

Adversarial-Aware Deep Learning Frameworks for Enhanced Network Threat Detection

Sridevi Allu¹, Sivaraju Kuraku², Naveen Shrivastava³, Mahesh Marri⁴,
Khaja Moiezuiddin Mohammed⁵, Rashmika Jalagam⁶

¹ American Airlines

² University of the Cumberland

³ T-Mobile

⁴ Amazon Web Services, Senior Electrical Engineer

⁵ IQVIA

⁶ Matilda Cloud

Abstract

Network Intrusion Detection Systems (NIDS) are used as the first line of defense against cyber threats since they protect vital data and network resources. Nonetheless, when Deep Learning (DL) is applied to NIDS, they are very susceptible to adversarial attacks, which interfere with their reliability. The paper describes a more improved network threat detection model with the CICIDS 2017 dataset, which includes both benign and malicious communication from different protocols. To make the dataset quality and robust, data cleansing, normalization, feature selection, and balancing of classes were performed using SMOTE. An LSTM model was suggested as it has the ability to exploit sequential and temporal relationships in network traffic. The performance of the LSTM was compared to that of the Support Vector Machine (SVM), the Random Forest (RF), and the Deep Neural Network (DNN). 96.5% F1-score, 97% accuracy, 97% precision, and 96% recall were achieved by the LSTM, outperforming the other models, according to the experiment's results. It has been proven through these results that LSTM is an excellent method for the detection of assault patterns that are in flux, which have better detection rates and lower false alarms, rendering it a potentially useful tool in improving network security.

Keywords: Computer Security, Image Forensics, Machine Learning, Adversarial Attacks, Network Intrusion Detection Systems (NIDS), Cybersecurity

DOI: 10.21590/sdfg152s

I. INTRODUCTION

The blistering development of Cyberspace and digital technology has already ingrained itself into contemporary civilization, which not only supports the everyday life of the people, but also plays a considerable role in the economy of the state, politics and social stability [1]. Cyberspace is also facing unprecedented attacks on its security, along with the popularization of the internet. Threats including malware, advanced persistent threats (APTs), phishing, botnets, denial of service attacks (DoS/DDoS), insider threats, etc., [2]. In addition to causing financial losses, these risks may also result in the disclosure of private information and invasions of privacy and have implications on the national security and social order [3].

A significant issue of concern to modern network security is the extent and complexity of cyberattacks. As organizations are increasingly relying on interconnected systems, including IoT technologies, the role of effective implementation of cybersecurity has become more timely. NIDS are some of the instruments that are used in securing such networks to detect and prevent any malicious acts in the network [4][5]. A major characteristic of adversarial attacks on NIDS is manipulation of features of networks that are not fixed such as packet sizes, network times, and flow patterns. In most cases, attackers may alter these features in order to evade detection [6]. They are adversarial attacks that have led to a significant concern since they focus on the vulnerability of the model's decision bounds to help attackers evade detection [7][8].

The use of machine learning (ML) and big data analytics into cyber models is a crucial advancement in real-time threat identification and response [9]. The ML algorithms can handle great quantities of information at a very rapid rate and establish patterns and anomalies that can be utilized to indicate potential cyber threats. With the help of such technologies and big data analytics, it is possible to monitor continuously and even realize real-time analysis of network traffic, user behaviour, a record of system states, and even define threats and remove them before they can do significant harm [10][11][12]. The deep learning (DL) architecture can offer a good solution to reducing false positives and improving threat detection, which transforms the cybersecurity system through an automated threat detector and live response system [13]. Thus, the great future of DL is high-technology intrusion detection systems; however, the fact that these systems can be easily compromised by the opponent is a severe problem that must be addressed to enhance their strength and security [14]. With these new technologies, companies that can transcend the limitations of the conventional cybersecurity solutions and develop more resilient protection against the ever-evolving threat landscape [15][16].

A. Motivation and Contribution

The rationale for this research is due to the increasing complexity and frequency of cyberattacks, which pose serious threats to the availability, confidentiality, and integrity of electronic systems. The antiqueness of intrusion detection types tends to fail to address the complexity and dynamism of attack patterns, producing high alarms and reducing the accuracy. Due to the existence of attempts to generate detailed datasets, such as CICIDS 2017 that simulates realistic network traffic and diverse attack scenarios, there is a great likelihood of developing and experimenting with complex detection models. An alternative promising direction towards enhancing detection accuracy and resilience can be the use of the DL methods such as networks with Long Short-Term Memory (LSTM) that can learn sequential and temporal dynamics in data. The reasons supporting this study are that there is need to come up with an intelligent and reliable IDS, which can effectively track known and new threats and improve the network security and resistance to cyber-attacks. This research offers several key contributions as listed below:

- Utilized an artificial network traffic environment, which is a realistic network traffic scenario with benign and various types of modern attacks, CICIDS 2017.
- The systematic cleaning of data applied in practice by managing missing values, reducing duplication and noise to have good quality and reliable data.
- Normalization and selection of features done to normalize the feature scales and eliminate duplication and only the model was trained using the most important characteristics.
- Conducted a comparative analysis of the performance of LSTM with traditional algorithms in the field of ML and DL in order to highlight its applicability.
- Assessed the model performance holistically through different measures of evaluation, including ROC analysis, recall, F1-score, accuracy, and precision.

