ISSN: 2230-9926

Available online at http://www.journalijdr.com

IJDR

International Journal of Development Research
Vol. 15, Issue, 11 pp. 69519-69524, November, 2025
https://doi.org/10.37118/ijdr.30250.11.2025

RESEARCH ARTICLE OPEN ACCESS

A DEEP LEARNING BASED INTRUSION DETECTION MODEL IN 5G ENVIRONMENT USING LSTM NETWORK AND HYPER PARAMETER TUNING

Marafa Salman Ibrahim[1], Micheal Olumuyiwa Odunsi[2], Barrie Amadu Wurie[1] and Samuel Idriss Kargbo[1]

[1]NKU - Nankai University, Binhai New Area, Teda, Tianjin, China
[2]Amazon, Durham, NC, USA

ARTICLE INFO

Article History:

Received 19th August, 2025
Received in revised form
20th September, 2025
Accepted 09th October, 2025
Published online 30th November, 2025

KeyWords:

Long Short-Term Memory Network, Intrusion detection system, deep learning, fifth generation network, hyperparameter tuning.

Correspondingauthor:
Marafa S. Ibrahim

ABSTRACT

In the fifth generation (5G) network environment, high-speed data transmission and low latency are critical. Robust intrusion detection not only helps enhance network security but also aids in maintaining efficient and uninterrupted data flow. Although existing IDS models utilizing hybrid and non-hybrid classification techniques have improved the accuracy of intrusion detection, the complexity of modern network intrusions requires the use of more advanced machine learning methods to analyze complicated network traffic patterns and distinguish subtle anomalies from legitimate behavior, This paper evaluates the effectiveness of Long Short-Term Memory (LSTM) networks in intrusion detection based on the CICIDS2018 dataset, aiming to measure its performance in predicting network intrusion behaviors through metrics such as accuracy, precision, recall, and F1 score. Experimental results indicate that LSTM achieves an accuracy of 98% in identifying intrusion patterns by leveraging the temporal dependence of network traffic features. Furthermore, enabling hyperparameter experiments show that the model maintains stable performance across all evaluation metrics and exhibits excellent robustness under optimized conditions. These findings underscore the LSTM model's potential as a real-time intrusion detection solution in dynamic 5G environments. As network threats become increasingly complex, integrating deep learning technologies like LSTM into the IDS framework can significantly enhance real-time threat mitigation capabilities. This study points out the need for further development of adaptive IDS models to address the evolving network security challenges. Future research should focus on optimizing the computational efficiency and generalization ability of such models to ensure their scalability across diverse network architectures.

Copyright©2025, Marafa Salman Ibrahim et al. *This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.*

Citation: Marafa Salman Ibrahim, Micheal Olumuyiwa Odunsi, Barrie Amadu Wurie, and Samuel Idriss Kargbo. 2025. *"*A Deep Learning Based Intrusion Detection Model in 5G Environment Using LSTM Network and Hyper Parameter Tuning.*". International Journal of Development Research*, 15, (11), 69519-69524.

INTRODUCTION

Due to the speed at which networking technologies are developing and the growing frequency of cyber threats, maintaining strong cyber security has become crucial. The ability to identify and stop hostile activity and unauthorized access within computer networks is a crucial part of cyber security. However, without robust security measures, ICT systems remain vulnerable to potential breaches and threats, underscoring the need for mechanisms to counter intrusions and attacks effectively. Usually, securing computer networks from unauthorized access and malicious actions relies heavily on intrusion detection. Conventional IDS solutions often depend on signature matching, which restricts their effectiveness against new and evolving cyber threats.Moreover, anomaly detection systems face challenges, including the complexity of defining rules for each protocol under analysis and evaluating accuracy. In addition, unrecognized patterns can potentially indicate harmful activities. (IDS) on the other hand offer significant advantages, including the ability to identify emerging threats and their adaptability through proper training using datasets with intermittent distributions. These qualities make IDS a fundamental tool for ensuring network security. By enabling real-time and adaptive monitoring, intrusion detection is essential for maintaining robust network security. Additionally. IDS are specifically designed to identify issues and vulnerabilities within target hosts. Their primary objective is to detect threats and operate out-of-band within a network architecture, avoiding interference with data transmission between senders and receivers. Despite this configuration, analyzers usually evaluate duplicates of inline traffic streams to predict potential attacks based on pre-developed algorithms, therefore reducing the need for manual intervention(LeCun,Bengio and Hinton, 2015).By taking advantage of weaknesses brought about by poor design, faulty programming, or neglect on the part of system administrators, intruders might obtain access. The IDS analyzes numerous data audits and classifies them as either malicious or normal activity.

