DOI: 10.51153/kjcis.v7i2.235

Prediction of Network Intrusion Using CNN-LSTM

Humza Rana [a], Farwa Zainab [b], Farwa Raoof [c], Attiya Zahoor [d]

a Department of Computer Science Bahauddin Zakariya University Multan Pakistan (Humza.Rana99@gmail.com)
b Department of Computer Science Bahauddin Zakariya University Multan Pakistan (farwazainab09@gmail.com)
c Department of Computer Science Bahauddin Zakariya University Multan Pakistan (farwaraoof99@gmail.com)
d Department of Computer Science Bahauddin Zakariya University Multan Pakistan (attiyazahoor7@gmail.com)

Corresponding Author: *Humza Rana (Humza.Rana99@gmail.com)*

Abstract

Nowadays, network attacks have become a global issue as they intrude upon the operation and execution of the computer network. Network attacks are a major predicament liable to cause the loss of important information, hacked personal / sensitive data, and extortion. Intruders engage advanced systems and logics for hacking classified data, as they rapture the cost-effective traditional techniques used in the prevention of network intrusion. To predict and inhibit such incursions, a more formidable and successful approach is required. It should encompass a powerful deep learning method, appropriate and efficient in such predictions. In this paper, the convolution neural network model combined with LSTM is proposed for the prediction of network intrusion, which is one-dimensional. The proposed model is multiclass and tuned by different parameters to obtain the best efficiency, in the case of the multiclass dataset. This multiclass model is trained on the two multiclass datasets to get the best accuracy from the model on datasets. The first dataset named as wireless network dataset, contains four or five types of intrusion. The second is the Microsoft Malware dataset that contains the eight or nine-class intrusion type. The experiment from the proposed model gives 0.996%, and 0.985% accuracy performance in multiclass prediction of network intrusion. The performance of the proposed hybrid CNN-LSTM model shows better performance than existing approaches.

Keywords: Network Intrusion, Long Short-Term Memory, Convolution Neural Network, Deep Learning, etc.

1. Introduction

In present-day environment, the use of internet is exhaustive and is further enhanced in processes requiring remote accesses and administrations. This scenario inspires the network intruders to access vulnerable networks, and benefit from the stolen personal data and valuable information, including login credentials, for their lucrative profits including financial [1]. The intruders and their intrusions are modified to cater, filter and monitor vulnerable targets, instantaneously adapt and effectively ingress intended system(s) covertly [2]. To cater for the immense risks, financial and otherwise, of compromised data and information, it has become imperative to be knowledgeable of vulnerable

systems and environments, and induce powerful techniques (software and hardware) to limit such accesses and identify the intruders. The traditional methods like IDS and different former methods have often proved redundant, failing to predict the intrusion due to various shortcomings [4].

Given their various adaptations, any particular intrusion may require an advanced method for predicting it. Different types of machine learning (ML) algorithms presented by researchers now suggest the use of deep learning models to solve this problem, as they can easily extract features from data [3]. Utilizing an Intrusion Detection System (IDS) mostly relies on response duration for detection; when a user presses a URL, it can lead to crashes [5]. The prediction of network intrusion is a protection-based technique designed to combat network-based intrusions. Nowadays, deep learning methods have demonstrated exceptional performance in network intrusion detection [6]. Neural network models are robust and perform well across multiple classes. These models also require less time for training and prediction [7]. Saba et al. (2022) [2] used the Long Short-Term Memory (LSTM) network in their experiments. The dataset they employed consisted of two classes, and they proposed a model for predicting malicious activity. Their experiment achieved an accuracy of 99.6%, but this was limited to two classes. While their model performed excellently, a multiclass approach is needed to further enhance accuracy.

Joshi et al. (2021) [8] proposed fuzzy logic for feature extraction implementation. After feature extraction, the data was trained using an Artificial Neural Network (ANN) model to evaluate its performance. The model was trained on the CTU-13 dataset, achieving an accuracy of 99.9% for two classes of malicious activity. However, it also needs to be implemented in a multiclass context to improve efficiency for various malicious types. Chen et al. (2022) [9] proposed a Graph Neural Network (GNN) for similarity-based malicious classification. The malware dataset used in their experiment comprised ten malicious families, and their model achieved an accuracy of 93%. There is a need for improvement in multiclass performance, as well as additional datasets to evaluate the model across different aspects. Amit et al. [10] presented an intrusion detection system utilizing EGA-PSO techniques to enhance the Random Forest algorithm. The NSL-KDD dataset was used in their experiment, and their proposed method achieved an accuracy of 98.9% for a two-class dataset. However, the model also needs to be trained for multiclass scenarios, requiring multiple datasets for thorough evaluation. Various deep learning models have been proposed for predicting network intrusions, but they often lack accuracy in multiclass applications. In this paper, we propose a combination of a Convolutional Neural Network (CNN) and an LSTM model for predicting network intrusion using multiclass datasets. By tuning the proposed CNN + LSTM model, high performance

was achieved on two multiclass datasets. This CNN + LSTM model is designed to operate effectively across multiple classes, trained on two distinct multiclass network intrusion-type datasets.