B. Novelty and Justification of the Study

This study is novel in the sense that it incorporates a LSTM network to detect network threats using the CICIDS 2017 dataset with the aid of a full reprocessing pipeline, including noise reduction, normalization, feature selection, and class balancing with SMOTE. The proposed LSTM model is the first to use temporal and sequential dependencies in network traffic, unlike the traditional ML model or the generic DL model, which allows it to more efficiently capture changing attack behavior. The increasing

need for intrusion detection systems (IDS) that can handle real-world threats serves as the rationale for this investigation, imbalanced, and high-dimensional datasets while maintaining high reliability and reducing false alarms. This approach provides a robust and scalable framework for enhancing network security against increasingly sophisticated cyber threats.

C. Organization of the Paper

The structure of this paper is as follows: Section II provides a review of related work on network threat detection, Section III describes the dataset, preprocessing steps, and model implementation, Section IV discusses the experimental results and comparative analysis, and Section V concludes the study with key findings and future research directions.

II. LITERATURE REVIEW

Several significant research studies on enhanced network development of this work have been guided and strengthened by the evaluation and analysis of threat detection.

Jahromi et al. (2020) The proposed method might be applied to real-time malware threat hunting, particularly for safety-critical systems where a poor model convergence could have catastrophic consequences, such as electronic health or the Internet of Battlefield/Military of Things. On datasets of ransomware, Windows malware, Android malware, and they use both static and dynamic analysis to assess the efficacy of the suggested approach in assessing Internet of Things (IoT) malware. They also provide a comparative overview of the performance of the suggested approach versus the conventional stacking LSTM method for the detection of IoT malware on an IoT-specific dataset. More precisely, the suggested approach outperforms conventional LSTM-based approaches in these crucial areas, finding IoT malware samples with an accuracy of 99.1%, with an MCC of 0.95 and an AUC of 0.985 [17].

Bharati and Tamane (2020) developed IDS focused upon finding of anomalies rather than detection of misuse. Additionally, ML may potentially solve threshold issues in anomaly detection. Similar to dangerous code datasets, network intrusion detection data sets are rather scarce. The most often used dataset for IDS evaluation is still KDDCUP-99%. A large number of tests on KDD or improved KDD models for ML-based IDS. This analysis uses the CSE-CICIDS-2018 dataset, which includes the most up-to-date fundamental system vulnerabilities. They use Random Forest, an ML-based intrusion detection system, for CSE-CIC-IDS-2018, which has an outstanding accuracy score of 99% [18].

Y. Li et al. (2019) designed a DL model to categories a lot of DGA data. By managing the massive dataset, they progressively gathered, to enhance the proposed ML framework, they developed a DNN model. Numerous Results from experiments show that the DNN model and the suggested framework are accurate. Specifically, they attain 97.79% classification accuracy in the DNN and 95.89% classification accuracy in the framework [19].

Singh, Mehtre and Sangeetha (2019) sought to identify insider risks by monitoring and analyzing user behavior activity sequences using a user behavior profiling technique. They discover that basic single-state LSTM is inferior to multistate LSTM. Using a publicly accessible insider threat dataset for training, Insider threats may be successfully identified using the proposed method using Multistate LSTM, which produces an AUC of 0.9047 on test data and 0.9042 on train data [20].

Kumar, Viinikainen and Hamalainen (2018) suggest and assess a model based on networks that employs ensemble ML techniques to assess malware communication network flows and detect mobile threats. In addition to preventing overfitting of the model, ensemble ML techniques also address the problems caused by the attackers' evolving behavior. Because Android-based mobile malware is so popular with consumers, it is the subject of investigation. They combined the results of five supervised ML algorithms—RF, PART, JRIP, J48, and Ridor—using ensemble techniques. According to the evaluation findings, the suggested model has a 98.2% accuracy rate in identifying known and unknown threats [21].

Table I summarizes recent studies on network threat detection, highlighting the proposed models, datasets used, key findings, and challenges faced

