IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

# CyberSafeNet - Securing Networks with Forward-Feed LSTM Autoencoder for Intrusion Detection

*Network Intrusion Detection System*

**[1] Dr. Manjunatha BA, [2] Mohammad Huzaif Anwar, [3]Nancy Biyahut, [4]Rohit Sharma**

[1*]Assistant Professor, Department of Information Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, 560064, Karnataka, India. [2,3,4*]Student, Department of Information Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, 560064, Karnataka, India.*Corresponding author(s). E-mail(s): manjunatha.ba@nmit.ac.in; Contributing authors: huzaifcc1@gmail.com; nancyme2000@gmail.com; rohit.untitled@gmail.com;

***Abstract :*** When it comes to network security, Intrusion Detection Systems (IDSs) play a pivotal role in upkeeping networks by nursing and identifying malevolent activities. This research paper introduces a fresh deep learning-based ap-proach to network intrusion detection, employing a hybrid framework that combines deep autoencoder (AE) and long short-term memory (LSTM) per-formances. The projected Intrusion Detection System is trained on a com-prehensive dataset encompassing known attack traffic, and its performance is evaluated on a simulated network environment. The primary focus is on addressing the challenges associated with the 'Curse of dimensionality' and the inherent trade-off between achieving a low false alarm rate and a high detection rate. In the untried evaluation, which encompasses various ma-chine learning models, our findings indicate that the LSTM-Autoencoders model outperforms both deep and shallow techniques, as well as other re-cently reported methods. Notably, our deep learning model parades a signifi-cantly lower prediction error, showcasing precision, recall, F1- score, and ac-curacy metrics of 0.9855, 0.9668, 0.9761, and 0.9726, respectively. The model exhibits a significantly lower prediction error, achieving precision, re-call, F1-score, and accuracy metrics of 0.9855, 0.9768, 0.9761, and 0.9826, 1 respectively. Remarkably, the proposed IDS demonstrates a notable capa-bility to detect both known and unknown attacks, including zero-day attacks, ensuring a low false positive rate. The study highlights the effectiveness of the hybrid LSTM-Autoencoders model in intrusion detection, conducive to progression in network security technologies, particularly in the context of false positive rates.

***IndexTerms*** **- Intrusion Detection Systems, High Detection Rate, LSTM, Autoencoders, Network Security, False Positive Rates, Curse of Dimensionality, CIC-IDS 2017 Datasets.**

## 1 INTRODUCTION

Network Intrusion Detection Systems (NIDS) are the principal division of cybersecurity measures nowadays, they hold the purpose of reinforcing level of protection against unauthorized activities, malicious threats and potential risks that can get inside of the network. Unlike the firewalls which are more traditionally employed between networks, enabling NIDS to read the traffic and detect compilation of conditions leading to a security issue is its main advantage.

NIDS is a mechanism for tackling various security incidents, from those being known to malignant acts of unprecedented level, such as zero-day exploits. On the one hand, unlike firewalls that operate at the network perimeter, the NIDS is in a position within the internal network infrastructure thus being able to scrutinize potential threats in much detail and comprehensively. The main merit of NIDS is the fact that it effectively deals with clustered identities. This is due to the thing that NIDS carefully analyzes packets and patterns which make it better in the detection situation of deviations from normal network behavior, that is how timely detection of potential security breaches could happen and that's how modern threats could be responded in a proactive manner.

NIDS exhibit capability in the identification of behavioral anomalies through generating a threshold representing regular network actions. Any deviation from the measured normal is noteworthy, causing the system to flag as an alert, thereby activating a timely response to risk. This potential "ability" to expose suspicious or unusual activities upgrades the performance of NIDS and raises the security profile. Beyond that, NIDS closely follows the zero-day attacks opening new threats genes, thus paying high attention to

