Hybrid Deep Learning Model for Enhanced Intrusion Detection

Ismail Sanaya Muhammad[1*], Yusuf Musa Malgwi[2]

[1]Department of Computer Science, Faculty of Computing and Artificial Intelligence, Taraba State University, Jalingo, Nigeria

[2]Department of Computer Science, Modibbo Adamawa University, Yola, Nigeria

[*]Corresponding Author

DOI: https://dx.doi.org/10.47772/IJRISS.2026.10100129

Received: 05 January 2026; Accepted: 10 January 2026; Published: 24 January 2026

ABSTRACT

The rapid growth of cyberattacks, especially Distributed Denial of Service (DDoS), has exposed the limitations of conventional Intrusion Detection System (IDS). These systems often struggle to cope with evolving attack strategies. In recent years, deep learning has provided new opportunities for improving IDS, as it can automatically discover hidden structures in complex data without extensive manual feature engineering. This study develops and evaluates three models, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and a Hybrid CNN-LSTM for intrusion detection using the CIC-DDoS2019 dataset. Preprocessing involved normalization, label encoding, and class balancing using Synthetic Minority Oversampling Technique (SMOTE). Feature selection was carried out using the information gain algorithm performance, the models were trained and evaluated using key metrics such as accuracy, precision, recall, f1-score and Area Under the Curve (AUC) to improve model performance. Experimental results shows that CNN achieved an accuracy of 99.94%, while LSTM performed slightly better with 99.96%, the hybrid CNN-LSTM outperformed both with 99.97% accuracy, precision, and recall, confirming that combining CNN's spatial learning with LSTM's temporal sequence modeling leads to superior detection. This study highlights the advantage of hybrid deep learning in network security, reducing both false positives and false negatives. It also provides a practical framework for building IDS capable of adapting to modern attack patterns. Future extensions could focus on real-time implementation, multi-class detection of different attack categories, and explainable AI for improved transparency.

Keywords: CIC-DDoS2019, Convolutional Neural Network (CNN), Hybrid CNN-LSTM, Intrusion Detection System (IDS), Long Short-Term Memory (LSTM).

INTRUDUCTION

In today's digitally interconnected world, information exchange and online communication are vital for economic growth and social interaction. However, this dependence on technology also exposes organizations to cybersecurity risks (Vevera & Botezatu, 2023). As digital infrastructures expand, they create opportunities for innovation but simultaneously increase vulnerabilities. Cybercriminals are now using advanced tactics to compromise data, disrupt services, and infiltrate networks.

Intrusion Detection Systems (IDS) play an important role in protecting networks by continuously monitoring traffic, identifying abnormal patterns, and alerting administrators when suspicious activities occur (Efe & Abacı, 2022). Nevertheless, traditional IDS models, which mostly rely on signatures or predefined rules, are struggling to keep up with constantly evolving threats. These systems often fail to recognize new or complex attack types and also find it difficult to process the massive amounts of data generated by modern networks.

Feature selection is another critical aspect in developing IDS. High-dimensional datasets often contain irrelevant or redundant attributes that may reduce accuracy or cause overfitting. Techniques such as Information Gain (IG) help select the most useful features, enabling the model to achieve better detection performance (Almotairi et

al., 2024; Dhawas et al., 2024). Furthermore, IDS performance is usually measured through detection accuracy, false positive rate, false negative rate, and evaluation metrics such as the Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC), which assess the model's ability to correctly classify malicious and benign activities (Olateju et al., 2024).

This study builds on these advancements by designing an IDS that integrates CNN and LSTM in a hybrid architecture with feature selection techniques, aiming to improve accuracy while minimizing false alarms.

The aim of this research was to improve intrusion detection in computer networks by applying deep learning models, particularly CNN, LSTM, and their hybrid integration, to strengthen cybersecurity systems.

The objectives are to:

1. Identify and select the most relevant features from the CIC-DDoS2019 dataset using the Information Gain algorithm.

2. Implement CNN, LSTM, and hybrid CNN-LSTM models for intrusion detection.

3. Evaluate the performance of these models using accuracy, precision, recall, F1-score, and ROC-AUC metrics.

4. Compare the effectiveness of the proposed models with existing IDS approaches to determine their accuracy and computational efficiency.

Machine Learning, Deep Learning, and Hybrid CNN-LSTM Approaches

The rapid growth of cyber threats and the complexity of network traffic have made intrusion detection a highly challenging task. Traditional machine learning approaches, while effective in some scenarios, rely heavily on manual feature engineering and tend to struggle with the vast scale and diversity of modern attack vectors. Deep learning (DL), as a subset of artificial intelligence, provides a more powerful solution by automatically learning features from raw data, capturing complex nonlinear relationships, and generalizing better to unseen threats (LeCun et al., 2015; Hindy et al., 2020). Over the last decade, DL-based models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Autoencoders, and hybrid architectures have been widely studied for Intrusion Detection Systems (IDS). This section reviews notable contributions and highlights their strengths, weaknesses, and implications for IDS design.

