Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 46 (2025), No. 4-A, pp. 939–948
DOI : 10.47974/JIOS-1819

Securing the future of wireless communications : Utilizing a blend of deep learning techniques for identifying risks in advanced communication networks

Megha Jain [‡]
Ravi Verma [†]
School of Computing Science and Engineering
VIT Bhopal University
Bhopal 466114
Madhya Pradesh
India

Sunil Kumar [§]
Department of IoT & Intelligent Systems
Manipal University Jaipur
Jaipur 303007
Rajasthan
India

Atul Srivastava *
Department of Computer Science and Engineering
Amity School of Engineering and Technology
Lucknow 201313
Uttar Pradesh
India

Anuradha Pillai [@]
Department of Computer Science and Engineering
Symbiosis Institute of Technology
Pune 412115
Maharashtra
India

[‡] *E-mail:* meghajain37@gmail.com
[†] *E-mail:* ravi.verma@vitbhopal.ac.in
[§] *E-mail:* sunil.kumard@jaipur.manipal.edu
[*] *E-mail:* atul.nd2@gmail.com (Corresponding Author)
[@] *E-mail:* anuradha.pillai@sitpune.edu.in

Vijay Shankar Sharma [#]
Department of Computer and Communication Engineering
Manipal University Jaipur
Jaipur 303007
Rajasthan
India

Abstract

The rapid development of wireless communication and mobile networks, combined with broad use of IoT devices, has created substantial security concerns for core and edge networks. This study underlines the importance of installing strong security measures to combat new threats and encourages the creation of flexible security systems that may be tailored to future communication technologies. This research focuses on the boundaries of networks, where multiple nodes provide various services through terahertz radio frequency signals. This article presents a deep learning methodology that integrates convolutional neural network layers with LSTM for the analysis of sequential data. Using the NSL_KDD dataset as a comparison with conventional machine learning techniques, the new model shows better accuracy, detection rate, and fewer false alarms. This suggests its effectiveness in recognizing different types of attacks and minimizing incorrect identifications better accuracy, detection rate, and fewer false alarms. This suggests its effectiveness in recognizing different types of attacks and minimizing incorrect identifications.

Subject Classification: *Primary 93A30, Secondary 49K15.*

Keywords: *CNN, Hybrid model, Intrusion detection systems (IDSs), LSTM, Machine learning algorithms, Wireless communications.*

1. Introduction

As wireless communications improve, mobile network topologies evolve, and IoE devices and apps become more popular, security issues arise. Core and edge networks face these difficulties, requiring a robust plan to reduce data security breaches. Given the advancing risks to communication networks, academic and corporate research must prioritise the development and integration of intelligent security systems. Even after 6G, these systems must adapt to the latest communication system technology.

Intruder Detection Systems (IDSs) signal, when unusual activity or known dangers occur to protect information systems. These systems are essential for monitoring computer systems for unusual behaviour that passes packet filtering. IDSs use two basic methods: AIDS and SIDS [1]. A

[#] *E-mail:* **vijayshankar.sharma@jaipur.manipal.edu**

hybrid IDS model utilising LSTM and CNNs mitigates these issues. This innovative solution uses CNN and LSTM architectures to reduce false alarms and improve accuracy.

Key Contribution

- The research presents an improved anomaly detection classification model that covers data collection, preprocessing, and classification.
- Created a hybrid deep learning model to classify various assaults in the NSL_KDD dataset.
- An advanced architecture for sequential data processing is created by combining CNN and LSTM layers.
- The paper emphasizes the significance of minimizing false alarms in anomaly detection systems. This ensures accurate identification of genuine threats while reducing unnecessary alerts.

This document has the following sections in order: Section 2 will address the literature review. Section 3 covers the intended methodology, including dataset description, preprocessing, and model architecture. Section 4 presents results, discussion, and experimental setup and model evaluation. Section 5 will conclude the report by summarizing the key findings and suggesting future research.

2. Literature Review

Network Intrusion Detection Systems (NIDS) protect networks by detecting malicious activity and unexpected network traffic. Using machine learning algorithms, feature selection, and optimization, many studies have proposed new ways to increase NIDS efficacy and efficiency. Hamed Haddad and colleagues [2] proposed a two-tier machine learning network anomaly detection solution. This model employs diverse categorisation techniques to improve the efficacy of IDS. To address performance issues in high-dimensional data fields, Kasongo et al. [3] stressed feature selection in machine learning-based IDSs. They advised using appropriate feature extraction methods to improve intrusion detection systems. Mittal et al. [4] employed neural networks to create power-saving protocols for wireless sensor networks to improve reliability and efficiency. Their research sought to improve hybrid wireless sensor network anomaly detection while conserving energy. Jaw et al. [5]

suggested using a attribute selection technique and an ensemble classifier for intrusion detection to quickly choose relevant features and ensure consistent attack classification in NIDS. Oleiwi et al. [6] used ensemble learning to detect communication network irregularities. Ensemble learning was used to improve communication network abnormality detection.