TABLE I. A COMPARATIVE ANALYSIS OF RECENT STUDIES ON PREDICTIVE MODELING OF NETWORK THREAT DETECTION

Author	Proposed Work	Results	Key Findings	Limitations & Future Work
Jahromi et al. (2020)	Enhanced LSTM-based real-time malware threat hunting for vital systems (medical, military, and IoT)	Accuracy: 99.1%, AUC: 0.985, MCC: 0.95	In IoT malware detection utilizing static and dynamic analysis, it outperformed regular LSTM	Needs testing on larger and more diverse real-world datasets for generalization
Bharati & Tamane (2020)	IDS focused on anomaly detection using ML (Random Forest) with CSE-CIC-IDS-2018 dataset	Accuracy: 99%	Demonstrated improved anomaly detection compared to traditional misuse detection	Limited dataset diversity; future work includes extending to multi-dataset environments
Y. Li et al. (2019)	DGA domain classification using two-level model, HMM, clustering, and DNN	Framework: 95.89%, DNN: 97.79%	DNN enhanced classification of large-scale DGA domains	Scalability and adaptability to real-world DNS traffic need improvement
Singh, Mehtre & Sangeetha (2019)	Insider threat detection using Multi-State LSTM + CNN ensemble	AUC: ~0.90	Multi-State LSTM outperformed basic LSTM in detecting insider threats	Needs testing on larger insider datasets to validate scalability
He et al. (2019)	Zero-day attack detection in power-grid controllers using RED + temporal deep learning	Accuracy > 99.9%, near-zero FPs, <360ms latency	Real-time detection with very low latency and high reliability	Evaluation limited to power-grid domain; broader applicability needs exploration
Kumar, Viinikainen & Hamalainen (2018)	Mobile threat detection using ensemble ML (RF, PART, JRIP, J48, Ridor)	Accuracy: 98.2%	An ensemble approach improved generalization and detection of Android malware	Requires extension to non-Android platforms and larger malware datasets

III. RESEARCH METHODOLOGY

The methodology of this study involves utilizing the CICIDS 2017 dataset, which contains both benign and malicious network traffic. Pre-processing steps, including data handling of missing values, noise, normalization using the min-max technique, and feature selection to retain only the most relevant attributes. Following data balancing, the dataset was subsequently separated into two groups: a training group consisting of 80% of the data and a testing group consisting of 20% of the data. After that, the suggested LSTM model was created, making use of its capacity to identify temporal relationships in sequential data. The trained model's efficacy in identifying intrusions was evaluated in detail using ROC-

AUC, F1-score, accuracy, precision, and recall. Figure 1 shows the proposed flowchart for Enhanced Network Threat Detection .

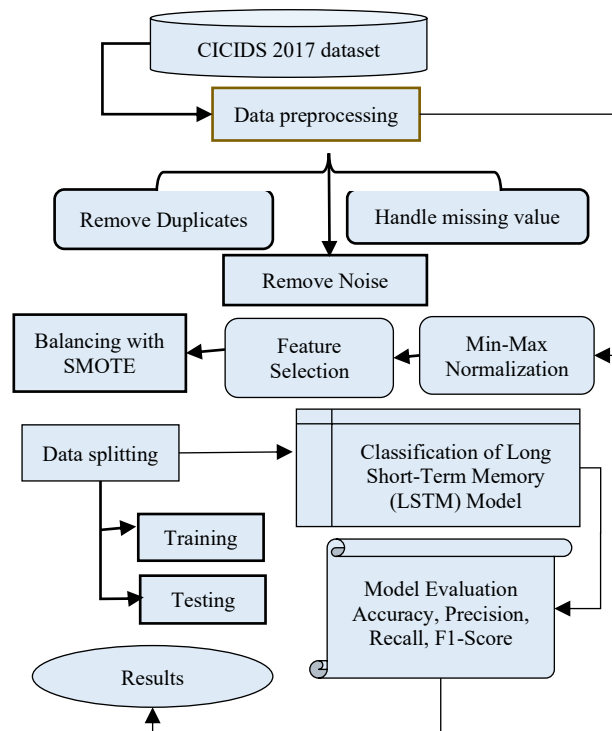


Fig. 1. Proposed Flowchart for Enhanced Network Threat Detection

The following section elaborates on each step outlined in the proposed Enhanced Network Threat Detection flowchart.

A. Data Gathering and Analysis

This study utilizes the CICIDS 2017 dataset, the FTP, SSH, HTTP, HTTPS, and email protocols, all states; this innocuous traffic includes 25 individuals. Five days' worth of network traffic was gathered, dumped with traffic from regular activities on one day, and attacked on the other days. The several assaults that were injected were Heartbleed, Web Attack, Infiltration, Botnet, Brute Force FTP, Brute Force SSH, "DoS," and "DDoS." Data visualizations such as pie chart and heatmaps were used to examine attack distribution, feature correlations etc., are given below:

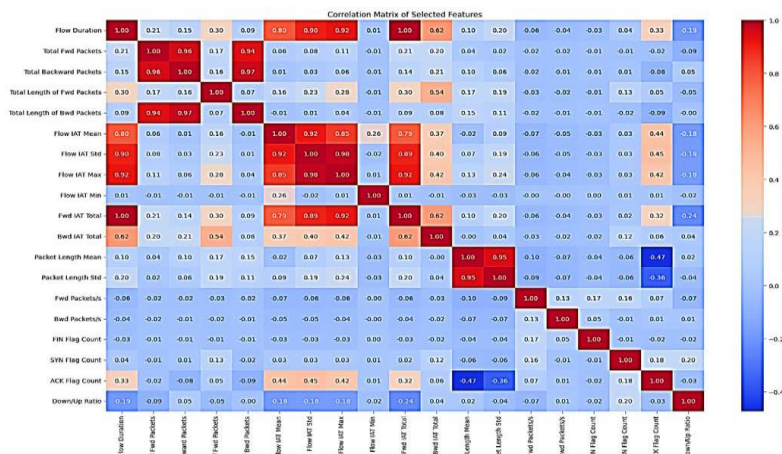


Fig. 2. Feature Correlation Heatmap

Figure. 2. The CICIDS 2017 dataset's feature associations are displayed in the correlation heatmap, with values ranging from -1 to $+1$. Strong positive correlations appear among traffic volume-associated characteristics that show redundancy, such include the total number of forward packets, flow duration, and flow bytes. Flow IAT Mean and Flow IAT Std also show high correlation, reflecting inter-arrival time dependency. Features like FWD Header Length and flag counts demonstrate weak or near-zero correlations, suggesting independent discriminatory power. The heatmap aids in identifying multicollinearity, guiding feature selection for improved model performance in network threat detection.

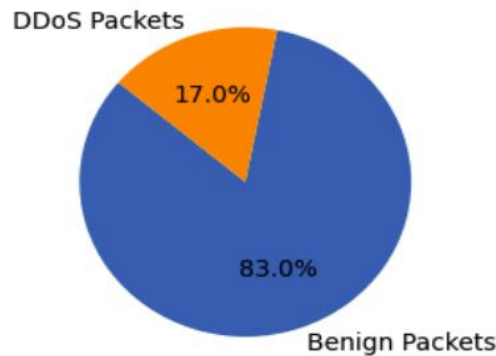


Fig. 3. Pie chart for CICIDS 2017 Dataset

Figure. 3 is a pie chart illustrating the composition of the CICIDS 2017 dataset based on network packet types. It shows that the majority of the dataset, specifically 83.0%, consists of Benign Packets, while the remaining 17.0% are DDoS Packets (Distributed Denial of Service).

B. Data Pre-processing

The dataset was pre-processed by handling missing values, removing noise, and then undergoing data transformation and normalization. The key reprocessing steps are outlined as follows:

- **Handle missing and null values:** In data preparation, handling missing and null values is a crucial step that might affect the models' precision and accuracy. The dataset for the current investigation is cleansed of entries with null or missing values. The performance of the model is improved, and computational complexity is decreased by deleting such entries.
- **Remove Noise:** Eliminating data noise reprocessing is necessary to improve the effectiveness of further studies or models and the calibre of the data. The kind of data and the type of noise determine which strategies are used. The kind of data and the tactics employed depend on the type of noise.

C. Data Normalization

The data was normalized using the min-max technique, which restricted the values to those between 0 and 1. To minimize the impact of outliers and maximize the effectiveness of the classifiers that were employed, this was done. Normalization was carried out using the mathematical method Equation (1):

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where X stands for the feature's initial value, X' for its normalised value, X_{min} for its minimum value, and X_{max} for its maximum value.

D. Feature Selection

The degree of dependence between two random variables can be measured using the linear correlation coefficient, which is one example of a linear measure of dependence. If correlations between network

traffic records are thought of as linear connections, then this measure is applicable. But given the communication that occurs in the actual world, additionally, there may be nonlinear correlations between the variables. It appears that two nonlinearly dependent variables cannot be related using a linear measure [22][23]. Therefore, assess the ability to analyze the relationship between two variables regardless of whether they are reliant on their other linearly or nonlinearly. For these reasons, this study aims to explore the process of selecting the most desirable attributes from a feature space, irrespective of the kind of connection that exists between them.

E. Data Splitting

To evaluate efficiency, the dataset was separated into subgroups for testing and training. The model parameters were estimated and trained using the first 80% of the data, with 20% more is reserved for testing and performance review.

F. Proposed Long Short-Term Memory Model

Text classification is the specialty of LSTM since it can identify long-term relationships between texts. The LSTM classifier is an RNN, which is a kind of multilayer network that uses the outputs of the previous layer as inputs for the subsequent layer [24]. LSTM's feedback links allow it to handle data sequences instead of just individual data points. An LSTM node is a component that includes a cell, forget gate, input gate, and output gate. The cell is responsible for storing data across time, while the cell's information flow is regulated by the three gates. The memory blocks that make up the LSTM layers are connected recurrently, and each memory block has three multiplicative gates. For a certain amount of time, gates carry out a continuous kind of write, read, and reset operation to guarantee that the temporary data is used. Equations (2) through (7) update the unit's input, x_t , h_{t-1} , c_{t-1} , and output, h_t , c_t :

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad \square \square \square$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (3)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (4)$$

Input transform

$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \quad (5)$$

State update

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (6)$$

$$h_t = o_t \odot \tanh(c_t) \quad (7)$$

The logistic sigmoid function with multiplication by elements is shown by the symbols σ and $\odot$ in the equations above, respectively. A memory cell ct , an input gate i_t , A forget gate f_t , an output gate o_t , and a hidden unit htt at each time step t make up the LSTM unit. The learnt parameters are W and U , and the additional bias is indicated by b . It makes sense that the input gate regulates the amount of updating of each unit, the forget gate regulates the amount of erasing of Memory cells and the output gate controls the amount of internal memory state that is displayed.