Kim et al. (2021) proposed a CNN-based IDS capable of detecting anomalies by learning patterns from raw traffic data. Their results demonstrated higher detection accuracy compared to traditional methods, largely because CNN was able to automatically extract hidden spatial features without requiring handcrafted inputs. Similarly, Yuan et al. (2021) designed a CNN model for detecting DDoS attacks in cloud computing environments. The system reached a detection accuracy of 98.3% on the CICIDS2017 dataset, showing CNN's effectiveness in large-scale network monitoring.

More recently, Aljawarneh et al. (2022) tested an LSTM-based system on the CICIDS2017 dataset and observed significant improvements in recall and F1-score compared to Random Forest and Support Vector Machines.

Their findings confirmed that LSTM models excel in classifying time-dependent traffic patterns, particularly in identifying slow and stealthy attacks that may not be detected by static models.

METHODOLOGY

In this study, it was essential to modify and develop the methodology in accordance with engineering principles. This study was to the development of a hybrid intrusion detection system that integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models. The combination leverages the strengths of both architectures: CNN is efficient at extracting spatial features from traffic data, while LSTM captures temporal dependencies as shown in Figure 1. To further improve performance, the model incorporates feature

selection using the Information Gain (IG) algorithm to reduce dimensionality and focus on the most relevant attributes.

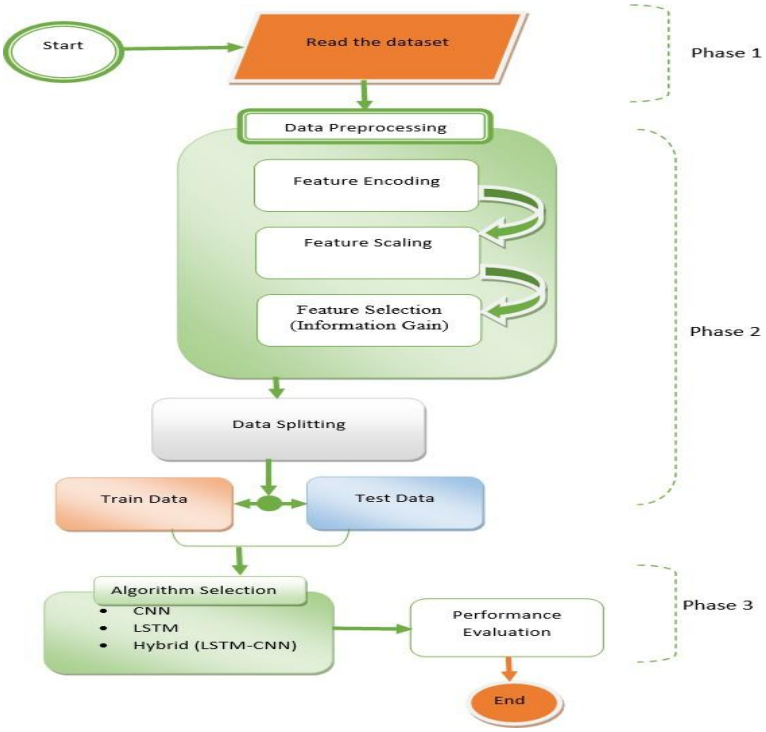

Figure 1: Research Methodology Mode

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are inspired by the structure and function of the animal visual cortex, enabling them to effectively learn and interpret sequential patterns within data. For the purpose of intrusion detection, a one-dimensional CNN (1D-CNN) was utilized to process textual input and uncover patterns indicative of malicious network activity. The 1D-CNN architecture comprises essential layers including convolutional, pooling, and fully connected layers. The design of the proposed model aligns with the structural framework illustrated in Figure 2.