1.1 Problem

- Various deep learning models have been proposed for intrusion detection and prediction; however, they often lack accuracy in multiclass scenarios and typically rely on only one dataset for experimentation.
- The IDS needs to be upgraded with a hybrid model to effectively handle the multiclass nature of intrusions and improve time performance in predictions.
- The proposed deep learning models for intrusion detection achieve satisfactory accuracy, but attention and residual methods have yet to be incorporated.
- A comparison of different deep learning models is necessary to demonstrate the real-time efficiency of the proposed model.

1.2 Contribution

- The Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models are proposed as a hybrid model for predicting network intrusion. This hybrid model is trained using hyperparameter tuning to achieve optimal performance on multiclass datasets.
- A residual connection is implemented to address the vanishing gradient problem, allowing the network to train more deeply and flow data effectively.
- An attention mechanism is applied to emphasize the relevant sections of the sequence, enhancing the model's capabilities. LSTM layers are incorporated to capture temporal dependencies and establish connections among intrusion events.
- The model is designed for multiclass intrusion-type datasets. The CNN layers extract features and identify patterns within the intrusion classes, while these features complement the LSTM layers in capturing temporal dependencies and linking intrusion events. Together, these methods enable the model to handle different datasets with varying numbers of classes.
- The model's performance is compared with that of other deep learning models. We also address the potential complexity of the model through hyperparameter tuning to improve performance and reduce computational complexity.

2. Related Work

In this section, we present various existing research studies. The deep learning methods utilized in these studies aim to predict network intrusion, along with the performance of machine learning (ML) models in detecting and predicting malicious intrusions. A hybrid model is also shown, highlighting its accuracy performance. The findings from the existing research are presented below.

2.1. Network Intrusion according to deep learning methods

Venkata et al. (2022) [11] proposed a CNN model for malware detection. They used various types of binaries as the dataset, which contained both malicious and non-malicious samples. Their proposed CNN model achieved an accuracy of 95%. Matthew et al. (2021) [12] also proposed a CNN model for classifying malicious Windows APIs, using a dataset of 5,385 samples that included eight malicious families. Their model achieved an accuracy of 98.17%. Lengfeng et al. (2022) [14] introduced a CNN model named MalShuffleNet, which was applied to the Malimg dataset containing 25 malicious families. Their model demonstrated a 99.03% accuracy, with the dataset consisting of two classes. Khan et al. (2022) [15] proposed a CNN model for classifying malicious families using the Microsoft Challenge dataset, which contains a large amount of data across nine classes of malware. Their model achieved an accuracy of 97.8%. It initially uses binary files and applies a deep neural network to the malicious image dataset. However, the model's accuracy needs improvement for multiclass classification. Martin et al. (2021) [16] developed a CNN model for detecting Android malware using the Derbin dataset, which contains different malicious families. Their model achieved 98% accuracy. However, performance improvements are needed for multiclass approaches, and the LIME technique was employed for comparing activations. Qiu et al. (2022) [14] proposed the MalShuffleNet model, using the Malimg dataset, which includes 25 families. Their model achieved 99.03% accuracy but needs further refinement for multiclass datasets. Currently, it is implemented on a single dataset, so additional multiclass datasets are necessary to evaluate model performance. A comparison of different deep learning models in multiclass datasets is also needed. Ahmad et al. (2023) [17] proposed an Inception V3 model for classifying malicious activity using the BIG dataset, which contains nine malware families. Their model achieved 99.6% training accuracy and 98.7% testing accuracy. However, they did not present comparisons between different deep learning models. The use of additional datasets is required to obtain a more accurate performance evaluation. Ketan et al. (2022) [18] developed a neural network for malicious classification using the Microsoft Big Challenge 2015 dataset, which includes nine malware families. Their ANN model achieved an accuracy of 90.17%, but performance improvements are needed for multiclass classification. The comparison is limited to a

small number of models, and deep learning models should be included in future comparisons. Lamia et al. (2022) [19] proposed a stack-based ensemble model for network intrusion detection, utilizing the NF-UQ-NIDS dataset. Their model achieved an accuracy of 98.40%. Additional datasets are needed, and comparisons of different deep learning models are necessary to assess model performance accurately. Kotian et al. (2021) [20] proposed a model for detecting malware in a cloud-based environment, using a dataset collected from various websites, including OpenStack and VirusShare. They employed the SMOTE method to balance the datasets, and their CNN model achieved an accuracy of 99.4%. Different types of datasets are required for further experimentation, and comparisons with other deep learning models should also be conducted. Wei et al. (2023) [22] introduced a 1D CNN for intrusion detection with BGSM techniques, using the KDD99 dataset. Their BSGM-QPSO-1DCNN model demonstrated an accuracy of 99.9%, but it only included five classes. Additional classes need to be implemented to evaluate model performance in multiclass scenarios, and multiple multiclass datasets should be used to assess how the model performs across various dataset types.