Mhawi et al. [7] suggested a novel NIDS framework based on hybrid attribute selection and ensemble learning framework. Their approach uses forest panel features and pertinent feature selection to improve intrusion detection systems' performance. Salim et al. [8] evaluated machine learning intrusion detection classifiers using multiple datasets and feature reduction techniques. Their research tested the proposed network intrusion detection model on different datasets. Xu and others [9] proposed intrusion detection using deep learning techniques, demonstrating the ability to effectively handle imbalanced data and optimize the selection of feature subsets. Using the NSL_KDD dataset, they evaluated a DNN model for network intrusion detection.

3. Proposed Methodology

The NSL_KDD dataset [10] is widely used in cybersecurity research, especially for IDS evaluation. The dataset is revised from KDD Cup 1999 fixing its redundancy and unrealistic traffic patterns. This dataset generates network traffic data in a controlled environment to simulate typical activities and cyberattacks. The dataset has two primary parts: training and testing. About 125,973 examples of normal and attack categories are in the training set. DoS, Probe, R2L, U2R, and Normal are the five categories of these instances. Normal network data without malicious intent is used to compare attack types. Like the testing set, 22,544 cases are classified with the same attack kinds. The 22-attack KDDTrain dataset is used for training. Testing then uses the KDDTest dataset, which adds 17 attack types. These attacks, with common traits, are divided into four training and evaluation groups. These groups cover four attack kinds- Denial of Service, Probe, User to Root, Remote to Local.

The NSL_KDD dataset must undergo several critical preparations before analysis and model training. The category columns of the dataset include protocol type, service, and flag. In machine learning methods, it is necessary to transform categorical data into numeric values. In the preprocessing module of Sklearn, LabelEncoder is used to transform categorical data to numeric values. LabelEncoder creates numerical labels

for each category before OneHotEncoder. At this stage, unique integers are assigned to each category of the variables. Next, we will use OneHotEncoder to convert these numerical labels into binary vectors. Each categorical variable is expanded into multiple binary columns representing unique categories. We then modify the dataset and normalize the feature scales. Usually, the performance of the model increases when this normalizing process is carried out with a standardizer. Standard deviation of 1 and a mean of 0 mark the standardized approach of the data. After category encoding and normalization, the encoded columns are concatenated with numerical features. Finally, we may feed processed data into a CNN+LSTM model for analysis and prediction. CNNs' spatial hierarchies and LSTM networks' temporal dependencies make this architecture suited for representing sequential network traffic

3.1 *Proposed Hybrid Method*

Figure 1 shows the hybrid deep learning technique for classification using the NSL_KDD dataset using CNN and LSTM layers. Sequential data analysis is supported by this resilient architecture. As outlined in algorithm 1, the model extracts different levels of abstraction from input sequences using Conv1D layers with 32, 64, and 128 filter sizes and 3 kernel sizes. Following two-pool MaxPooling1D layers preserves significant features and condenses spatial dimensions. A 100-unit LSTM layer can express long-term dependencies in sequential data like network traffic patterns. Class predictions and LSTM categorization are done with a Dense layer. In the NSL_KDD dataset, the model is trained to classify DoS, Probe, R2L, U2R, and Normal events using the Adam optimiser.

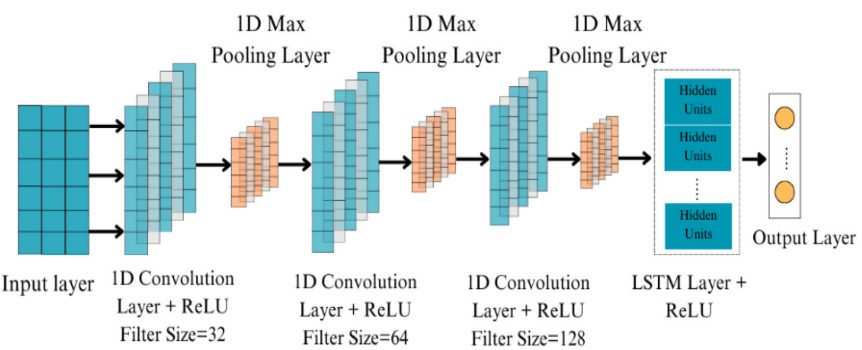

Figure 1

Visual Representation of Proposed framework.

Algorithm 1: Proposed Hybrid Deep Learning Model for NSL_KDD Dataset

Input: The NSL_KDD dataset contains network traffic data.

Output: Trained a Hybrid ML model to classify different kinds of attacks.