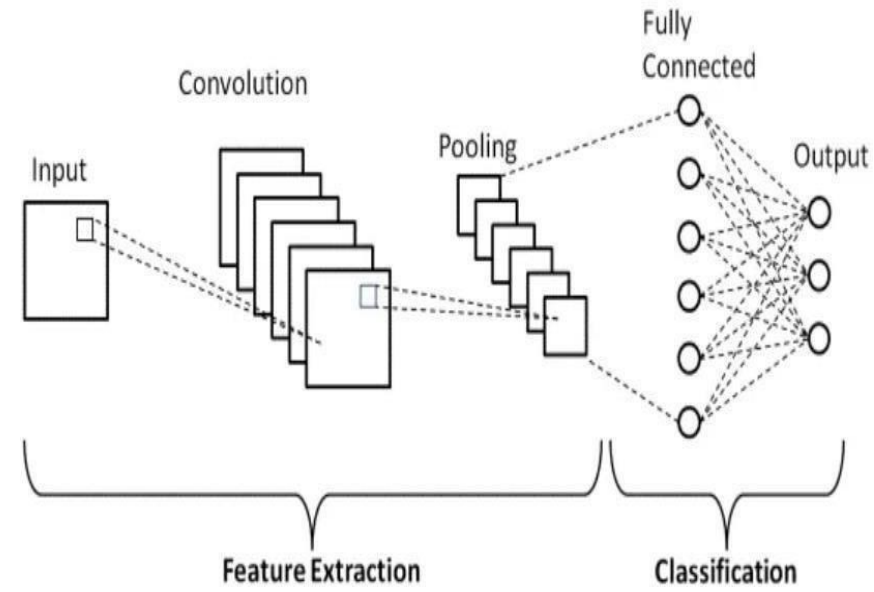

Figure 2: Cnn Architecture

Long Short-Term Memory (LSTM)

The LSTM architecture was composed of LSTM cells, which have three essential gates: the input gate,the forget gate, and the output gate. This architecture as prescribed in Figure 3 enables the LSTM to preserve critical information across extended sequences, efficiently capturing the dependencies essential for precise anomaly detection in network behavior. Subsequent to processing through the LSTM cells, the data was routed to a fully connected layer, which transforms the LSTM outputs into a classification-compatible format, typically yielding a binary decision on whether the incoming sequence signifies normal activity or a potential intrusion.

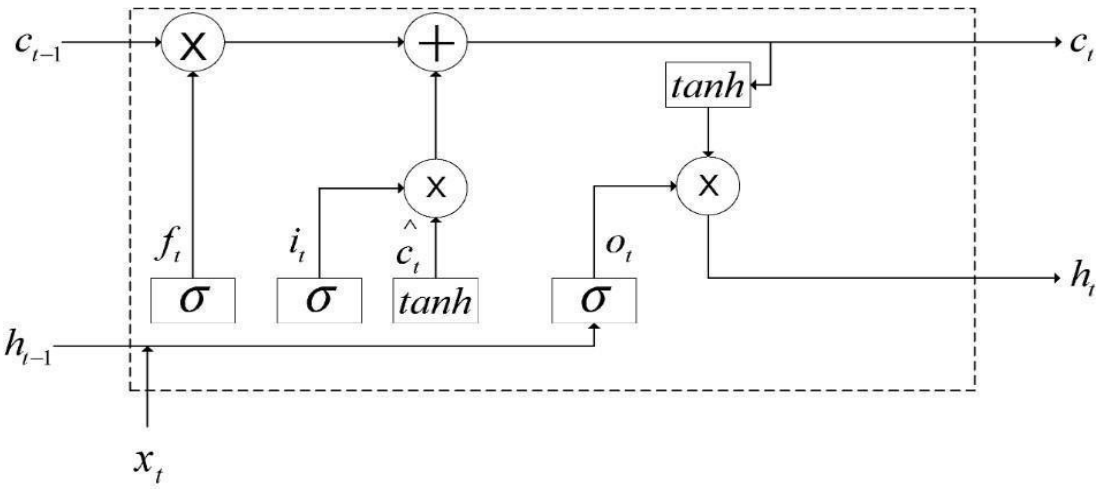

Figure 3: LSTM Architecture

Hybridization Process

To hybridize the CNN and LSTM algorithm for network intrusion detection utilizing the CIC DDoS2019 dataset from Kaggle, it is essential to comprehend the dataset's architecture and the requisite preprocessing procedures. The CIC-DDoS2019 dataset comprises diverse network traffic characteristics gathered during simulated DDoS assaults. This dataset has labelled records that signify both regular and malicious traffic, rendering it an exceptional resource for classification tasks. Hence, subsequent to preprocessing the dataset, feature extraction was conducted, which was essential for preparing the data for CNN input. Given that CNNs are proficient in spatial feature extraction, here the dataset was transformed into a three-dimensional tensor format of (samples, time steps, features). This transformation enables the study to utilize the sequential characteristics of network traffic data, allowing the model to discern temporal trends as shown in Figure 4.