2.2. Network Intrusion according to ML

Almutairi et al. (2022) [23] proposed various machine learning algorithms for network intrusion detection using the NSL-KDD dataset, which includes both binary and multiclass approaches. They evaluated several algorithms, including Support Vector Machine (SVM), Random Forest (RF), Bayesian classifiers, and J48. The RF model performed the best, achieving an accuracy of 99.9%. However, since the experiment relied on a single dataset, it is necessary to implement multiple datasets with an increasing number of classes to fully evaluate the model's performance across different scenarios. Talukdar et al. (2024) [21] proposed a machine learning-based intrusion detection system utilizing the SMOTE-Tomek algorithm. They conducted their experiments using the WSN dataset, employing algorithms to balance the imbalanced dataset. Their model achieved an accuracy of 99.9%. However, it was trained on a single dataset, and the complexity of the model needs improvement. Additionally, a feature selection method is required to reduce this complexity. The machine learning model used in this experiment is time-consuming, so implementing a deep learning model could enhance both complexity management and time performance.

2.3. Network Intrusion according to Hybrid Models

Muhammad et al. (2022) [13] proposed a CNN-LSTM model for malware detection, achieving an accuracy of up to 99%. The dataset used in their experiment contained two malicious classes and was collected from Kaggle.

Cao et al. (2022) [6] developed a network intrusion detection model based on Gated Recurrent Units (GRU) and CNN. They utilized various datasets in their experiments, including NSL-KDD, CIC-IDS2017, and UNSW_NB15. Their proposed model achieved accuracy rates of 85.1%, 99.6%, and 99.6% on these datasets, which contain 15, 2, and 9 types of malicious classes, respectively. However, improvements are needed in the execution time and accuracy of the proposed model, as the parameters used are quite high.

2.4. Summarizing

The related work discussed above highlights the use of deep learning (DL) and machine learning (ML) methods in predicting network-based intrusions. Existing methods demonstrate satisfactory performance, but several gaps remain, such as the limited number of classes and datasets utilized. Additionally, there are limitations in the use of hybrid models. Comparisons among different approaches within deep learning networks are also scarce. From the literature, it is evident that DL models generally perform better in terms of efficiency and time performance compared to ML models in the prediction and detection of intrusions. Specifically, the CNN model achieves up to 99% accuracy in prediction and detection.

3. Methodology

The methodology outlines the proposed model, which integrates a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) model for network intrusion detection. The CNN-LSTM model is designed for multiclass prediction of network intrusions. This section details the step-by-step architecture of the model, including its layers and parameters. The proposed architecture specifies initial steps such as label encoding, data scaling, and data splitting before defining the model itself. This hybrid model is then tuned using various parameters to achieve optimal performance.

3.1. Propose Model Architecture

Step 1. Let Xdata be the Input data & Y be the Labeled Data

Step 2. Label Encoding and Categorical Operation in Y

- labels=labels-Encoder(Y)
- categorical = to-categorical(labels)

Step 3. Perform Min Max Scaler in Xdata

- Scaling = Min-Max-Scaler(Xdata)

Step 4. Divide the data into Trains, Tests

- Train_X, Test_X, Train_X, Test_Y=split(Scaling, categorical, tests-size=0.3)

Step 5. Reshape the Training & Testing Data

- Train_X_Shaped=Train_X. Reshaped
- Test_X_Shaped=Test_X. Reshaped

Step 6. Initialize the Propose Model CNN + LSTM

- Input_layers=Input(shape=(Train_X_Shaped))
- Conv_1 =Conv1D(64,3, activation-value='relu', padding-value='same') (input_layers)
- Conv_2 =Conv1D(64,3, activation-value='relu', padding-value='same') (Conv_1)

Apply Residual Connections in Model

- Resid= Conv1D(64,3, activation-value='relu', padding-value='same') (input_layers)
- Resid = Add()([residual,conv2])
- Pool_1 =Max_Pooling_1D(2)Resid
- Conv_3 =Conv1D(64,3, activation='relu', padding='same') (Pool_1)
- Pool_2=Max_Pooling_1D(2)(Conv_3)