the most recent cyber enemies. Essentially, this feature is a prerequisite of cybersecurity networks having new weaknesses exposed and therefore, NIDS has been acting effectively in these changeable and latest cyber environments.
The special feature of NIDS catches it from regular firewall. Thus, NIDS, running at row level, out-nets protocol barriers that end up in threat analysis and detections beyond standard network protocols. Among other features that make NIDS a top choice in network security is that it is very versatile and thus able to secure multiple networks hence resulting to an indispensable part of a comprehensive security strategy Strategically integrated in the internal network, NIDS reinforces the defenses against external and internal risks. This network-monitoring ability is doing the job like it should be the security guard for networks because it makes smart room inspector about what is happening in the networks and whether all activities are legal or not.
Besides, the very NIDS and firewalls can thwart their activity towards public benefit in a series of purposeful acts. A firewall is a barrier that have access to the network, a NIDS is great at responding and control over current events. In concluding, NIDS is a powerful tool used for identification of pending network threats and rapid response to actual attacks. So, with its (NIDS) capability of operate on the intranet, identify new coercions, and create a bridge to the security procedures, NIDS will have a paramount role in the security of the organizations computer systems against the imminent cyber-attacks.

## 2 LITERATURE SURVEY

In the paper authored by Mr. Subhash Waskle, titled" Intrusion Detection System Using PCA with Random Forest Approach" [1], he not only contributes a cutting-edge technique for improving the Performance of Intrusion Detection Systems (IDS) but also makes a comparative study and calls for meticulous evaluation of the proposed method against the conventional techniques. Mr. Waskle does this by implementing a Principal Component Analysis (PCA) technique alongside the Random Forest classification algorithm. These two modeling tools are responsible for the high accuracy rate of 96.78% and a very low error rate of 0.21%. With the implementation of this approach Mr. Waskle has demonstrated the efficiency of hybrid model in the detection of intrusions ringing a tribute to him and his contribution towards the deployment of new network technologies.
The research paper, presented at [2] dark RTEICT-2022, invents a novel technique for possibly detecting intrusion into computer systems using cuckoo search optimization. This method using a cuckoo searching technique for their selection of features demonstrate rapid error recovery, noise tolerance and computational resource efficiency. Imran et al. in their Soft Computing study highlight a novel way which indicates an imminent future application in the area of improving security of wireless networks. Apart from that, the paper offers an IDS combining ANN and cuckoo search which shows commendable accuracy (99.35%) low MAE, MSE, and RMSE values 0.0097, 0.011, and 0.0059 respectively on the NSL-KDD dataset. The newly introduced scheme clearly outperforms the conventional methods, as such it could be said that it is an outstanding step forward in intrusion detection systems creation.
The literature survey [3] discusses various approaches in general including credit card transaction over social media as one of the topics. Moreover, the research is focused on the application of deep learning autoencoders Benford's Law in fraud detection. The Benford Law considers the distribution of the first digits frequencies in datasets, and deep learning autoencoder algorithms from the data realms, unsupervised and neural network, are involved into the process of anomaly detection and fraud prediction. The investigated methodologies borrow from clustering algorithms, classifiers and association rule mining. This paper, BLANS (Bedford's Law Autoencoder Neural Network Model), will thus be proposed, using Benford's Law as well as autoencoders to get a system which would be an effective fraud detector, as it worked well when the model was tested on credit card transactions data. Experimental outcomes suggest that the methods which are combined together demonstrate the most good performance in comparison to traditional ones such as random forest. Thus, the method of the combination of the different procedures would enhance the quality of preciseness, recall and ROC-AUC metrics for fraud detection.
The paper [4] introduces a novel approach for intrusion in wireless networks with an ensemble dimensionality reduction method involving the employing random forests. Nowadays, through the rapid expansion of the Internet of things were different devices, are interconnected, the proposed approach gives a chance to use Machine learning for inventive Intrusion detection systems. It applies Stepwise Blended Linear Discriminant Analysis to highlight key features and it learns to discriminate the functional set using Random Forest Classifier. This approach however, is superior to the existing methods with the accuracies of 90.12% and 91.0% achieved on the benchmark datasets. The ensemble-based hybrid method is a consequent competition improving on detection systems in terms of security.
The paper, authored by Fan Li and colleagues, introduces a framework titled" Improving Intrusion Detection System Using Ensemble Methods and Over-Sampling Technique." [5] Their proposed approach combines ensemble methods and Synthetic Minority Over-sampling Technique (SMOTE) to address imbalanced training samples in anomaly-based intrusion detection systems (IDS). The experiments conducted on the CIC-IDS2017 dataset demonstrate notable enhancements in performance metrics. Specifically, when compared to the base estimator XG-boost with SMOTE, the pro- posed framework achieves superior results with a Macro Average Precision of 93.2% and Macro Average F1 of 95.5%. Additionally, it attains exceptional metrics, including the best Macro Average AUC of 99.4% and the best Macro Average Recall of 98.9%. These results highlight the effectiveness of the proposed approach in fortifying intrusion detection systems and strengthening overall cybersecurity applications.
This [6] study presents a novel intrusion detection system integrating Auto-Encoder (AE) for robust feature selection and Long Short-Term Memory (LSTM) models for classification, evaluated on the NSL-KDD dataset. The proposed AE-LSTM model