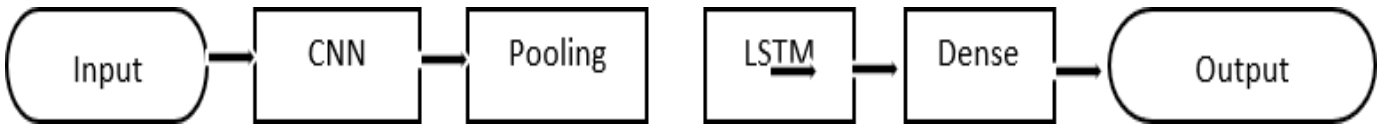

Figure 4: Hybrid CNN-LSTM Architecture

Upon establishing the hybrid model, it was imperative to integrate it with an appropriate optimizer, such as Adam, and a suitable loss function corresponding to the classification type.

Model training entails utilizing the training set while consistently evaluating performance measures on the validation set, such as accuracy, precision, recall, and F1-score. These measurements offer insights into the model's performance and inform further modifications to enhance its efficacy. Post-training, the model must be assessed on an independent test set to determine its capacity to generalize to novel, unseen data. Examining confusion matrices and classification reports facilitated the assessment of the model's performance across various traffic classes.

Ultimately, the integration of CNNs and LSTMs was targeted at developing a formidable framework for identifying anomalies in network traffic, facilitating a thorough examination of both acute traffic surges and prolonged trends that may signify probable DDoS attacks. The use of the CIC-DDoS2019 dataset, with its diverse traffic patterns and attack simulations, establishes a robust basis for improving the accuracy and reliability of network intrusion detection systems.

To evaluate the performance of the proposed model, several confusion matrix components were utilized. These include True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), with Recall also referred to as the True Positive Rate (TPR) being a central focus. Recall for a specific class is calculated by dividing the number of correctly identified instances within that class by the total number of actual instances belonging to it.

RESULTS

The dataset was read into the program code using the 'read-csv' module from the panda's library because the file format was CSV (Comma separated values). This was possible because pandas have the 'read-csv' function, which reads and displays the dataset's characteristics in a tabular cell format. Pandas was an additional Python library containing modules that can load data from a variety of sources. The used code snippets for accessing the dataset is depicted in Figure 5.

```
In [2]: df = pd.read_csv("./dataset/cicddos2019.csv")

In [3]: df.head()

Out[3]:
```

	Unnamed: 0	Protocol	Flow Duration	Total Fwd Packets	Total Backward Packets	Fwd Packets Length Total	Bwd Packets Length Total	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	...	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min
0	0	17	216631	6	0	2088.0	0.0	393.0	321.0	348.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	1	17	2	2	0	802.0	0.0	401.0	401.0	401.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	2	17	48	2	0	766.0	0.0	383.0	383.0	383.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	3	17	107319	4	0	1398.0	0.0	369.0	330.0	349.5	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	4	17	107271	4	0	1438.0	0.0	389.0	330.0	359.5	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 5: Read and Display the Dataset

Data Scaling

The purpose of data scaling in this study feature dominance was avoided because features with larger scales could dominate the learning process and potentially result in biased outcomes. As a result, equal weight was given to all of the features in the learning process when the dataset was scaled while also facilitating faster convergence and stable training for the CNN, LSTM and CNN-LSTM algorithms employed. Moreover, as some of the features within the dataset contain ranging units or data types, scaling was required to maintain a consistent scale. Therefore, for scaling down the numerical values to 1-unit scale the standard scaler library of the Sklearn library was utilized. Furthermore, for encoding the categorical features, the Label Encoder library was utilized as shown in Figure 6 and 7.

```
In [81]: # features scaling
         scaler = StandardScaler()
         x[x.columns]= scaler.fit_transform(x)
```

Figure 6: Data Scaling

```
In [12]: le =  LabelEncoder()
         for i in df:
             if df[i].dtype=='object':
                 df[i] = le.fit_transform(df[i])
             else:
                 continue
```

Figure 7: Label Encoding

Feature Selection

As aforementioned the study incorporates a feature selection technique using the information gain algorithm. The essence of the feature selection was to reduce the dataset dimensionality and further eradicate less correlated features in intrusion detection. The algorithm 'information gain' allows the specification of the number of features to be selected using a value called the k-threshold. Here, the value of k as a threshold was set to 0.01 during the experiment. The features were reduced from 80 to 55, Table 1 shows some of the features which were selected.

Table 1: Selected Features

S/№	Feature	Information Gain	S/№	Feature	Information Gain
1.	Label	0.709112	9.	Subflow Bwd Packets	0.323292
2.	Avg Packet Size	0.6261	10.	Init Fwd Win Bytes	0.311382
3.	Fwd Packet Length Mean	0.573798	11.	Flow IAT Min	0.301779
4.	Avg Fwd Segment Size	0.573758	12.	Down/Up Ratio	0.301492
5.	Fwd Packets Length Total	0.572012	13.	Subflow Fwd Packets	0.292666
6.	Packet Length Min	0.569434	14.	Total Fwd Packets	0.29169
7.	Fwd Packet Length Min	0.563304	15.	Bwd Packets Length Total	0.277634
8.	Subflow Fwd Bytes	0.560981	16.	Subflow Bwd Bytes	0.275578

In this study, the dataset was splited into training and test halves; the training set was utilized to train the CNN, LSTM, and CNN-LSTM algorithms, while the test set was utilized to assess the model's performance. Seventy percent of the dataset was utilized to train the models, while the remaining thirty percent was used to assess the models' performance.