Procedure: Step 1: Build the architecture of the hybrid deep learning model:

1) Build a sequential model.
2) To capture different levels of abstraction in the input sequence, we use 1Dconv layers with filter widths of 32, 64, and 128, and the size of the convolutional kernel is set to 3.
3) To decrease the spatial dimensions while keeping noticeable features, use a MaxPooling1D layer with a pool size of two.
4) Add an LSTM layer with 100 units to capture long-term relationships in sequence data.
5) To convert the output of the LSTM into appropriate category predictions, we use a dense layer for classification.

Step 2: Compile the model:

1) Using the Adam optimizer during training.
2) Employ categorical cross-entropy as the loss function for multi-class classification.
3) Specify the metrics for evaluating the model, including Accuracy, Detection rate, and False alarm rate.

Step 3: Train the model:

1) Divide the preprocessed dataset into sets for training and validation.
2) Utilizing the validation data for validation during training, train the model using the training set of data.
3) Monitor training performance and adjust hyperparameters as needed.

Step 4: Assess the model:

1) Evaluate the trained model on a test dataset.
2) Examine the model's performance indicators, such as Accuracy, Detection rate, and False alarm rate.

Step 5: Interpret the results:

1) Assess how well the model classifies different forms of attacks.

4. Result and Discussion

The proposed model underwent thorough testing on the NSL_KDD dataset. In this evaluation, we examined various optimal configurations of training hyperparameters. This includes three types of convolution kernel sizes, an initial learning rate of 0.001, and one-dimensional convolution filters ranging from 32 to 128. The ReLU activation function added non-linearity to the model's computations. The model minimizes loss functions using the Adam optimizer for 50 epochs of training to maximum performance. The model also includes a single layer of 100-unit LSTM with ReLU activation. This complex architecture increases model predictions by capturing temporal correlations inside the dataset. We carefully examine the accuracy, DR, and FAR of the model to evaluate its efficacy.

4.1 *Experimental Configuration*

We tested the anomaly detection classification model using Python simulation. This model will be implemented using TensorFlow and Keras on a Windows 10 PC with 8GB of memory, a 1TB SSD, and an NVIDIA GeForce GTX processor.

4.2 *Evaluation of the Suggested Model*

Accuracy, Detection rate, and False alarm rate are used to compare our hybrid strategy with rule-based ensemble methods, Naive Bayes, GAR Forest, and SVM techniques. The NSL_KDD dataset indicates that the suggested model performs well in classification. This model uses LSTM and CNN architectures. The model classifies events more accurately than prior methods at 97.24% accuracy. Its high detection rate (DR) of 97.02% suggests it can dependably spot patterns and provide accurate forecasts, demonstrating it can find optimistic situations. The low false alarm rate (FAR) of our model is 3.76%, which contributes to its superiority. This indicates that this model reduces misinformation, which is very important in applications that could lead to injuries or unnecessary actions. The low false alarm rate of this model demonstrates its reliability and accuracy. Figures 2 provide ML/DL comparisons of these results. Table 1 compares our model to other popular ML/DL approaches using the NSL_KDD

dataset. Figures 3, 4, 5, and 6 show confusion matrix for each attack category.

Table 1

Evaluation of Machine Learning Techniques, employed dataset NSL_KDD.

References	Classification Framework	Accuracy	Detection Rate	False Alarm Rate
[11]	Rule-based Ensemble method	80.00	81.00	N/A
[2]	Naïve Bayes	83.00	82.00	4.83
[12]	GAR Forest	85.00	N/A	15.00
[4]	SVM	95.00	96.00	5.11
Proposed Method	CNN+LSTM	97.24	97.02	3.76

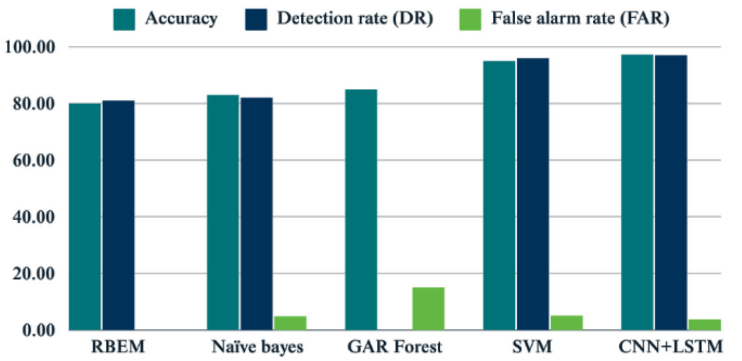

Figure 2

A comparison of the suggested framework's expected results with those of many well-known ML techniques utilizing the NSL_KDD dataset.