The quality of how intrusion detection technology used has a significant impact on an IDS's efficacy.

Research Questions or Objectives

- What are the specific security challenges faced by IDS systems. What are the limitations of existing intrusion detection techniques deployed in networking domain?
- What design strategies can be applied in building a DL-based AI NIDS that is capable of accurately detecting and responding to intrusions in 5G networks?
- How well does the suggested NIDS perform in terms of identifying network intrusions?
- What practical guidelines can be provided for the deployment and optimization of the NIDS models?

Common IDS Process Model

There are three basic functional components that can be used to characterize many IDS:Sources of Information: The different sources of event data utilized to ascertain the occurrence of an intrusion can originate from multiple levels within the system. The most commonly monitored sources include network activity, host systems, and application-level events.

Analysis: The component of intrusion detection systems responsible for organizing and interpreting events gathered from various information sources determines whether these events indicate ongoing or past intrusions.

Response: There are two types of actions the system takes in response to an intrusion detection: active measures and passive measures. Active measures involve automated interventions carried out by the system, whereas passive measures entail reporting the intrusion detection system's findings to human operators, who are then responsible for taking appropriate action based on the reports.

Intrusion data sources: IDS can be categorized not only by their detection methodologies but also by the type of data sources they rely on to identify unusual activities. Generally, IDS technology falls into two main categories depending on the sources of the data input. HIDS primarily analyzes data collected from various system logs, including operating system logs, Windows server logs, and storage logs. This approach makes HIDS particularly effective at identifying insider threats which do not generate network traffic, allowing it to identify suspicious activities that might otherwise go unnoticed according to Creech and Hu (2014).

IDS Structure: The structures of IDS refer to arrangement of its functional components in relation to one another. The IDS architecture consists of two primary components: the host, which operates the IDS software, and the target, which represents the system being monitored for potential security threat.

Host-Target Co-location: In the early development of IDS, most were deployed directly on the systems they were designed to protect. This approach was largely influenced by the prevalence of mainframe systems and the expense at which computers were acquired, which made deploying a separate IDS system financially prohibitive. However, from a security perspective, this poses a significant risk, as a successful breach of the target system could allow an attacker to disable the IDS as part of their overall attack strategy.

Host-Target Separation: IDS architects have been enhancing security features by modifying the system to operate independently of its original host. The ongoing developments in IDS architecture have led to a design where the control and analysis components run on separate machines. This architectural shift has not only improved the overall functionality of IDS but has also enhanced its security. By distributing these components, it becomes increasingly hard for intruders to identify whether an IDS is available. Additionally, as the number of workstations and personal computers continues to increase, this approach further strengthens the system's ability to remain concealed while effectively monitoring network activities.

Related Study: This study explores various deep learning approaches by looking at and analyzing a number of experiments. Understanding the latest developments and difficulties in the using deep learning for cyber security. The CIC-IDS2017 and NSL-KDD datasets was used to investigate how well their suggested model performed. According to results generated after experimenting, the model exhibits faster data preprocessing and possibly shorter training time, while also achieving a higher TPR for majority of the attack occurrences. Notably, the classification accuracy reached 99.91% for the multi-target, according to the CIC-IDS2017 dataset (Liu, Gu, and Wang, 2021). Both datasets offer valuable perspectives on this performance Using a BiLSTM network for learning temporal dependencies and a CNN to fetch spatial data, a deep hierarchical network model is built. CNN (Convolutional neural network) and LSTM networks are popular instances of these types of design. The efficiency of the suggested network intrusion detection method is confirmed by experimental assessments conducted on the NSL-KDD and UNSW-NB15 datasets.