G. Evaluation Metrics

A range of performance metrics was used to evaluate the effectiveness of the proposed design. Comparing the actual values with the expected outcomes of the trained models allowed for the calculation of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Following that, crucial F1-score, recall, accuracy, and precision were among the measures that were calculated, as explained below:

1) Accuracy

The ratio of the quantity of examples (input samples) in the dataset that the trained model accurately predicted to the total number of occurrences is Equation (8):

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad \square \square \square$$

2) Precision

The ratio of properly predicted positive cases to all projected positive occurrences is a measure of the model's accuracy. Precision indicates how good the classifier is in predicting the positive classes and is expressed as Equation (9):

$$Precision = \frac{TP}{TP+FP} \quad \square\square\square$$

3) Recall

This measure is the proportion of correctly anticipated positive events to all instances that ought to have turned out positively. It may be expressed mathematically as Equation (10):

$$Recall = \frac{TP}{TP+FN} \quad \square\square\square\square$$

4) F1 score

It combines the accuracy and memory harmonic mean, that is, it helps to balance recall and precision. Its range is [0, 1], mathematically, it is given as Equation (11):

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \square\square\square\square$$

A true negative (TN) or true positive (TP) is what is counted as itself. In the event that the example cannot be decided as itself, such as when a positive prediction turns out to be negative or a negative prediction turns out to be positive, a fake positive (Fake Positive) or false negative (Fake Negative) is recorded.

5) Receiver Operating Characteristic Curve (ROC)

The percentage of cases that are successfully labelled as positive vs the percentage that are wrongly classed as positive for numerous decision cut-off points is plotted visually by the ROC. TPR is also known as recall or sensitivity, whereas FPR is equivalent to 1-specificity.

IV. RESULTS AND DISCUSSION

In this section, the experimental setup is described, and the training and testing procedures show how well the model performs. The application of the foundation of the suggested model was the use of GPU-enabled TensorFlow (v2.x) as a back-end and Keras (v2.x) as a high-level API. The experimental setup was constructed using an NVIDIA Tesla V100 to enable accurate and efficient predictions and was equipped with an Intel(R) Xeon(R) Gold 6130 processor, 64GB of RAM, and 2 TB of storage, and was connected to a 1 Gbps Ethernet network. The suggested model was trained using CICIDS 2017 and premised on the key metrics of performance, including F1-score, recall, accuracy, and precision, as shown in Table II. The prediction results of the proposed LSTM framework for enhancing network threat detection. A high score of 97.5% assisted the model in achieving impressive accuracy, which was supported by a high recall of 96% and an accuracy of 97%, demonstrating that the model could accurately identify threats with few false positives and false negatives. Additionally, the model's ROC-AUC of 0.99 indicates its capacity to distinguish between benign and malignant network activity, and its F1-score of 96.5% indicates a reasonable trade-off between recall and accuracy.

TABLE II. PREDICTION RESULTS OF LSTM MODEL FOR ENHANCED NETWORK THREAT DETECTION USING CICIDS 2017 DATASET

Performance matrix	LSTM
Accuracy	97.5
Precision	97
Recall	96
F1-score	96.5
ROC-AUC	0.99

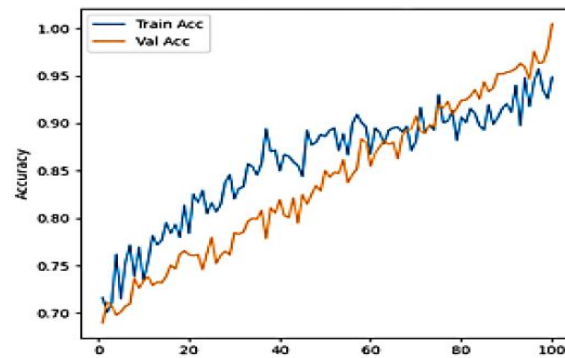


Fig. 4. Accuracy curves for the LSTM Model

Figure 4 shows the suggested model's training and validation accuracy curves across 100 epochs. Both curves have a consistent upward movement which shows the constant improvement in performance as the training goes on. The training accuracy begins at approximately 0.70 and increases towards approximately 1.0 whereas the validation accuracy follows the same pattern and therefore begins at a slightly lower level but reaches above 0.95. The near coincidence of the two curves especially during the later periods, is an indication of good generalization ability that does not overfit. Generally, the findings demonstrate. The model's capacity to produce training and validation data with high accuracy.