Result Presentation

As per the goals of the current study, the performance of the models was assessed using the most crucial performance measures like accuracy, precision, recall, F1-score, and AUC (Area Under the Curve). The performance of the CNN, LSTM, and Hybrid CNN-LSTM models on the trained dataset is tabulated in Table 2 and Figure 8. The values of all the evaluation metrics are given as decimal points representing high levels of performance.

Table 2: Result Presentation

Model	Accuracy	Precision	Recall	F1-Score	AUC
CNN	0.9994	0.9994	0.9994	0.9994	0.9999
LSTM	0.9996	0.9996	0.9996	0.9996	0.9999
Hybrid CNN+LSTM	0.9997	0.9997	0.9997	0.9997	0.9999
Note: Precision, Recall, and F1-Score are macro averaged, representing performance across both classes (0 = no intrusion, 1 = intrusion)					

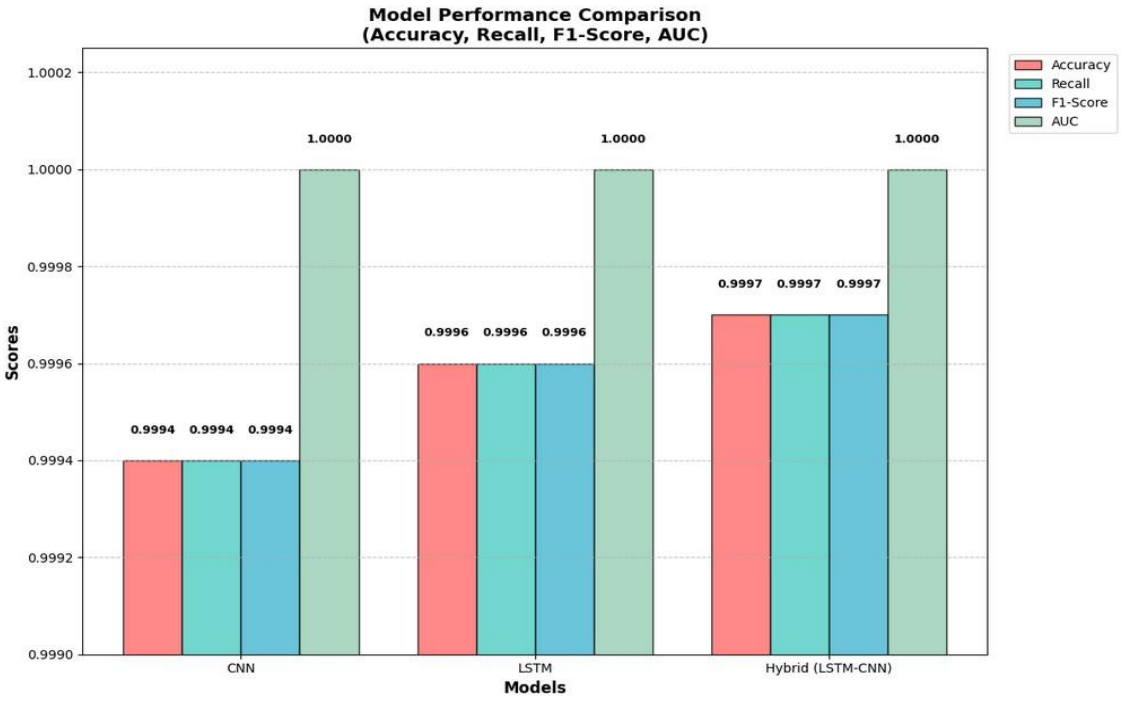

Figure 8: Models Performance Chart

confusion Matrix

In an attempt to rigorously investigate the performance of the model, the confusion matrices was also considered. The CNN, LSTM, and Hybrid CNN-LSTM model's confusion matrices as shown in Figure 9, 10, and 11.