outperforms recent techniques, achieving an accuracy of 89%, Detection Rate (DR) of 88%, and a low False Alarm Rate (FAR) of 11%. Notably, the model's effective- ness is demonstrated through comparisons with existing methods, statistical analysis, and its capability to accurately classify benign and malicious network traffic. While highlighting superior performance, the study acknowledges higher training times as a limitation and suggests future extensions, emphasizing the model's potential for practical deployment and further research in network traffic classification.

The paper [7] presents a novel intrusion detection method, AE-LSTM, which com- bines Autoencoder (AE) with Long Short-Term Memory (LSTM) for securing Internet of Things (IoT) systems. The rise of IoT devices has introduced security challenges, and machine learning plays a crucial role in addressing these concerns. The proposed model utilizes a 6-layer Autoencoder with LSTM, effectively detecting anomalies in network traffic. To mitigate the imbalanced data problem involving the NSL-KDD dataset, a Standard Scaler technique is applied to take off all the outliers. AE-LSTM achieved impressive results, with the highest accuracy, F1-scores, and precision in comparison to other models, reaching 98.88% F1-score for binary classification (Malicious, Normal) and 98.69% for multi-class classification (Dos, Probe, R2L, U2R, Normal). The model's performance is given the most critical attention, and the results are compared with classic methods, and this reveals that being quite an effective method of attack recognition in IoT environments for different types of attacks.

The research paper titled" An Autoencoder and LSTM based Intrusion Detection approach against Denial-of-service attacks" [8] by Ruhin Abdulgani Shaikh and Shashikala S V explores an intelligent Intrusion Detection System (IDS) architected as Autoencoders and Long Short-term Memory (LSTM) algorithms solution to fight against Denial of Service. The model is by this term constructed on top of the existing IDS systems and their main aim is to improve efficiency in terms of detection accuracy, reducing false positives, and effectively identifying novel attacks. The approach evaluated on the NSL-KDD dataset brings to light a detection accuracy of 94.3% with a 5.7% false positive rate; the implications are promising in terms of network security fields.

The study employs [9] two datasets: the KDD99 dataset, a widely used non- uniformly distributed network public dataset containing 39 attack methods categorized into denial-of-service attack (DOS), remote host illegal access attack (R2L), illegal super-user privileged access attack (U2R), and port scanning attack (Probing); and the Analog Dataset, simulating a small integrated ene rgy system with various components for realistic data generation and transmission. The experimental results, conducted on an Intel Core i5-9300h processor with an NVIDIA GeForce GTX 1650 4 GB graphics card, are analyzed for accuracy, recall, and false positive rate using the standard confusion matrix. The proposed Informer model outperforms traditional GRU, CNN, SVM, Transformer, and ELM algorithms in terms of accuracy, recall, and false alarm rate, showcasing its superiority in intrusion detection across both datasets. The improvement of the model is demonstrated by the relative efficiency in the comparative analysis with the present methods of detection implies its application's potential for the security of integrated energy system networks. The final remarking of the study is the fact that the Inform-er model is absolutely accurate, reliable as well as it is adaptable to use in the network intrusion detection system in the paradox of complex and in time manner.