Add LSTM Layers in the Model

- Lstm_out=LSTM(50, return_sequences=true)(Pool_2)

Apply Attention_Type Mechanism

- Atten = Atten()(Lstm_out,Lstm_out)

Flatten Layer

- Flatten=Flatten()(Atten)

Dense Layer

- Dense_1 = Dense_Layer(128, activation-value='relu')(Flatten)
- Outputs_layers= Dense_Layer(Y.shape, activation='SoftMax')(Dense_1)
- Model =Model(inputs=input_layers, outcomes=Outputs_layers)
- Model. Summary();

Step 7. Model Compilation

- Model.compiler(optimizer-value='Adam', lossed-function="categorical-cross-entropy", metrics=['categorical-accuracy']

Step 8. Model Training

- Model. Fit(X_train, Y_train, epochs=500, batch_size=3000, validation_split=0.3)

Step 9. Prediction By Model

- Y=Model. Predict(X_test, axis=1)
  Print(Y)

The above methodology shows the steps of the proposed CNN-LSTM hybrid model for the prediction of network intrusion. The Xdata is training input data and Y is labeled and several classes in it. In Y the categorical operation for label conversion. It first converts the string data into numerical code by using the label encoder mechanism after the label encoding operation is performed on the string type label data stored in variable label_encoding. Then categorical operation is performed on the label-encoded data which is used for multiclass operation and stored in categorical. Scaling steps come for normalizing the Xdata. The data splitting method is applied to scaling and categorical classes for X_train and Y_train. Then reshaping of data X_train, X_test for the model. Initialize the hybrid proposed model CNN + LSTM which contains the input layers network then first layer convolution one-dimensional layers with 64 filters and 3 shape size with input data having padding same,. The second convolution layer with 64 filters and an activation function relu in it has padding and the activation function relu takes input of first conv1d. Apply

the residual mechanism in it to train the model deeply and the overcome gradient issue in it allows the data flow in it effectively. The LSTM layers contain 50 neurons with the return_sequences parameter set to true, providing output after receiving input from the second pooling layer (Pool_2). An attention mechanism is applied to highlight the relevant parts of the sequence. Flattening layers transform the output of the attention mechanism into a single dimension. The dense fully connected layers consist of 128 neurons with the ReLU activation function, and the final layer is a SoftMax layer, with an output shape matching Y. The model compilation process includes the Adam optimizer, a loss function suitable for multiclass prediction, and metrics for accuracy. Finally, the model is trained on the training and testing data for 500 epochs with a batch size of 3000. The model's predictions are then evaluated on the test data. This architecture comprises nine steps that define the flow and functionality of the proposed hybrid model.

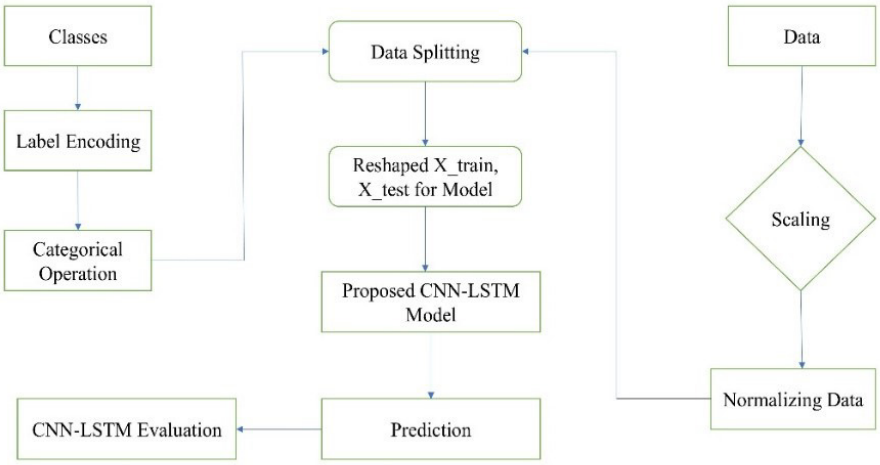

Figure 1 Data Preprocessing Method

Figure 1 illustrates the data preprocessing method of the proposed CNN-LSTM model. First, the data must be scaled and normalized for optimal model operation. The classes of intrusion types undergo a label encoding process to convert them into numerical form. Following this conversion, a categorical operation is performed on the label-encoded data to facilitate multiclass classification. After normalizing the data and processing the classes, both datasets are combined for data splitting to prepare the training and testing sets. The training data needs to be reshaped for input into the hybrid CNN-LSTM model. The reshaped data, along with the categorical data, is provided to the proposed model. Once the model is trained, it performs predictions on the testing data. Evaluation metrics are

then applied to assess the performance of the CNN-LSTM model. This model is designed to predict each class of intrusion type separately, enabling effective multiclass classification.