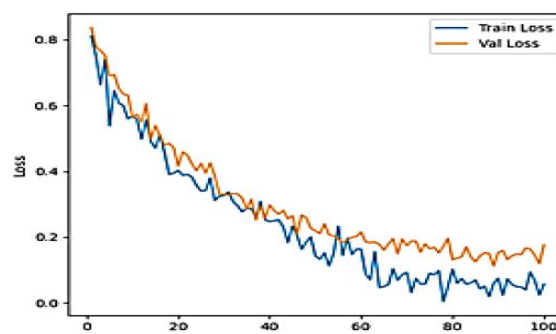


Fig. 5. Loss curve for the LSTM Model

The training and validation loss patterns for the proposed model after 100 epochs are shown in Figure 5. The curves both exhibit steady downward trend which is a sign of good learning in the training process. The model has effectively reduced the training set mistakes, as seen by the training loss decreasing more quickly and leveling out at a lower value than the validation loss. The validation loss also reduces significantly and levels off at about 0.2 which is an indication of good generalization capability with relatively small variations. By and large, it can be seen that the plot is well-trained with no indication of serious overfitting since it shows a high performance in training as well as validation dataset.

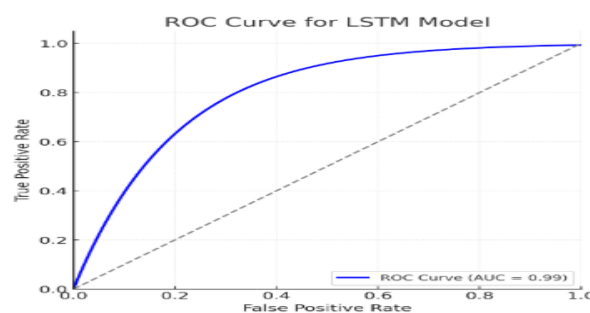


Fig. 6. ROC Curve for LSTM Model

Figure 6 shows the LSTM model's ROC curve demonstrates the model's exceptional ability to distinguish between malicious and benign network activity. At different thresholds, there are relatively few false positives and a high percentage of true positives, as seen by the curve's sharp ascent in the upper-left corner. The model has achieved a wonderful ROC-AUC of 0.99 indicating its near-perfect classification and great generalization capability. It proves that the suggested LSTM model is very trustworthy and efficient in detecting network threats.

A. Comparative Analysis

To verify the suggested LSTM model's effectiveness, a comparative accuracy analysis against other existing models is provided in Table III. In this comparison, the CICIDS 2017 dataset is used to do a comparative study of various models to improve network threat detection. The Random Forest (RF) model produced an F1-score of 90.5%, a high recall of 96.9%, and an accuracy of 94%, with intermediate precision of 84.9% and 96.9%. Support Vector Machine (SVM) model exhibited more accuracy of 96.8% and high precision (92.5%) but recall was not indicated and had an F1-score of 91.2%. Although the Deep Neural Network's (DNN) F1-score was not disclosed, it was discovered to get a 96.4% accuracy rate with balanced precision (96.2%) and recall (96.5%). The Long Short-Term Memory (LSTM) model has the highest accuracy (97.5%), high precision (97%), recall (96%), and F1-score (96.5%), all of which illustrate its greater efficacy in identifying network threats.

TABLE III. PERFORMANCE COMPARISON OF DIFFERENT MODELS FOR ENHANCED NETWORK THREAT DETECTION

Model	Accuracy	Precision	Recall	F1-score
RF [25]	94	84.9	96.9	90.5
SVM[26]	96.8	92.5	-	91.2
DNN[25]	96.4	96.2	96.5	-
LSTM	97.5	97	96	96.5

The greatest benefit of the LSTM model suggested is that it possesses the greatest accuracy percentage of 97.5% when compared to alternative network threat identification methods. The fact that it can find sequential patterns in data allows it to very well detect complex attack patterns, and in addition to having a fair balance between F1-score, recall, and accuracy. A high level of performance is an indicator that it is powerful, stable and capable of operating in real time to detect threats within a network, unlike the traditional models of ML.

V. CONCLUSION AND FUTURE STUDY

The modern network security requires the use of advanced Network Intrusion Detection Systems (NIDS) since attacks become more sophisticated. The traditional signature-based NIDS are not able to deal with zero-day and dynamic attacks. In contrast, NIDS based on the use of MLs have become a feasible solution, however, they are susceptible to adversarial evasion attacks, which manipulate network traffic with a subtle approach, aiming to evade detection. The experimental work's findings allow us to conclude that the provided LSTM model is more effective than the rest of the classifiers that were employed in the experiment based on the network threat detection on the CICIDS 2017 dataset. It was found that the LSTM model was the most accurate and its accuracy was 97.5%, showing that this model is more ready to find the temporal correlations and sequence patterns in network traffic. This shows that the LSTM is more efficient than the other ML and DL methods in intrusion detection applications because it is more accurate, has fewer false alarms, and has strong detection capabilities. It will seek to enhance the prior threat prediction using the multiple DL approach in order to ascertain the long-term pattern of past data to tackle the challenging issue of cyberattacks in the future.

