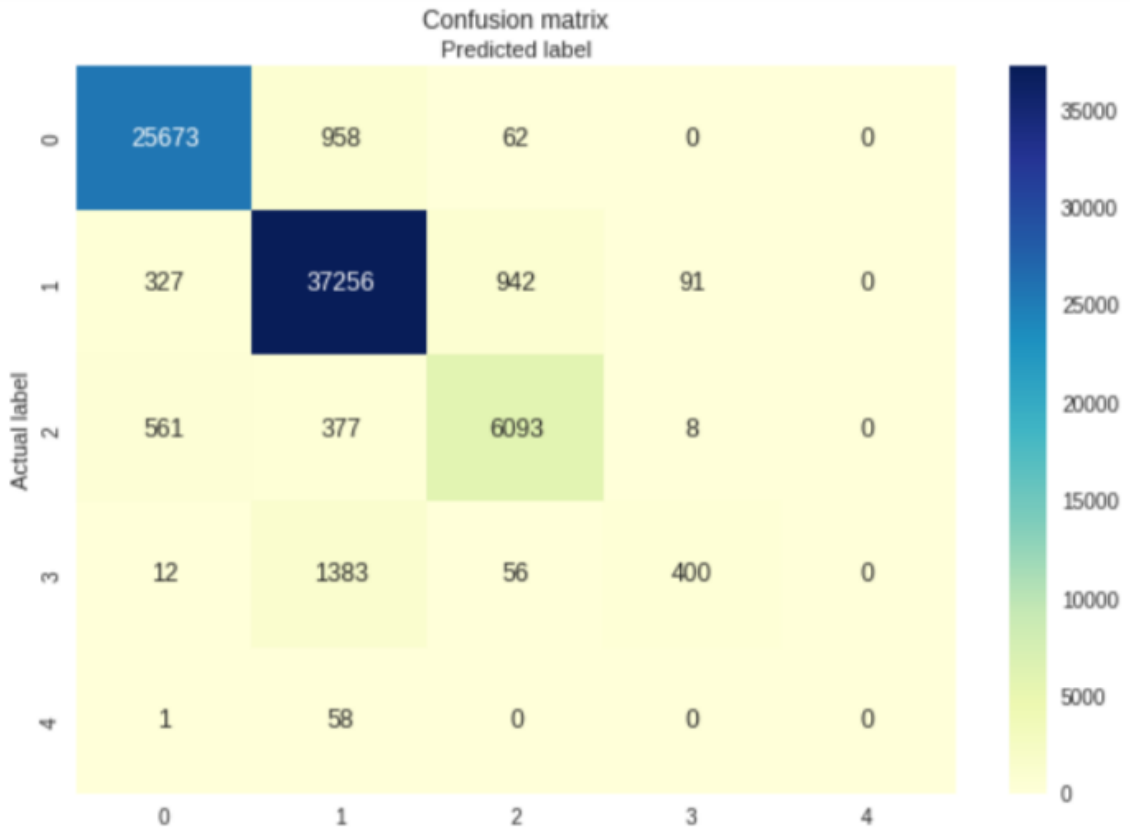

Figure 11: Confusion metrics of RNN

Figure 12: Training and Validation accuracy

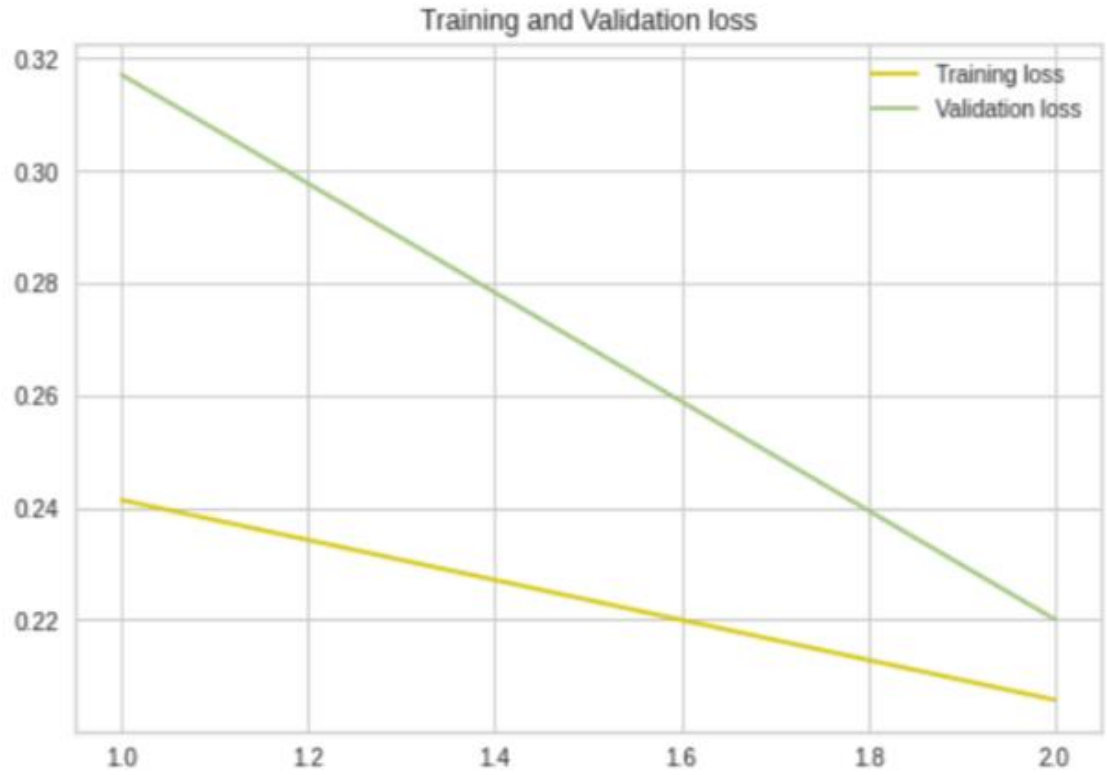

Figure 13: Training and Validation loss

	precision	recall	f1-score	support
DoS	0.97	0.96	0.96	26693
Normal	0.93	0.96	0.95	38616
Probe	0.85	0.87	0.86	7039
R2L	0.80	0.22	0.34	1851
U2R	0.00	0.00	0.00	59
accuracy			0.93	74258
macro avg	0.71	0.60	0.62	74258
weighted avg	0.93	0.93	0.93	74258

Table 4: Performance metrics of RNN

Discussion

Several experiments were carried out to evaluate the model in this research. The novel CNN-LSTM model outperformed traditional intrusion detection algorithms, according to the findings. The models were trained and tested using the NSL-KDD dataset. RNN has the least accuracy of 93.49% compared to CNN with 97.96% and CNN-LSTM has the highest accuracy of 98.39%. According to the results, CNN-LSTM showed 100% precision for DoS attacks, CNN only with 99% and RNN has the least precision for DoS with 97%. In these three models, only CNN-LSTM has detected User to Root(U2R) attacks for precision, recall and fi-score with 73%, 24% and 40% respectively and the rest of the models, CNN and RNN didn't detect U2R attacks. For CNN only algorithm, recall for Root to Local(R2L) attacks showed 82%, which is higher than CNN-LSTM hybrid system with 70% and RNN has the least detection rate of 22%.

7. CONCLUSION AND FUTURE WORK

This study built a combination of Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) for network intrusion detection, which is a very effective approach. Without using any hyperparameter adjustment, unprecedented high accuracy was achieved on a basic NSL KDD dataset. Deep learning algorithms for intrusion prevention are shown to be very promising and successful in this research.

However, one disadvantage of this method is that all of the tests were performed on a single dataset. Because the signature of the attack traffic varies frequently, it is critical to evaluate it on more recent datasets. In the future, the research should be expanded to include live network testing of the algorithm, as well as focus on using DL as an attribute extraction tool to develop competent data illustrations in the event of additional anomaly recognition difficulties in a more recent dataset.