Table 1 Proposed Model CNN-LSTM Parameters

Optimizer	**Adam**
Metrics	Categorical Accuracy
Loss Function	Categorical Cross Entropy
Batch Size	3000
LSTM Neuron	50
Output Function	SoftMax
Validation Split	0.3
Conv1d Filters	64

Table 1 presents the various parameters used in the proposed CNN-LSTM model during the training process. It includes details on optimizers, metrics, loss functions, batch size, epochs, output functions, and validation splits, along with their respective values. All parameters are optimized for multiclass performance. Notably, the Conv1D filter value is set to 64.

4. Experiment

The experiment was conducted on a Core i7 seventh-generation system with an Intel graphics card. The proposed model was trained on two multiclass datasets: the Wireless Sensor dataset, which contains five classes of malicious activity, and the Microsoft Malware dataset, which includes ten classes. The performance of the presented architecture is compared with various algorithms to assess model efficiency.

1) Accuracy

$$\text{Accuracy} = \frac{\text{T-P+T-N}}{\text{TP+FP+FN+TN}}$$

2) Precision

$$\text{Precision} = \frac{\text{T-P}}{\text{TP+FP}}$$

3) Recall

$$\text{Recall} = \frac{\text{T-P}}{\text{TP+FN}}$$

4) F1 Score

$$\text{F1\_Score} = 2\,\frac{\text{Precision.Recall}}{\text{Precision + Recall}}$$

5) Confusion Matrix

$$\begin{bmatrix} \text{TP} & \text{FP} \\ \text{FN} & \text{TN} \end{bmatrix}$$

4.1 Wireless Sensor Network Dataset

The Wireless Sensor Network dataset contains 374,661 samples representing five families of malicious network intrusions. Following the experiment, the proposed model demonstrated its results based on this dataset, which includes 18 features.

Table 2 Number of Classes in Wireless Sensor Network Dataset

No of classes	Intrusion-Type
Normal	340066
Black-Hole	10049
Gray-Hole	14596
Flooding	3312
TDMA	6638

Table 2 shows the number of classes of intrusion types in the WSN dataset. It contains the 5 number of classes.

Table 3 Proposed Model Performance on Wireless Sensor Network Dataset

Model	Dataset	Accuracy	Precision	F1 Score	Recall
Proposed Model	WSN-Dataset	0.996	0.996	0.996	0.996
LSTM-Model	WSN-Dataset	0.988	0.971	0.971	0.972
GRU-Model	WSN-Dataset	0.988	0.972	0.974	0.971
KNN-Model	WSN-Dataset	0.981	0.981	0.981	0.981

Table 3 displays the performance of the proposed model on the Wireless Sensor Network dataset, which is a multiclass dataset. It includes metrics such as accuracy, precision, recall, and F1 score. The model demonstrates the best performance on this multiclass dataset. Comparisons are made with various deep learning models, including LSTM, GRU, and KNN algorithms.

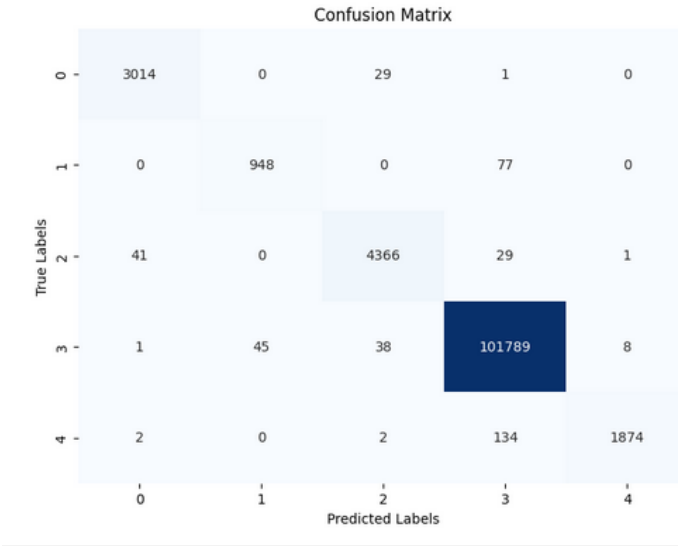

Figure 2 Confusion-Matrix by Proposed Model on WSN Dataset

Figure 2 presents the confusion matrix generated by the proposed model using the Wireless Sensor Network dataset. After making predictions, the model displays the results for each class, showing the samples corresponding to each type of intrusion.