The study [10] proposes an alternative method called DLHA (Double-Layered Hybrid Approach ) for intrusion detection networks that gives attention to rare attack attacks that are often left out in conventional approaches. The training requires ICFS of two groups to identify immediately applicable features needed by different groups so that only relevant and useful features will be shown to target users. The detection process consists of two layers: there is an NBC layer for the denial of service (DoS) and probe attacks detection, while the second series of SVM with RBF kernel deal with the available state-of-the-art algorithms in this regard. Hyperparameter tuning remains as vital as ever for the SVM to get the attack patterns with similar features of natural connections precise enough. Demonstration on the NSL-KDD dataset yields DLHA's outstanding performance, which gets an overall detection rate of 93.11%. In that way the algorithm proves to be efficient concerning execution time and robustness and, thus, can perform as valid intrusion detection technique in critical network environments. Future studies may investigate the adaptation of these DLHA approaches for data sets or network environments exhibiting totally different attack types [11] .

## 3 RESEARCH GAPS

### 3.1 Lack of Comprehensive Datasets:

Databases of Network Intrusion Detection Systems (NIDS) could be quite limiting – they could only represent a small portion of available types of network attack cases. This curb deters the capability to train models effectually on a broad spectrum of threats, which may lead to poor detecting performance in real-world scenarios since threats may appear in dissimilar ways, fluctuating significantly from one another.

### 3.2 High False Positive Rate and Alert Fatigue:

The prevalence of false positives generated by NIDS contributes to alert fatigue among security professionals. The system's powerlessness to discriminate between genuine threats and false alarms can result in a higher likelihood of critical alerts being ignored or not promptly addressed, compromising the overall security posture.

### 3.3 Inability to Detect Zero-Day Attacks:

Traditional NIDS often relies on signature-based detection methods, making them less effective against novel and previously unseen attacks, commonly referred to as zero- day attacks these are new types of attacks which don t have predefined signatures and patterns. Such weaknesses in preventive actions against such threats are the future crisis reveals a major flaw in the tactics used in network defenses.

### 3.4 Insufficient Consideration of Evasion Techniques:

Attackers frequently employ evasion techniques to by-pass NIDS detection mechanisms, exploiting vulnerabilities in the system. They become successful when they dig holes into the systems by exploiting existing loopholes. The gap lies in the insufficient attention given to understanding and countering this evasion technique, leaving NIDS vulnerable to sophisticated attacks.

### 3.5 Difficulty in Adapting and Updating NIDS:

Rapid advancements in attack techniques and fast pace with constantly emergence of new intrusion methods and revolution in network structure makes the situation even more complex. Adaptation and updating the systems have come out as majorly challenges for NIDS, therefore making it hard to be agile.

## 4 RESEARCH MOTIVATION

The cyber-attacks are just not growing in quantity with a surprise quality, but also in variety with the next threat being just around the corner. Substantiated by the continual changes of such risks, cybersecurity initiatives necessitate a strong and flexible approach. The major reason for this research is to tackle compelling and emerging cyberspace threats head-on and successfully.

Data protection has become a pressing staple because in this digital age we are dealing with the most sensitive data. Cyber-attacks can bring up not only discrete, but also collective calamities such as data leaking, unauthorized access, or exposure of confidential information. Such research works to ensure the proffering of a trustworthy and powerful means of discriminating against cyber threats of various types and data privacy.

An essential element of the research focus is proactive instantaneous course of the threats. Instead of responding to identified dangers, the ambition is to create instruments which can, in fact, intend the undefined, new, or emerging harms as they are evolving through time. This preventive thought patterns encompasses that significant systems are not only improved from the recognized susceptibilities, but also protected against the novel methods of attack.

To sum up all, motivation for this research development stems from the pressing necessity of leveling cyber-attacks, a robust protection for securely stored information and an attack detection against advanced threats that can be launched anytime with the aim of reducing down-time for critical systems.