8 Bibliography

[1] Y. Ding and Y. Zhai, "Intrusion Detection System for NSL-KDD Dataset Using Convolutional Neural Networks," *Association for Computing Machinery,* 2018.

[2] N. Das and T. Sarkar, "Survey on Host and Network Based Intrusion Detection System," *Advanced Networking and Applications,* vol. 6, no. 2, pp. 2266-2269, 2014.

[3] C. YIN, Y. ZHU, J. FEI and X. HE, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access,* vol. 5, 2017.

[4] G. E. Hinton, S. Osindero and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation,* vol. 18, no. 7, 2006.

[5] Y. Imrana, Y. Xiang, L. Ali and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Systems with Applications,* vol. 185, 2021.

[6] B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," *IEEE,* p. 2016.

[7] Y. Li, R. Ma and R. Jiao, "A Hybrid Malicious Code Detection Method based on Deep," *International Journal of Security and Its Applications,* vol. 9, 2015.

[8] M. Latah and L. Toker , "An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks," *CCF Transactions on Networking,* p. 261–271 , 2021.

[9] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access ,* vol. 7, 2019.

[10] Y. Yang, K. Zheng, C. Wu and Y. Yang, "Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network," *Security and Privacy Techniques in IoT Environment,* 2019.

[11] "Convolutional Neural Networks," in *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*, 2018.

[12] A. M. Denton, M. Ahsan, D. Franzen and J. Nowatzki, "Multi-scalar Analysis of Geospatial Agricultural Data for Sustainability," in *IEEE International Conference on Big Data (Big Data)*, 2016.

[13] J. Kim, J. Kim, H. L. Thi Thu and H. Kim, "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection," *International Conference on Platform Technology and Service (PlatCon),* 2016.

[14] L. Mohammadpour, T. C. Ling, C. S. Liew and C. Y. Chong, "A Convolutional Neural Network for Network Intrusion Detection System," *Proceedings of the APAN ,* 2018.

[15] V. Rajan, "Towards Efficient Intrusion Detection using Deep Learning Techniques: A Review," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 6, no. 10, 2017.