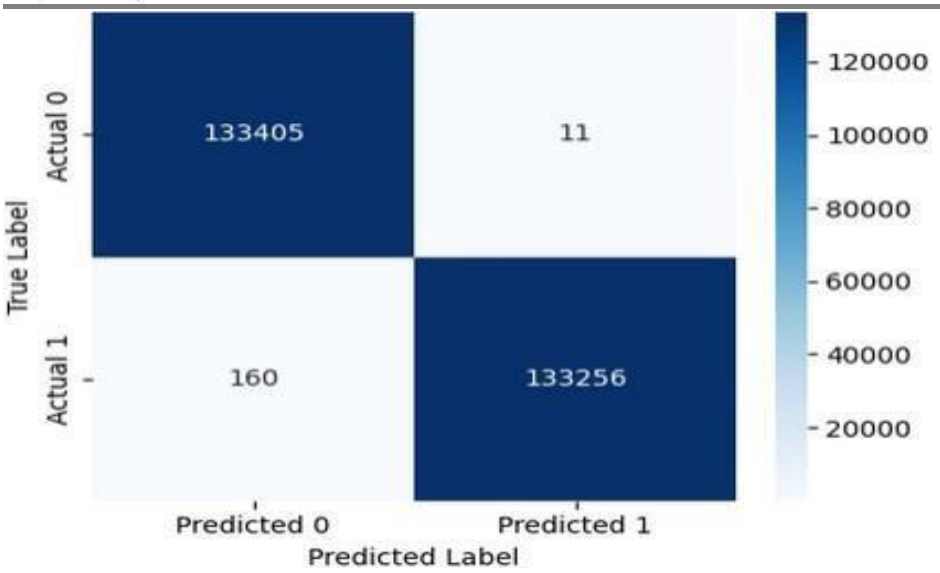

Figure 9: CNN Heat Map

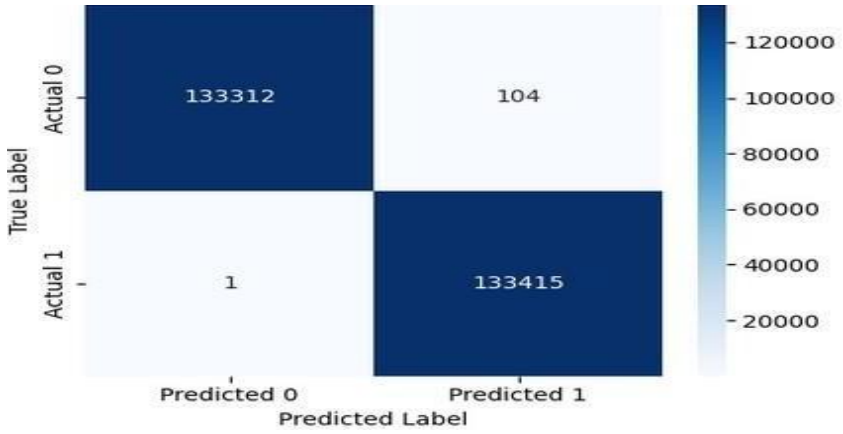

Figure 10: LSTM Heat Map

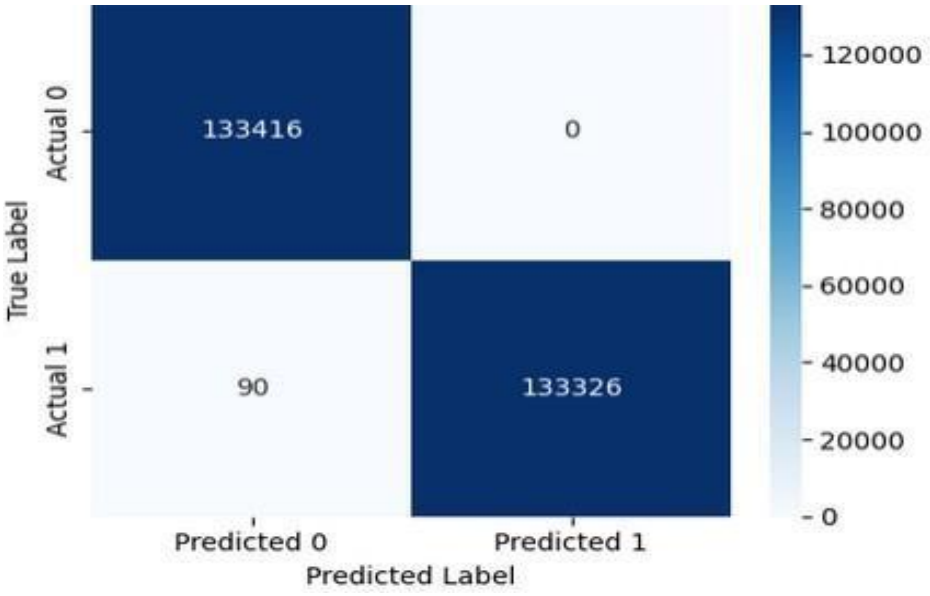

Figure 11: Cnn-Lstm Heat Map

State-Of-The-Art Comparison

Table 3 and Figure 12 present the state-of-the-art comparison between the three utilized algorithms and four other utilized algorithms in similar research by other authors. Five features are listed as columns in Table 10 for state-of-the-art comparison, i.e., authors, the dataset utilized, the utilized algorithm, the selected best algorithm by each author, and the respective scores in the last column. Performance evaluation of the used CNN, LSTM, and CNN-LSTM model was conducted on the selected features of dataset. In comparison with algorithms applied by the four other researchers in the research area of ML and DL, the CNN-LSTM model applied in this study performed with maximum performance, i.e., 99.97% accuracy. A graphical chart for a comparison of stateof-the-art algorithms.

Table 3: State-of-the-Art Comparison

Authors	Dataset Used	Algorithm Used	Best Algorithm	Accuracy (%)
Olugbenga et al. (2024)	CIC-DDoS2019	DT, LR, NB, AdaBoost, Gradient Boosting	DT	99.47
Alghazzawi et al. (2021)	CIC-DDoS2019	CNN, BI-LSTM, CNN-BI-LSTM	CNN-BI-LSTM	94.52
Bolodurina et al. (2020)	CIC-DDoS2019	SVM, RF and GBM	RF	94.81
Current Study	CIC-DDoS2019	LSTM, CNN, CNN-LSTM	CNN-LSTM	99.97

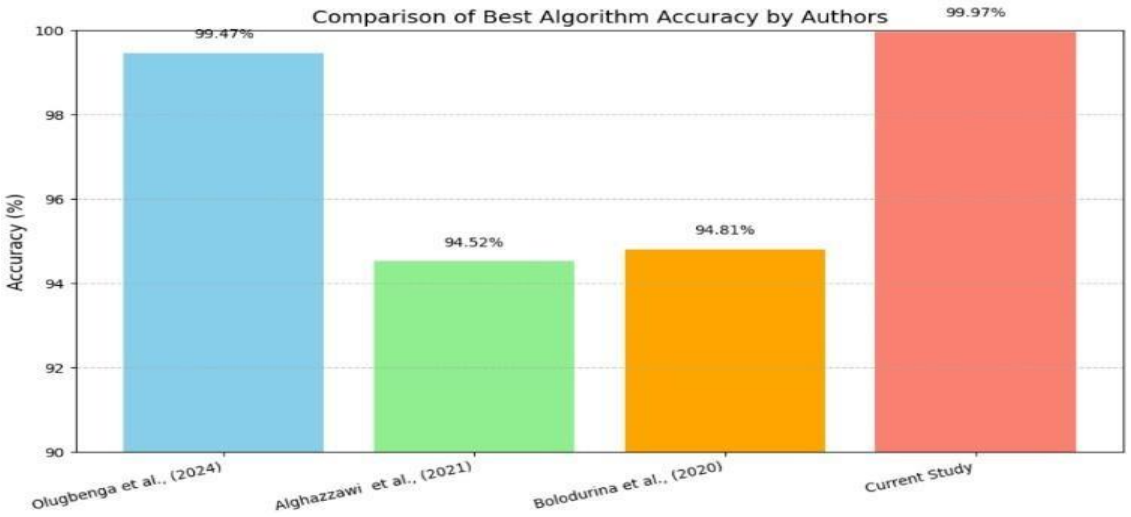

Figure 12: Comparison Of State-Of-The-Art Algorithms Using Same Dataset Based On % Of Accuracy Difference.

DISCUSSION

A lot of research has concentrated on the general topic of cybersecurity with particular interest in network threat detection and mitigation. Network security has remained important since the introduction of computer systems and the advancement of networking technology. As there have been improvements in technology, the adversaries have also become more intelligent, designing more sophisticated and intelligent network security threats. This ever-changing environment has prompted a large body of research that seeks to create measures for safeguarding