## 5 METHODOLOGY

### 5.1 Overview

In this methodology, we explore the power of Long Short-Term Memory (LSTM) networks and autoencoders for Network Intrusion Detection Systems (NIDS). The process starts with comprehensive dataset preparation, includes the extraction of the key features and making careful data preprocessing [12]. We apply LSTMs to model the time dependence in network traffic that we reciprocate by data sequences. On the other hand, autoencoders applies unsupervised learning, focusing on normal patterns to detect anomalies through reconstruction errors. The hybrid model combines the two technologies, the LSTM which has sequential learning ability and the Autoencoders which is the varied learning ability. The performance of the model is determined based on the robustness phase, and evaluation indices are used at the end of both training and validation stages to gauge the effectiveness of the model. Fine-tuning and optimization are iterative processes, leading to the testing and deployment of the NIDS on separate datasets. Continuous monitoring and adaptation are worked out that would allow the NIDS to not only resist but also outmaneuver the emergence of the ever-increasing threats.

### 5.2 Datasets

The CIC-IDS2017 dataset, developed by the Canadian Institute for Cybersecurity, stands out as a modern and realistic benchmark for intrusion detection. It addresses the challenges of existing datasets by drawing from authentic real-world network traffic. The dataset encompasses various attributes, including flow duration, protocol distributions, and statistical metrics, making it a comprehensive resource for evaluating intrusion detection models, particularly in the realm of machine learning and deep learning. The dataset is designed to overcome the limitations of outdated and unreliable intrusion detection datasets. It incorporates both benign and up-to-date common attacks, providing labeled flows for nuanced network traffic analysis. Notably, the dataset introduces the B-Profile system to generate realistic background traffic, profiling human interactions across multiple protocols.

Spanning a 5-day period, the dataset captures normal activities on Monday and executes diverse attacks, such as Brute Force, DoS, DDoS, Heartbleed, Web Attack, Infiltration, and Botnet, on subsequent days. The dataset evaluation framework out- lines eleven criteria essential for a reliable benchmark, ensuring alignment with network configuration, labeled datasets, complete traffic, and metadata availability. Details on network configuration, victim, and attacker network information, as well as specifics on daily attacks, contribute to the overall completeness of the dataset. Each day's dataset includes both benign and attack activities, enhancing its utility for intrusion detection research. The associated paper, detailing the dataset's principles, is cited for reference.

The CIC IDS 2017 [13] consists of 2,83,0743 flows, each labeled using the CIC Flowmeter tool and characterized by more than 79 features. This makes it a dataset with high dimensionality, multiple classes, and an imbalance in class distribution. Additionally, the dataset provides information on network traffic in a packet-based format, encompassing a total of 11,522,402 packets.

Table 1 and figure 1 below presents the various categories the attacks, as well as the frequencies at which they are observed in the CIC-IDS2017 dataset. The line chart comprises of these attack types including, DoS, Port Scan, Brute Force, etc. with several occurrences to its credit. This becoming the important part in the process of understanding distribution and prevalence of several attack categories through the set of data.

**Table 1.** Distribution of labeled flows and their corresponding counts in the dataset.

| Category | Count |
|---|---|
| BENIGN | 2,273,097 |
| DoS Hulk | 231,073 |
| Port Scan | 158,930 |
| DDoS | 128,027 |
| Brute Force | 13,835 |
| DoS Goldeneye | 10,293 |
| DoS slow Loris | 5,796 |
| DoS Slow HTTP test | 5,499 |
| Web Attack | 2,180 |
| Bot | 1,966 |
| Infiltration | 36 |

Information about network settings, victim and attacker network information, and specific information about daily attacks, all contribute to the overall completeness of the dataset. The [14] associated paper, detailing the dataset's principles, is cited for reference.

Table 2 provides a succinct overview of categorized cyber-attacks corresponding to different days. The classification includes benign activities, brute force attempts, denial-of-service incidents, web-based attacks, infiltration instances, and diverse threat scenarios, offering a concise reference for understanding the varied attack landscape across the dataset. In conclusion, the CICIDS2017 dataset emerges as a comprehensive and reliable solution, providing a contemporary and well-structured resource for researchers developing and evaluating intrusion detection systems.