4.2. Microsoft Malware Dataset

The Microsoft Malware dataset contains 10,868 samples, representing 10 classes. The experiment conducted using the proposed model shows the performance, detailed below.

Table 4 Proposed Model Performance on Microsoft Malware Dataset

Model	Dataset	Accuracy	Precision	F1 Score	Recall
Proposed Model	Microsoft Malware Dataset	0.985	0.985	0.985	0.985
LSTM Model	Microsoft Malware Dataset	0.966	0.810	0.812	0.830
GRU Model	Microsoft Malware Dataset	0.973	0871	0.872	0.873
KNN-Model	Microsoft Malware Dataset	0.970	0.971	0.970	0.972

Table 4 displays the performance of the proposed model on the Microsoft Malware dataset, which is a multiclass dataset. It includes metrics such as accuracy, precision, recall, and F1 score. The model demonstrates the best performance on this multiclass dataset.

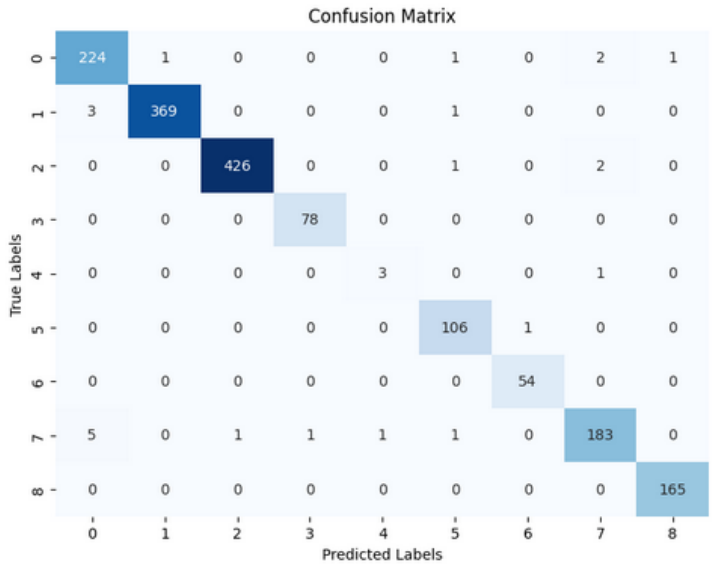

Figure 3 Confusion-Matrix by Proposed Model on Microsoft Malware Dataset

Figure 3 presents the confusion matrix generated by the proposed model using the Microsoft Malware dataset. After making predictions, the model displays the results for each class, showing the samples corresponding to each type of intrusion.

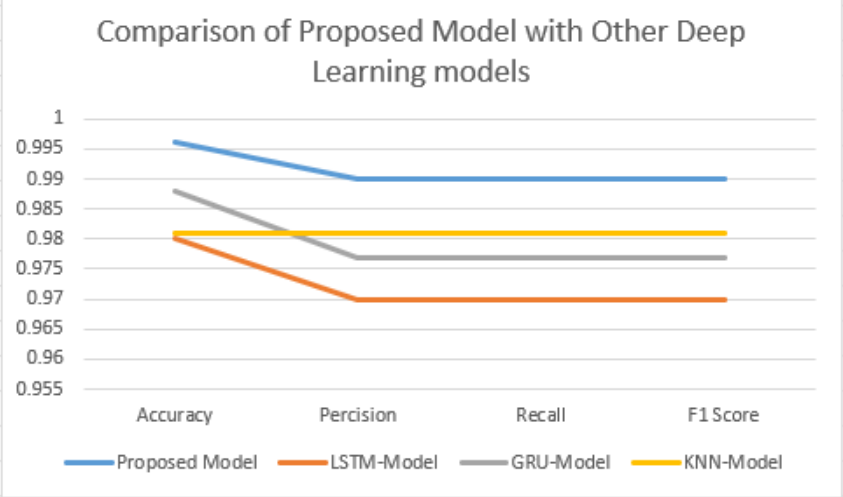

Figure 4 Proposed Model Comparison with WSN Dataset

Figure 4 compares the proposed model with different deep learning models in terms of accuracy, precision, F1 score, and recall as used in the experiment. It includes the proposed CNN-LSTM alongside comparative models, which are LSTM, GRU, and KNN. This graph displays the results for the Wireless Sensor Network (WSN) dataset.