According to findings, the system classification accuracy fell between 83.58% and 77.16% respectively by Jiang et al. (2020). Sharma et al. (2018) used an LSTM based detection model to detect botnet attacks on IoT networks. The suggested method shows excellent accuracy and detection rates by analyzing network traffic to identify botnet activity. The results highlight how important LSTM models are for enhancing IoT network security Here, a deep learning anomaly-based NIDS was presented. Alsulaimanand Alzaidi(2018) suggest a model that learns typical network behavior and detects intrusions by combining an LSTM with a deep auto encoder. The model effectively detects different sorts of attacks because it was trained on data which has labels that distinguishes between regular and aberrant network traffic. (Dong, 2020) suggested an IDS that used MCA (Multivariate Correlation Analysis) in conjunction with LSTM. By using IG (Information Gain) for feature selection, their MCA-LSTM model narrowed down the dataset to a pertinent subset. The Triangle Area Map (TAM) matrix created from the chosen features was then fed into the LSTM for incursion prediction. The UNSW-NB15 datasets and NSL-KDD were used to assess the model, and it achieved an 82.15% test accuracy for 5-class classification on NSL-KDD. The study did not examine the effect of dataset size or take into account a wider variety of performance criteria, such the F1 score, even though it outperformed alternative approaches.

METHODOLOGY

The proposed architecture of the model to be developed will be adopted in the following stages in this research:

Collection of data: Among the available IDS datasets, the CIC IDS 2018 dataset was selected, which includes modern DoS attacks and benign network traffic to simulate real-world scenarios. Halting it early is normally used to stop training if the validation loss stops getting better in order to avoid over fitting. It is designed based on eleven key criteria, such as attack diversity, labeling, complete capture, and comprehensive interaction.

Data exploration and visualization: Data exploration and visualization involve the examination of datasets for their structure, patterns, and anomalies. These include summarization of key statistics, missing value testing, and feature distribution analysis. Visualization applications such as scatter plots, histograms, and heat maps help make sense of complex data relationships. The data provide insights into feature selection and pre-processing operations for machine learning models.

Correlation analysis

- The matrix diagonal represents how each variable is perfectly correlated with itself, the values on the diagonal are always 1.
- Correlation coefficients are the figures in the cells. They can range between -1 and 1.
- Symmetry of the matrix is caused by the fact that the correlation between A and B is the same as B and A.

Feature engineering with PCA

To improve the model performance, locating and selecting pertinent features is important. This technique's fundamental objective is to increase model accuracy by removing unnecessary or redundant data, decreasing the dimensionality of input variables, and streamlining intricate models (Venkatachalam, 2019). Incorporating feature selection techniques within NIDS can significantly enhance their ability to detect security threats. By focusing on the most critical aspects of network traffic data, feature selection improves NIDS accuracy and reduces false positives, thereby optimizing system performance (Tharaet al. 2021).

There are several approaches to feature selection and extraction, including the following:

Filter methods: It uses statistical metrics like correlation or mutual information to understand the importance of each feature separately.

Wrapper methods: These techniques use a model trained on each subset of features and its performance is evaluated on a validation set. Based on its contribution to model performance, the best feature subset is chosen.

Embedded methods: These techniques let the model identify the most pertinent features on its own by incorporating feature selection into the model training procedure. Principal Component Analysis (PCA) is used to cut down dimensions by finding subset of features that have the potential to improve predictive performance and reduce complexity. In contrast to feature selection algorithms usually divided into wrapper and filter techniques, PCA does not select from the existing features. Instead, it transforms them into a different set of uncorrelated components, ordered by the amount of variance they retain from the original data

Sampling: After extraction of certain features from our original data is done, then following is the splitting of our data into validation, testing, and training datasets after specific features have been extracted from our raw data. The model will be trained using the training subset, and the test subset will be will be utilized to evaluate its final performance.