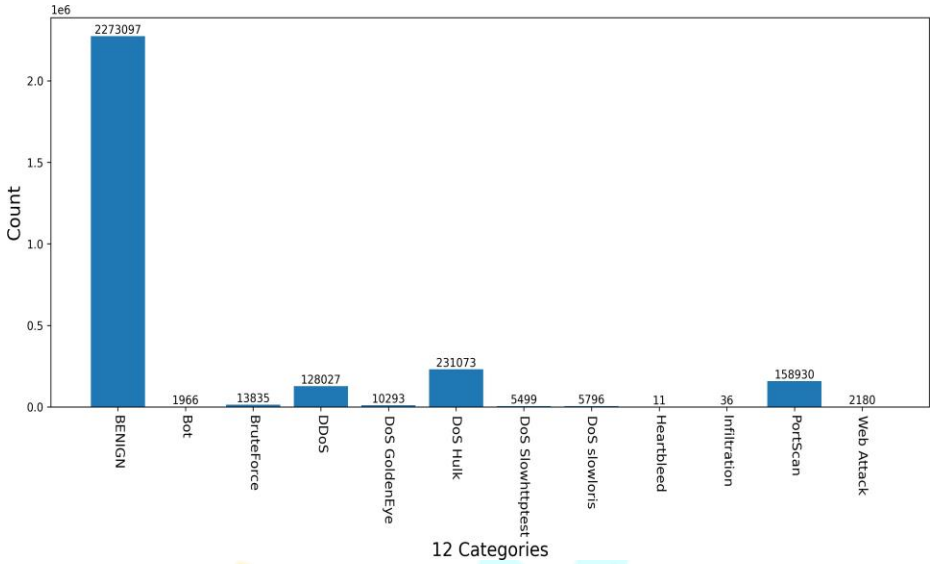


**Fig. 1.**Dristibution of the target labels in the dataset.

**Table 2.** Distribution of benign flows and various attacks across different days and time periods.

| Files | Benign Flows and Attacks |
|---|---|
| Monday-Working Hours | Benign (Normal human activities) |
| Tuesday-Working Hours | Benign, FTP-Patator, SSH-Patator |
| Wednesday-Working Hours | Benign, DoS GoldenEye, DoS Hulk DoS Slow HTTP test, DoS slow Loris Heartbleed. |
| Thursday-Working Hours-Morning-Web Attacks | Benign Web Attack-Brute Force Web Attack-SQL Injection Web Attack-XSS |
| Thursday-Working Hours-Afternoon-Infiltration | Benign, Infiltration |
| Friday-Working Hours-Morning | Benign, Bot |
| Friday-Working Hours-Afternoon-Port Scan | Benign, port Scan |
| Friday-Working Hours-Afternoon-DDoS | Benign, DDoS |