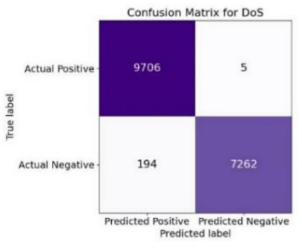

Figure 3

Denial of Service Class

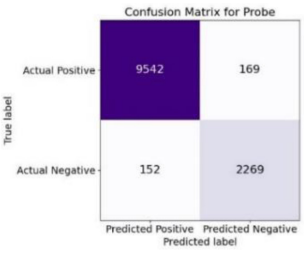

Figure 4

Probe Class

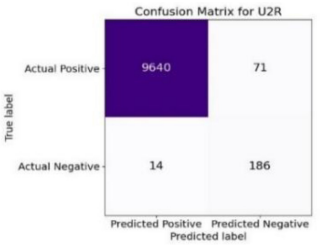

Figure 5
User to Root class

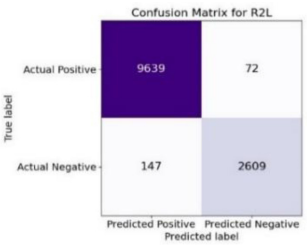

Figure 6
Remote to Local Class

5. Discussion and Conclusion

The quest of more effective intrusion detection systems never stops in the realm of cybersecurity as network threats change. We suggested a novel framework that combines LSTM architectures with CNNs. This hybrid model leverages Long Short-Term Memory networks' (LSTM) advantages in temporal correlation capturing and Convolutional Neural Networks' (CNN) strengths in spatial feature extraction. Designed on a Conv1D layer, this architecture seeks to find spatial patterns in the input data. We then extract salient features and simplify calculation by means of the Max Pooling 1D layer. LSTM layers enable the model to identify the intricate temporal correlations in network traffic, hence improving its capacity to identify minor anomalies suggesting possible hostile activity. Using the synergy between LSTM and CNN, our method seeks to establish this mixed paradigm as a basic element in safeguarding networks by thorough experiments and verification, enhancing the precision of intrusion detection.

References

[1] G. Zachos, I. E. Livieris, K. G. Margaritis, P. Pintelas, and S. Petridou, "An anomaly-based intrusion detection system for Internet of Medical Things networks," *Electronics*, vol. 10, no. 21, p. 2562 (2021).

[2] H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, and K.-K. R. Choo, "Two-tier network anomaly detection model: A machine learning approach," *Journal of Intelligent Information Systems,* vol. 48, pp. 61–74 (2017).

[3] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset," *Journal of Big Data,* vol. 7, no. 1, p. 105 (2020).

[4] M. Mittal, V. Sharma, R. Garg, M. Singh, and S. Verma, "Machine learning techniques for energy efficiency and anomaly detection in hybrid wireless sensor networks," *Energies,* vol. 14, no. 11, p. 3125 (2021).

[5] E. Jaw and X. Wang, "Feature selection and ensemble-based intrusion detection system: An efficient and comprehensive approach," *Symmetry,* vol. 13, no. 10, p. 1764 (2021).

[6] H. W. Oleiwi, M. A. Al-Rawi, A. H. Khudayer, A. A. S. Al-Waisy, and T. Abdellatif, "MLTS-ADCNS: Machine learning techniques for anomaly detection in communication networks," *IEEE Access,* vol. 10, pp. 91006–91017 (2022).

[7] D. N. Mhawi, H. A. Jalab, R. A. Al-Khateeb, and A. A. M. Dinar, "Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems," *Symmetry,* vol. 14, no. 7, p. 1461 (2022).

[8] S. Q. Mohammed and M. A. E. S. Hussein, "An innovative multi-dataset performance analysis of machine learning classifiers based on features reduction for intrusion detection," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 8s, pp. 553–569 (2024).

[9] B. Xu, J. Liu, Q. Liu, H. Guo, and Y. Zhang, "Strengthening network security: Deep learning models for intrusion detection with optimized feature subset and effective imbalance handling," *Computers, Materials & Continua,* vol. 78, no. 2 (2024).

[10] HoaNP, NSL-KDD Dataset, GitHub repository, 2024. [Online]. Available: https://github.com/HoaNP/NSL-KDD-DataSet.git [Accessed: Jun. 14, 2024].

[11] D. Gaikwad and R. Thool, "DAREnsemble: Decision tree and rule learner based ensemble for network intrusion detection system," in Proc. First Int. Conf. on Information and Communication Technology for Intelligent Systems: Volume 1, S. C. Satapathy and S. Das, Eds. Cham: Springer International Publishing, pp. 185–193 (2016).

[12] N. Kanakarajan and K. Muniasamy, "Improving the accuracy of intrusion detection using GAR-Forest with feature selection," in Proc. First Int. Conf. on Information and Communication Technology for Intelligent Systems, pp. 539–547 (2016).

Received October, 2024