LSTM Model: Objective of LSTM networks is to address the vanishing gradient issue, an optimization challenge in training neural networks, ensuring effective learning of long-term dependencies. This study leverages LSTM for anomaly detection. The particular LSTM model is displayed in Figure3.1. $X = \{x_{\{1\}}, x_{\{2\}}, \dots, x_{\{n\}}\}$Is the LSTM model's input sequence, which is the intermediate feature vector of the network's output,$X = \{h_{\{1\}}, h_{\{2\}}, \dots, h_{\{T\}}\}$Is the LSTM model's vector sequence of hidden layer nodes, and $X = \{y_{\{1\}}, y_{\{2\}}, \dots, y_{\{n\}}\}$ is the matching output vector sequence. Every step of the LSTM, a couple of "gates" is used to control and calculate the transmission state. These gates remember the feature information that must be retained for a long period while forgetting irrelevant feature information. Through the $h_{\{(t-1)\}}$ state of earlier time series and input vector $x_{\{t\}}$ of the present time sequence, the LSTM model primarily determines and computes the transition between $c_{\{t\}}$ and $h_{\{t\}}$ Three calculation control 'gates' are involved in this process: 'output'$o_{\{t\}}$, 'forget',$f_{\{t\}}$, 'input', and $i_{\{t\}}$.

Dense Layer

An activation function is performed after a linear transformation by the dense layer:

An activation function is performed after a linear transformation by the dense layer:

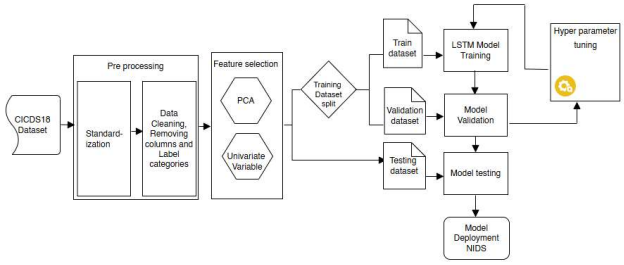

Figure 3.1 The model's architecture

$y = W + b$ for: $b \in R$: Bias term
$w \in R^{\{n\}}$: Weight vector,

This produces the final output, $y \in R$as the model's prediction

Loss Function:$L(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N} = (y_i, \hat{y}_i)^2$where: y: Ground truth, $\hat{y}_i$: Predicted values, N: Number of samples.

Optimization

Adam optimizer as an optimization strategy for gradient descent, the method demonstrates strong efficiency in handling large problems, especially those characterized by vast amounts of data or high-dimensional parameters

- The Adam optimizer minimizes the loss function by updating weights using gradients computed via back propagation
- The model is trained for some specific epochs with a batch size of B=32. Each epoch iterates over the training set to minimize the lossEpoch Loss $= \frac{1}{B}\sum_{j=1}^{B} L\left(y^j, \hat{y}^j\right)$
- Finally evaluates performance on a validation set $X_{\{test\}}, Y_{\{test\}}$ after each epoch to track generalization performance.

Activation function: For neural networks, it is essential for activation functions to be used due to the calculation of the weighted total of inputs and biases to decide whether to activate a neuron. They serve as a gate which determines if a value coming in is more than the predetermined threshold.

Sigmoid function: The sigmoid function, is commonly used in feed-forward neural networks with non-linear activation. This is a real function because of its positive gradient across its domain. However, one of its major drawbacks is the vanishing gradient problem, which can hinder effective back propagation in deeper layers (Thara, et al. 2021)

Hyperbolic tangent function (Tanh): Unlike the sigmoid function, it produces a zero-centered output within the range of -1 to 1, which can improve the training process through back propagation(LeCun,Bengio and Hinton, 2015). However, similar to the sigmoid function, Tanh also suffers from the vanishing gradient problem, which can slow down learning in deep networks.