people and organizations from network intrusion and security breaches. This in-depth research examines intrusion detection model as an essential part of cybersecurity to address the immediate necessity of safeguarding data and systems.

In this study, machine learning methods, i.e., CNN, LSTM, and Hybrid CNN-LSTM, were utilized to present a viable solution to the issue of network threats in network intrusion detection. The study utilized feature selection methods, the information gain algorithm, to select the most effective features for efficient threat detection. The models were trained and tested on the CIC-DDoS2019 dataset. The research process consisted of three phases, the data preparation which comprises, preprocessing the dataset by imputing missing values, removing duplicate characters, normalizing numerical data, and encoding categorical attributes. Feature selection technique using the information gain approach was carried out to achieve dimensionality reduction and improve detection accuracy. Secondly, the preprocessed and normalized data was divided into training and testing datasets in the ratio 70:30. The three machine learning models CNN, LSTM, and Hybrid CNN-LSTM were trained and tested on the data. Detection performance of all the models was examined using classification measures precision, recall, F1-score, and accuracy. And lastly a graph comparison study of the models was done to determine the optimal algorithm. The analysis showed that Hybrid CNN-LSTM performed better than the other CNN and LSTM with 99.97% accuracy, demonstrating their better capability to distinguish between normal and attack traffic. But they also worked fine, albeit with a little less accuracy.