REFERENCES

1. S. Naseer et al., “Enhanced Network Anomaly Detection Based on Deep Neural Networks,” *IEEE Access*, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2863036.
2. H. P. Kapadia, “Cross-Platform UI/UX Adaptions Engine for Hybrid Mobile Apps,” *Int. J. Nov. Res. Dev.*, vol. 5, no. 9, pp. 30–37, 2020.
3. Q. Li, W. Li, J. Wang, and M. Cheng, “A SQL Injection Detection Method Based on Adaptive Deep Forest,” *IEEE Access*, vol. 7, pp. 145385–145394, 2019, doi: 10.1109/ACCESS.2019.2944951.
4. B. Biggio et al., “Evasion attacks against machine learning at test time,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013. doi: 10.1007/978-3-642-40994-3_25.
5. N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in *2016 IEEE European symposium on security and privacy (EuroS&P)*, 2016, pp. 372–387.
6. D. D. Rao, “Multimedia-Based Intelligent Content Networking for Future Internet,” in *2009 Third UKSim European Symposium on Computer Modeling and Simulation*, 2009, pp. 55–59. doi: 10.1109/EMS.2009.108.
7. A. Bridges, T. R. Glass-Vanderlan, M. D. Iannacone, M. S. Vincent, and Q. (Guenevere) Chen, “A Survey of Intrusion Detection Systems Leveraging Host Data,” *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–35, Nov. 2020, doi: 10.1145/3344382.
8. M. Tavallaei, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, 2009. doi: 10.1109/CISDA.2009.5356528.
9. S. S. S. Neeli, “Decentralized Databases Leveraging Blockchain Technology,” vol. 8, no. 1, pp. 1–8, 2020.
10. L. Belli and D. C. M. Doneda, “Municipal Data Governance: an analysis of brazilian and european practices / Governança de Dados Municipal: uma análise das práticas brasileiras e europeias,” *Rev. Direito da Cid.*, vol. 12, no. 3, pp. 40–63, Dec. 2020, doi: 10.12957/rdc.2020.44310.
11. V. Benson and T. Buchanan, “Social big data and its integrity: The effect of trust and personality traits on organic reach of Facebook content,” in *Cyber Influence and Cognitive Threats*, 2019. doi: 10.1016/B978-0-12-819204-7.00008-7.
12. A. Thapliyal, P. S. Bhagavathi, T. Arunan, and D. D. Rao, “Realizing Zones Using UPnP,” in *2009 6th IEEE Consumer Communications and Networking Conference*, 2009, pp. 1–5. doi: 10.1109/CCNC.2009.4784867.
13. A. Aldweesh, A. Derhab, and A. Z. Emam, “Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues,” *Knowledge-Based Syst.*, vol. 189, p. 105124, Feb. 2020, doi: 10.1016/j.knosys.2019.105124.
14. I. Goodfellow, P. McDaniel, and N. Papernot, “Making machine learning robust against adversarial inputs,” *Commun. ACM*, vol. 61, no. 7, pp. 56–66, Jun. 2018, doi: 10.1145/3134599.
15. C. Cisco, “Cisco IoT System Security: Mitigate Risk, Simplify Compliance, and Build Trust,” *White Pap.*, 2015.
16. Bandaru, S. R., Shyam, D., Manoharan, P., Syed, M. H., Ragireddy, U. K., & Addepalli, P. V. (2023). Data-Driven Approaches for Forecasting Cost Overruns in Infrastructure Projects. *European Journal of Technology*, 7(5), 23-34.
17. Dhuli, S., Prabu, M., Muzaffer, H. S., Uday Kumar, R., & Prasanth Varma, A. (2023). A Survey on Critical Success Factors Influencing Project Success Across Multiple Industries. *J Contemp Edu Theo Artific Intel: JCETAI*-106.