Softmax function: (Goodfellow, Bengio, and Courville, 2016) noted that the softmax function is a standard tool in deep learning, especially for classification. By mapping a vector of real numbers into normalized values, it produces a probability distribution in which all outputs fall within the [0,1] interval and add up to 1.

Performance Evaluation Metrics: The performance of IDS is defined using a variety of categorization metrics, some of which may go by several names. A two-class classifier's confusion matrix, which is utilized in evaluating IDS, each row in this matrix contains occurrences in the real class, and each column represents instances

projected to belong to a specific class.Typically, IDS performance is assessed as follows:

True Positive Rate (TPR): The ratio of accurately predicted attacks to all actual attacks is used to compute the TPR. The TPR would be 1 in a perfect world, when every encroachment is found, but this is rarely the case in reality. The Detection Rate (DR) or Sensitivity are other names for the TPR. The TPR is represented mathematically as
False Positive Rate (FPR): The FPR measures how many normal instances are incorrectly flagged as intrusions compared to the total number of normal instances.

False Negative Rate (FNR): The False Negative Rate (FNR) refers to the proportion of anomalies or attacks that are incorrectly classified as normal by IDS.

Accuracy: The Classification Rate (CR), also known as Accuracy, measures the overall effectiveness of an IDS in correctly identifying both anomalous traffic behavior and normal ones.

Receiver Operating Characteristic (ROC) Curve: Shows visually the representation of performance by mapping the True Positive Rate (TPR) on the vertical axis against the False Positive Rate (FPR) on the horizontal axis.

Confusion matrix: Table 3.2 presents the confusion matrix, a widely used tool for evaluating classification performance. It includes four standard elements: True Positives (TP), True Negatives (TN), False Positives (FP, or Type I errors), and False Negatives (FN). According to (Dagur, A., Singh, K., Mehra, P.S. and Shukla, D.K. (eds.) (2025).), an effective classifier seeks to maximize TP and TN while minimizing FP and FN to improve reliability and accuracy.

Table 3.2 Confusion matrix

		Intrusion Observed	
		True	False
Predicted Conditions	Positive	TP	FP
	Negative	FN	TN

RESULTS AND DISCUSSION

Handling Dataset: CICIDS2018 dataset comprises network traffic data collected over eight days, with each day's data stored in a separate file. Due to hardware limitations, the experiment was conducted using data from the first three days, specifically the files corresponding to 14-02-2018, 15-02-2018, and 16-02-2018. These files were selected as they provide a representative sample of the dataset while ensuring computational feasibility.

Data merging: Since the dataset represents a time series with consistent column names and an equal number of columns across all files, a vertical merging operation was performed to merge the three chosen files into one consolidated dataset. The merged dataset was saved as "merged.csv" for further processing.

Data preprocessing: Upon initial inspection, the merged dataset contained approximately 80 columns, some of which were irrelevant to the desired objectives of the study. To streamline the dataset, only the features relevant to the study were retained, while the remaining columns were dropped. The selection of relevant features is detailed in Table. The features were retained because they are roughly within the following categories; Label: The target variable indicating whether the flow is benign or malicious. Temporal Dependencies: Features like timestamps, time of flow, and packet arrival times capture the sequential nature of network traffic, which is essential for LSTM. Flow Characteristics: Features like packet lengths, rates, and flags provide insights into the behavior of network flows. Statistical Features: These help in identifying anomalies or patterns in the data.

Feature Correlation: As seen in Figure 4.0, this figure correlation shows that, as one of the future value rises, the other value reduces, and this makes sense, for example, A large packet size may be negatively correlated to response time, because the response decreases as packets size increases. This indicates an inverse relationship between the features. However, since its correlation is between 0 and -0.05, then this is regarded as a weak negative correlation.