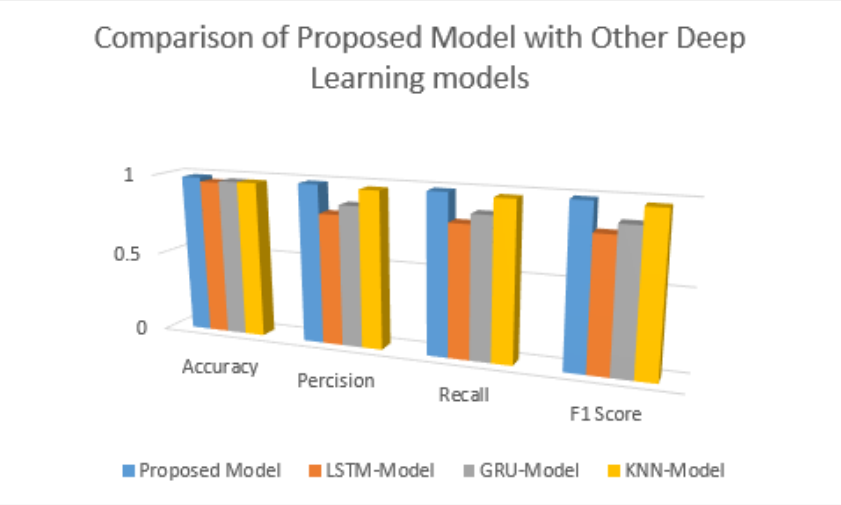

Figure 5 Proposed Model Comparison with Microsoft Malware Dataset

Figure 5 compares the proposed model with different deep learning models in terms of accuracy, precision, F1 score, and recall as used in the experiment. It includes the proposed CNN-LSTM alongside comparative models, which are LSTM, GRU, and KNN. This graph displays the results for the Microsoft Malware dataset.

5. Discussion

The hybrid CNN-LSTM model demonstrates an accuracy of 99.6% on the Wireless Sensor Network (WSN) dataset, which contains five classes. On the Microsoft Malware dataset, the model achieves 98.5% accuracy in a multiclass setting with nine classes. This model is specifically designed to handle multiclass datasets by integrating two deep learning architectures: CNN and LSTM. Additionally, various techniques, such as attention mechanisms and residual connections, are employed to enhance the model's performance. The model is finely tuned with different parameters, including activation functions and optimizers. In the experiment, two multiclass datasets are used to compare the proposed model with other deep learning models, including LSTM, GRU, and a machine learning model (KNN). Different metrics, such as accuracy, precision, recall, and confusion matrix, are utilized to evaluate the model's performance and accurately predict each class. Graphs illustrating the proposed

model's metrics alongside those of different deep learning models are also presented for both datasets, showcasing the superior results of the hybrid model. After applying various neural networks to these datasets, the proposed model consistently achieves the best accuracy compared to the others. This method for predicting network intrusion is well-suited for modern applications, as the hybrid deep learning approach significantly improves both prediction time and accuracy.

6. Conclusion

The proposed CNN-LSTM model demonstrates the best performance on two multiclass datasets for network intrusion detection. Built using an attention mechanism and residual connections, the model enhances efficiency in complex tasks and provides accurate predictions for each class. It achieves 99.6% accuracy on the Wireless Sensor Network (WSN) dataset and 98.5% accuracy on the Microsoft Malware dataset, indicating that the model is reliable for multiclass predictions of network intrusion. After utilizing various metrics in the experiment, the model shows excellent performance, reflecting its effectiveness in prediction. The comparison with other deep learning models, including GRU, LSTM, and KNN, reveals that our proposed model outperforms the others.

7. Future

The model should be trained on a larger dataset to accommodate a greater number of classes. Additionally, feature selection techniques should be applied to extract useful features that enhance performance. To improve the proposed model, additional layers can be integrated to facilitate more effective network intrusion predictions. Furthermore, the model should be implemented in an
in an Intrusion Detection System (IDS) to prevent network-based intrusions while ensuring fast performance.

References

1. L. Nenov, K. Kassev, and D. Chanev, “Investigation of algorithms for virus detection using neural networks and machine learning,” 2021 6th Jr. Conf. Light. Light. 2021 - Proc., pp. 5–8, 2021, doi: 10.1109/Lighting49406.2021.9599087.

2. S. Iqbal, A. Ullah, S. Adlan, and A. R. Soobhany, “Malware Prediction Using LSTM Networks,” Lect. Notes Networks Syst., vol. 350, pp. 583–604, 2022, doi: 10.1007/978-981-16-7618-5_51.

3. Y. Wu, D. Wei, and J. Feng, “Network attacks detection methods based on deep learning

techniques: A survey," Secur. Commun. Networks, vol. 2020, 2020, doi: 10.1155/2020/8872923.

4. A. P. Singh, "Encrypted Malware Detection Methodology without Decryption using Deep Learning based Approaches," 2022.

5. S. Chaudhari, A. Kamthe, M. Kudmethe, N. Barapatre, and S. Bahenwar, "Detection of SMS Fraudulent using ANN Algorithm .," vol. 24, no. 2, 2020.