18. Ragireddy, U. K., Addepalli, P. V., Bandaru, S. R., Shyam, D., Manoharan, P., & Syed, M. H. (2023). An Overview of Different Automation Strategies Used in Cloud-Based CI/CD Pipelines for Software Deployment. *International Journal of Technology, Management and Humanities*, 9(01), 116-127.
19. Addepalli, P. V., Bandaru, S. R., Shyam, D., Manoharan, P., Syed, M. H., & Ragireddy, U. K. (2023). Performance Analysis of Machine Learning Models for Cybersecurity Risk Classification in IT Projects. *Universal Library of Engineering Technology*, (Issue).
20. Mylavarapu, K. B., Pilli, J., Jannu, P. K., Mohammad, J. A., & Panchali, S. H. (2023). Artificial Intelligence for Continuous Quality Assurance in Cloud-Native Applications: A Review. *European Journal of Technology*, 7(5), 35-49.
21. Jannu, P. K., Mohammad, J. A., Panchali, S. H., Mylavarapu, K. B., & Pilli, J. (2023). A Survey on Energy-Efficient Microarchitecture Design Strategies for Mobile CPUs and GPUs: Techniques and Future Directions. *International Journal of Technology, Management and Humanities*, 9(04), 137-151.
22. Taluri, R. (2021). Cloud-Native Architectures for Enterprise Financial Data Management, Analytics, and Regulatory Reporting Compliance. *International Journal of Emerging Trends in Computer Science and Information Technology*, 2(2), 101-111. <https://doi.org/10.63282/3050-9246.IJETCSIT-V2I2P112>
23. Taluri, R. (2022). Cloud Data Engineering Strategies for Large-Scale Financial Data Integration and Intelligent Corporate Performance Reporting. *International Journal of Emerging Research in Engineering and Technology*, 3(4), 176-188. <https://doi.org/10.63282/3050-922X.IJERET-V3I4P119>
24. Keshireddy, S. R., Nagumotu, V. T., Kavuluri, H. V. R., Pathani, A. K., Dasari, A., & Chilakapati, V. K. (2023). An AI-Based Framework for Detecting IoT Botnets Through Network Traffic Analysis and Modeling. *Universal Library of Engineering Technology*, (Issue).
25. Kavuluri, H. V. R., Pathani, A. K., Dasari, A., Chilakapati, V. K., Keshireddy, S. R., & Nagumotu, V. T. (2023). Cloud Data-Center Network Verification: Approaches, Algorithms and Toolchains. *American Journal of Computing and Engineering*, 6(2), 18-32.
26. Chilakapati, V. K., Keshireddy, S. R., Nagumotu, V. T., Kavuluri, H. V. R., Pathani, A. K., & Dasari, A. (2023). A Survey on 5G/6G Radio Access Networks: Technologies, Protocols, and Performance Metrics. *International Journal of Technology, Management and Humanities*, 9(02), 79-93.
27. Keshireddy, S. R., Nagumotu, V. T., Kavuluri, H. V. R., Pathani, A. K., Dasari, A., & Chilakapati, V. K. (2023). An AI-Based Framework for Detecting IoT Botnets Through Network Traffic Analysis and Modeling. *Universal Library of Engineering Technology*, (Issue).
28. M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020, doi: 10.1109/COMST.2020.2988293.
29. A. N. Jahromi, S. Hashemi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "An Enhanced Stacked LSTM Method With No Random Initialization for Malware Threat Hunting in Safety and Time-Critical Systems," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 5, pp. 630–640, Oct. 2020, doi: 10.1109/TETCI.2019.2910243.
30. M. P. Bharati and S. Tamane, "NIDS-Network Intrusion Detection System Based on Deep and Machine Learning Frameworks with CICIDS2018 using Cloud Computing," in *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)*, 2020, pp. 27–30. doi: 10.1109/ICSIDEMPC49020.2020.9299584.
31. Y. Li, K. Xiong, T. Chin, and C. Hu, "A Machine Learning Framework for Domain Generation Algorithm-Based Malware Detection," *IEEE Access*, vol. 7, pp. 32765–32782, 2019, doi: 10.1109/ACCESS.2019.2891588.
32. M. Singh, B. M. Mehtre, and S. Sangeetha, "User Behavior Profiling using Ensemble Approach for Insider Threat Detection," in *ISBA 2019 - 5th IEEE International Conference on Identity, Security and Behavior Analysis*, 2019. doi: 10.1109/ISBA.2019.8778466.

33. S. Kumar, A. Viinikainen, and T. Hamalainen, "Evaluation of ensemble machine learning methods in mobile threat detection," in 2017 12th International Conference for Internet Technology and Secured Transactions, ICITST 2017, 2018. doi: 10.23919/ICITST.2017.8356396.
34. S. S. Panwar*, P. S. Negi, and Y. P. Raiwani*, "Implementation of Machine Learning Algorithms on CICIDS-2017 Dataset for Intrusion Detection using WEKA," *Int. J. Recent Technol. Eng.*, vol. 8, no. 3, pp. 2195–2207, Sep. 2019, doi: 10.35940/ijrte.C4587.098319.
35. A. Kushwaha, P. Pathak, and S. Gupta, "Review of optimize load balancing algorithms in cloud," *Int. J. Distrib. Cloud Comput.*, vol. 4, no. 2, pp. 1–9, 2016.
36. S. S. S. Neeli, "Serverless Databases: A Cost-Effective and Scalable Solution," *Int. J. Innov. Res. Eng. Multidiscip. Phys. Sci.*, vol. 7, no. 6, p. 7, 2019.
37. K. Atefi, H. Hashim, and T. Khodadadi, "A Hybrid Anomaly Classification with Deep Learning (DL) and Binary Algorithms (BA) as Optimizer in the Intrusion Detection System (IDS)," in 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), Feb. 2020, pp. 29–34. doi: 10.1109/CSPA48992.2020.9068725.
38. J. Lee, J. Kim, I. Kim, and K. Han, "Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles," *IEEE Access*, vol. 7, pp. 165607–165626, 2019, doi: 10.1109/ACCESS.2019.2953095.