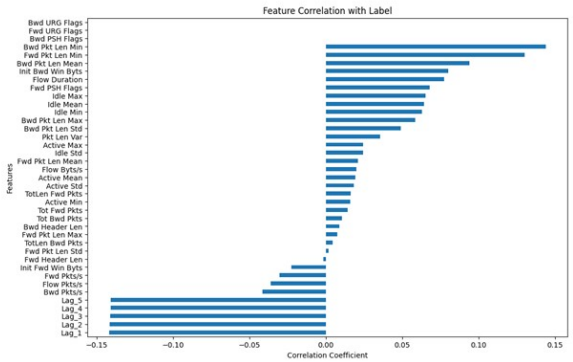

Figure 4.0. Feature correlation image

Handling imbalanced data: The labeled column of the dataset showed a notable class imbalance, with some attack kinds being over represented. The actions listed below were made in order to resolve this issue:

Attack Categorization, Removal of Minor Classes. The dataset is subsequently normalized, and the processed data is split into test and also train subsets. The proposed model incorporates an LSTM network, with a dropout layer employed to enhance robustness and computational efficiency.The model uses the SoftMax activation function, which generates probability distributions for each class, giving the highest probability to the targeted class. The total of the probabilities is approximately equal to 1, and the output values are always in the range of 0 and 1. For optimization, the "Adam" optimizer is utilized, which adjusts neural network parameters like weights and losses using dynamically modified rate learning and weight updates during back propagation. Adam, short for Adaptive Moment Estimation, is particularly effective for handling gradients with noisy and high curvature. It accelerates the learning process by computing decaying average exponentially both for the past and current gradients. The neural network's training procedure is guided by the "Categorical-Cross-Entropy" loss functions. When target values fall within the range {0, 1, 2...n}, categorical-cross-entropy is used. For every class, it shows the difference of actual and expected probability distributions. With an ideal cross-entropy value of zero, the loss diminishes as the model gets better.

The Model: The deep learning model employed in this study consists of two layers, which are fundamental components in deeply connected neural networks. The validation process was conducted by training the model within epoch's interval. The size of the dataset and the limitations imposed by constrained computer resources played a role in this decision. The model may learn from the data iteratively since each epoch represents a full forward pass through the entire dataset. The batch size, which is essential to learning process optimization, is the quantity of training samples processed in a single forward pass.

The Experiment: The experiment was carried out using Python programming language and its libraries which is not limited to Matplotlib, NumPy, Keras, Scikit-learn, Pandas, and others. The model was constructed with an LSTM architecture, incorporating dropout and dense output layers. Training was performed with a batch size of 32 in 10 epochs due to the large volume of the data size. The SoftMax activation function and Adam optimizer were utilized, while the loss was computed using the Mean Squared Error (MSE). To evaluate this IDS model, a two-stage implementation approach was employed.

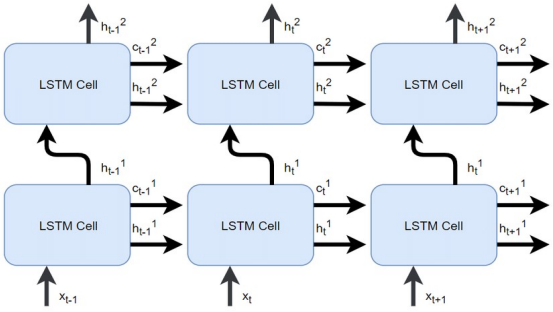

Figure 4.1 The model representation

First, the model was trained and tested without hyperparameter tuning to establish a baseline performance. In the second stage, the model was enhanced through the integration of PCA for its feature's selection, followed by hyperparameter tuning to optimize the performance of the proposed learning classifier. This comparative evaluation provided insights into the impact of parameter optimization, selection of features on the detection accuracy of the model.

Table 4.2. Classification report of non-hyper parameter tuning

	PRECISION	RECALL	F1-SCORE
Benign	0.97	0.98	0.97
Malign	0.99	0.98	0.98
Macro Average	0.98	0.98	0.98
Accuracy			0.98

The model demonstrated strong classification performance across both classes. For Benign traffic, it achieved 97% precision, 98% recall, and 97% F1-score. For Malign traffic, it recorded 99% precision and 98% for both recall and F1-score, indicating effective detection of normal and intrusive activity.