6. B. Cao, C. Li, Y. Song, Y. Qin, and C. Chen, "Network Intrusion Detection Model Based on CNN and GRU," Appl. Sci., vol. 12, no. 9, 2022, doi: 10.3390/app12094184.

7. M. Maithem and G. A. Al-Sultany, "Network intrusion detection system using deep neural networks," J. Phys. Conf. Ser., vol. 1804, no. 1, pp. 113–128, 2021, doi: 10.1088/1742-6596/1804/1/012138.

8. C. Joshi, R. Kumar, and V. Bharti, "A Fuzzy Logic based feature engineering approach for Botnet detection using ANN," J. King Saud Univ. - Comput. Inf. Sci., no. xxxx, 2021, doi: 10.1016/j.jksuci.2021.06.018.

9. Y.-H. Chen, J.-L. Chen, and R.-F. Deng, "Similarity-Based Malware Classification Using Graph Neural Networks," Appl. Sci., vol. 12, no. 21, p. 10837, 2022, doi: 10.3390/app122110837.

10. A. K. Balyan et al., "A Hybrid Intrusion Detection Model Using EGA-PSO and Improved Random Forest Method," Sensors, vol. 22, no. 16, pp. 1–20, 2022, doi: 10.3390/s22165986.

11. E. Venkata Pawan Kalyan, A. Purushottam Adarsh, S. Sai Likith Reddy, and P. Renjith, "Detection Of Malware Using CNN," 2022 2nd Int. Conf. Comput. Sci. Eng. Appl. ICCSEA 2022, 2022, doi: 10.1109/ICCSEA54677.2022.9936225.

12. M. Schofield et al., "Convolutional Neural Network for Malware Classification Based on API Call Sequence," pp. 85–98, 2021, doi: 10.5121/csit.2021.110106.

13. M. S. Akhtar and T. Feng, "Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time," Symmetry (Basel)., vol. 14, no. 11, 2022, doi: 10.3390/sym14112308.

14. J. Wang, S. Wang, and Y. Wang, "Malware Classification based on a Light-weight Architecture of CNN : MalShuffleNet," pp. 2–5, 2022.

15. M. Khan, D. Baig, U. S. Khan, and A. Karim, "Malware Classification Framework using Convolutional Neural Network," 1st Annu. Int. Conf. Cyber Warf. Secur. ICCWS 2020 - Proc., 2020, doi: 10.1109/ICCWS48432.2020.9292384.

16. M. Kinkead, S. Millar, N. McLaughlin, and P. O'Kane, "Towards explainable cnns for android malware detection," Procedia Comput. Sci., vol. 184, no. 2019, pp. 959–965, 2021, doi: 10.1016/j.procs.2021.03.118.

17. M. Ahmed, N. Afreen, M. Ahmed, M. Sameer, and J. Ahamed, “An inception V3 approach for malware classification using machine learning and transfer learning,” Int. J. Intell. Networks, vol. 4, no. September 2022, pp. 11–18, 2023, doi: 10.1016/j.ijin.2022.11.005.

18. K. Gupta, N. Jiwani, M. H. U. Sharif, R. Datta, and N. Afreen, “A Neural Network Approach For Malware Classification,” 3rd IEEE 2022 Int. Conf. Comput. Commun. Intell. Syst. ICCCIS 2022, pp. 681–684, 2022, doi: 10.1109/ICCCIS56430.2022.10037653.

19. S. Ann, “Network Intrusion Detection Using Stack-Ensemble ANN,” pp. 1104–1109, 2022.

20. P. Kotian and R. Sonkusare, “Detection of Malware in Cloud Environment using Deep Neural Network,” 2021 6th Int. Conf. Converg. Technol. I2CT 2021, pp. 1–5, 2021, doi: 10.1109/I2CT51068.2021.9417901.

21. M. A. Talukder, S. Sharmin, M. A. Uddin, M. M. Islam, and S. Aryal, “MLSTL-WSN: machine learning-based intrusion detection using SMOTETomek in WSNs,” Int. J. Inf. Secur., vol. 23, no. 3, pp. 2139–2158, 2024, doi: 10.1007/s10207-024-00833-z.

22. W. Ma, C. Gou, and Y. Hou, “Research on Adaptive 1DCNN Network Intrusion Detection Technology Based on BSGM Mixed Sampling,” Sensors, vol. 23, no. 13, 2023, doi: 10.3390/s23136206.

23. Y. S. Almutairi, B. Alhazmi, and A. A. Munshi, “Network Intrusion Detection Using Machine Learning Techniques,” Adv. Sci. Technol. Res. J., vol. 16, no. 3, pp. 193–206, 2022, doi: 10.12913/22998624/149934.