The study was implemented in the Python programming language, aided by external libraries that include NumPy, Pandas, Sklearn, and Matplotlib. These utilities enabled effective data preprocessing, model training, and performance evaluation. The study demonstrates the capacity of machine learning solutions to improve network security and effectively counter intrusion threats.

CONCLUSION

This study highlights the effectiveness of machine learning techniques for intrusion detection using the CIC-DDoS2019 dataset. Three models CNN, LSTM, and Hybrid CNN-LSTM were tested and evaluated through classification metrics, confusion matrices, and AUC scores. Among them, the hybrid CNN-LSTM consistently delivered the best results, achieving 99.97% accuracy, precision, recall, and F1-score. Its low rate of false positives and false negatives underscores its reliability in distinguishing between normal and malicious traffic.

When benchmarked against existing state-of-the-art methods, the Hybrid CNN-LSTM model outperformed previously reported results, confirming its robustness in handling complex attack scenarios. These findings validate Hybrid CNN-LSTM as the most effective model in this study and emphasize the promise of machine learning for strengthening cybersecurity defenses and mitigating network-based threats.

REFERENCES

1. Alghazzawi, D., Bamasag, O., Ullah, H., & Asghar, M. Z. (2021). Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection. Applied Sciences, 11(24), 11634.
2. Aljawarneh, S., Alsharif, N., & Alsharif, L. (2022). An intelligent intrusion detection system using deep learning for cloud environments. Journal of Information Security and Applications, 65, 103130. https://doi.org/10.1016/j.jisa.2021.103130
3. Almotairi, A., Atawneh, S., Khashan, O. A., & Khafajah, N. M. (2024). Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. Systems Science & Control Engineering, 12(1), 2321381.
4. Bolodurina, I., Shukhman, A., Parfenov, D., Zhigalov, A., & Zabrodina, L. (2020). Investigation of the problem of classifying unbalanced datasets in identifying distributed denial of service attacks. In Journal of Physics: Conference Series 1679 (4), 042020.
5. Dhawas, P., Ramteke, M. A., Thakur, A., Polshetwar, P. V., Salunkhe, R. V., & Bhagat, D. (2024). Big Data Analysis Techniques: Data Preprocessing Techniques, Data Mining Techniques, Machine Learning Algorithm, Visualization. In Big Data Analytics Techniques for Market Intelligence, 183-208.
6. Efe, A., & Abacı, İ. N. (2022). Comparison of the host-based intrusion detection systems and networkbased intrusion detection systems. Celal Bayar University Journal of Science, 18(1), 23-32.

7. Hindy, H., Atkinson, R., Tachtatzis, C., Colin, J.-N., Bayne, E., & Bellekens, X. (2020). Utilising deep learning techniques for effective zero-day attack detection. arXiv. https://arxiv.org/abs/2006.15344
8. Kim, J., Kim, J., Kim, H., Shim, M., & Choi, E. (2021). CNN-based network intrusion detection against denial-of-service attacks. Electronics, 9(6), 916. https://doi.org/10.3390/electronics9060916
9. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444 https://doi.org/10.1038/nature14539
10. Olateju, O., Okon, S. U., Igwenagu, U., Salami, A. A., Oladoyinbo, T. O., & Olaniyi, O. O. (2024). Combating the challenges of false positives in AI-driven anomaly detection systems and enhancing data security in the cloud. Available at SSRN 4859958.
11. Olugbenga, A. M., Adenike, A. E., Esther, O. O., Michael, A. A., Samsudeen, B. O., & Ibrahim, A. O. (2024). Development of a Distributed Denial of Service Detection Model Using Ensemble Machine Learning Techniques. Adeleke University Journal of Science, 3(1), 205-218.
12. Vevera, A. V., & Botezatu, U. E. (2023). Hyperconnected horizons: decoding the digital sovereignty of European smart cities. In Smart Cities International Conference (SCIC) Proceedings, 11, 393-400.
13. Yuan, X., AlSaleh, I., Al-Samawi, A., & Nissirat, L. (2021). Novel machine learning approach for DDoS cloud detection: Bayesian-based CNN and data fusion enhancements. Sensors, 24(5), 1418. https://doi.org/10.3390/s24051418