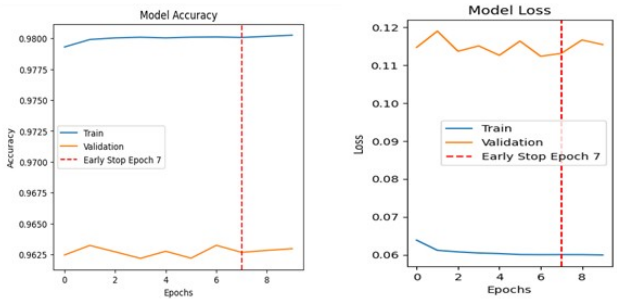

Figure 4.2 and 4.3 showing model accuracy and model loss

According to this graph, the training accuracy remained consistently high, starting just below 0.98. In contrast, validation accuracy fluctuated, highlighting variability in generalization. Early stopping at epoch 7 prevented over fitting, indicating the need for careful hyper parameter tuning to improve model stability.

Hyper parameter tuning: Hyper parameter values play a crucial role in fine-tuning an LSTM model for optimal performance. Dropout is a regularization technique that prevents over fitting by randomly deactivating a fraction of neurons during training. The number of dense layers specifies the number of fully connected layers, which perform feature extraction and classification. The number of LSTM layers controls the model's ability to learn sequential relationships, with deeper models learning more complex patterns. LSTM units control the number of memory cells per layer, influencing the model's ability to retain information over time.

Table 4.3 tuned model

	PRECISION	RECALL	F1-SCORE
Benign	0.97	0.98	0.97
Malign	0.99	0.98	0.98
Macro Average	0.98	0.98	0.98
Accuracy			0.98

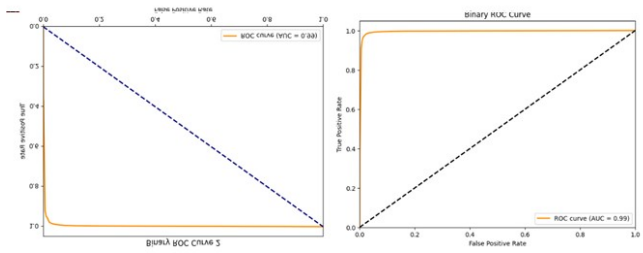

Figure 4.4 ROC Curve of the non-tuned model and Figure 4.5 ROC Curve of the tuned model

Dense dropout regularizes fully connected layers using dropout, ensuring better generalization by dissuading reliance on single neurons. The evaluation metrics presented in the tabular results table 4.3 indicate that for the benign class, the precision rate achieved was 97%, the recall was 98%, and the F1-score was 97%. Similarly, for the malign class, the precision rate stood at 99%, the recall was 98%, and the F1-score was 98%.

Comparative analysis of the tuned and non-tuned model: The plots in 4.4 and 4.5 is a non-tuned and tuned curve plot of the model, which plots the TPR (True Positive Rate) against the FPR (False Positive Rate) at various classification thresholds. The curve allows for the visualization of the sensitivity (recall) against specificity trade-off, with the best classifier having a curve that reaches the top- left corner. The AUC-ROC (Area under the ROC Curve) is a key metric that evaluates a model's ability to distinguish between two classes. A higher AUC-ROC value signifies better classification performance, with a value close to 1 indicating that the model makes highly accurate predictions. As observed in both graphs, the AUC-ROC value is nearly 1, which suggests that the model exhibits good discriminatory performance between the classes. Furthermore, the model's effectiveness is backed by the precision, F1-score, and recall metrics, which also demonstrate consistently high values. While this study primarily focuses on examining the use of LSTM networks in enhancing intrusion detection systems (IDS) in a 5G network environment, it also pertinent to explore the performance of other datasets for additional comparative analysis. For instance, the study by Venkatachalam, 2019), which utilized the UNSW-NB15 dataset, reported a model performance of 96% accuracy, 97% precision, 97% F1-score, 96% recall, and a ROC-AUC score of 0.98.Similarly, Magán-Carrión, et al. (2020) trained their model using techniques such as linear regression, random forest, and various support vector machine variants, evaluating performance based on precision, recall, F1-score, ROC-AUC, and weighted average. Their model achieved an average accuracy of approximately 95%. Comparatively, the findings in this research demonstrate a slight performance improvement over existing approaches, further validating the effectiveness of the proposed model.

CONCLUSION

In this research work, the possibility of leveraging deep learning techniques for intrusion detection within a fifth-generation (5G) network environment was explored, specifically focusing on LSTM networks. The research aimed to analyze network parameters and behavioral patterns using the widely recognized CICIDS2018 dataset, which contains diverse cyber-attack scenarios. To optimize the detection process, PCA was employed as the ideal feature selection method, ensuring that only the most relevant attributes were considered for classification. Following feature selection, an LSTM-based deep learning model was implemented to evaluate its effectiveness in identifying network intrusions. The model's performance was assessed with the help of standardized evaluation metrics, amongst them is accuracy, precision, recall, and F1-score. Additionally, hyper parameter tuning was applied to fine-tune the process in which the model learns, enhancing its ability to detect and classify anomalies with greater accuracy. This approach further

emphasizes the potential of LSTM networks in strengthening intrusion detection mechanisms within advanced network infrastructures, contributing to improved cyber security resilience in 5G environments.

REFERENCES

Alsulaiman, S. and Alzaidi, H.M. (2018). Anomaly-based network intrusion detection system using deep learning techniques. *International Journal of Advanced Computer Science and Applications*, 9(9).

Creech, G. and Hu, J. (2014). A semantic approach to host-based intrusion detection systems using contiguous and discontinuous system call patterns. *IEEE Transactions on Computers*, 63(4), pp.807–819.

Dagur, A., Singh, K., Mehra, P.S. and Shukla, D.K. (eds.) (2025). *Intelligent Computing and Communication Techniques: Proceedings of the International Conference on Intelligent Computing and Communication Techniques (ICICCT 2024), New Delhi, India, 28–29 June 2024, Volume 2*. 1st ed. Boca Raton, FL, USA: CRC Press. https://doi.org/10.1201/9781003530190.

Dong, R.-H. (2020). Network intrusion detection model based on multivariate correlation analysis–long short-time memory network. *IET Information Security*, 14(2), pp.166–174.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*. Vol. 1, No. 2. Cambridge, MA, USA: MIT Press.

Hsu, C.-M. (2019). Using long short-term memory-based convolutional neural networks for network intrusion detection. In: *Wireless Internet: 11th EAI International Conference (WiCON 2018)*, Taipei, Taiwan, 15–16 October 2018. Cham, Switzerland: Springer.

Jiang, K., Wang, W., Wang, A. et al. (2020). Network intrusion detection combined hybrid sampling with deep hierarchical network. *IEEE Access*, 8, pp.32464–32476.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *Nature*, 521(7553), pp.436–444.

Liu, Z., Gu, G. and Wang, J. (2021). A hybrid intrusion detection system based on scalable K-means + random forest and deep learning. *IEEE Access*, 9, pp.75729–75740.

Magán-Carrión, R., Urda, D., Diaz-Cano, I. et al. (2020). Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. *Applied Sciences*, 10(6), p.1775.

Sharma, S., Garg, S., Laxmi, V. et al. (2018). LSTM-based botnet detection model for Internet of Things. *International Journal of Distributed Sensor Networks*, 14(11).

Thara, D.K., Premasudha, B.G., Nayak, R.S. et al. (2021). Electroencephalogram for epileptic seizure detection using stacked bidirectional LSTM_GAP neural network. *Evolutionary Intelligence*, 14, pp.823–833.

Venkatachalam, J.P. (2019). UNSW-NB15 dataset feature selection and network intrusion detection using deep learning. *International Journal of Recent Technology and Engineering*, 7(5), pp.443–446.

****