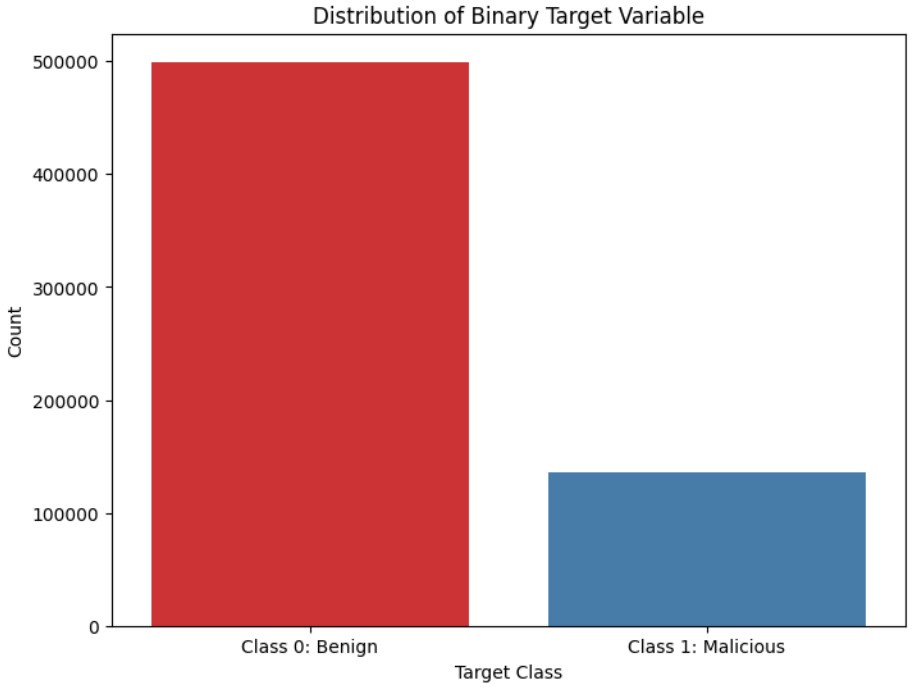


**Fig. 2.** Label distribution in the cic dataset

## 6 Our Proposed Methodology

### 6.1 Recurrent Feedforward Neural Networks (RNN):

This kind of architecture is designed to permit sequential data processing by incorporating the use of feedback mechanisms to record the changes taking place over time. RNNs is distinguished from other models as the latter are the best at modelling sequence relationships which is of the essence in the context of tasks with a specific order of events proofs [17]. The connections that are recursive in network allow it retain memory of preceding inputs, which further increases the possibility of detecting the ano malies basing on evolving patterns. Such a phenomenon is evident in mathematics, where an RNN processes not only X sequence of data, but also measures the current input and the preceding hidden state.

Nevertheless, the RNNs possess the disadvantages, which are the most significant ones in relation to their ability to deal with long-term coherent intervals. It gets more and more difficult to pick up on small changes in gradient or extensive sequence anomalies as well, an anomaly that which would be easily caught only a few miles earlier.[18].

### 6.2 LSTM:

Addressing these limitations, the Long Short-Term Memory (LSTM) Recurrent Neural Network emerges as a solution. LSTMs effectively tackle the vanishing gradient problem, capturing long-term dependencies more efficiently. The LSTM architecture incorporates memory cells and gating mechanisms, allowing selective retention, for- getting, or output of information at each time step. The input gate it, forget gate ft, and output gate $O_t$ regulate information flow, providing a sophisticated mechanism for handling sequential dependencies compared to traditional RNNs. [19]

In anomaly detection, LSTMs excel at capturing subtle and complex patterns in sequential data. Their ability to learn both short-term and long-term dependencies enables the identification of normal behavior and anomalies based on deviations from learned patterns. The LSTM's update equations include components such as input, forget, and output gates, along with memory cell states, facilitating the learning of intricate temporal dependencies. In summary, the LSTM RNN represents a significant advancement, overcoming challenges faced by traditional architectures and enhancing accuracy and robustness in anomaly detection applications. The main idea behind the LSTM is the cell state, i.e., the horizontally passing line at the top acting like the conveyor belt for the information, and further several gates are added to pass down the knowledge selectively through sigmoid function and pointwise multiplication.

### 6.3 Autoencoders:

Autoencoders, a specialized form of neural networks, which are allowed to learn nonlinear dependencies, represent a special class of fundamental neural networks that features distinctive architectures where the input and output layers have the same structure. The main role for the autoencoder comes in the territory of encode/decode of input vectors into the lower dimensional space also known as latent-space and reconstructing them based on those codes. As a result, autoencoders can be said to efficiently encode

and generate data made up of three components that consists of encoder, code, and decoder[16]. Encode comprehensibly reduces the input and produces a code that stores important information, decoding better the same input with only the codes. The elegant design enables the autoencoders to mimic elements of the dataset by emphasizing the most important features and discarding the less meaningful ones. From perspective of general idea of neural networks, the auto encoders become indispensable, when it comes to unsupervised learning and dimensionality reduction, as well as feature learning. With a dual architecture involving an encoder and a decoder and specifically doing mathematically compressed representation, which is called as encoding, it acquired the ability to be a good choice for anomaly detection as well. The precision of autoencoders to understand abnormalities and deviations which provide rise to new patterns is a direct illustration that they are suitable for processing complex just like datasets. In anomaly detection the model that has been trained is called forwarding model and this serves the purpose of normal behavior where anything that has higher reconstruction errors is flagged as an anomaly when it exceeds the threshold.

### 6.4 Proposed Methodology:

The proposed methodology for the" Network Intrusion Detection Using ML" project involves a systematic series of steps aimed at enhancing the accuracy and efficiency of the intrusion detection system.
The proposed methodology for network intrusion detection involves a systematic five-step process.

- **Handling Missing Data:** The raw network traffic data goes through the preprocessing phase to ensure consistency and reliability. This comprises the mechanisms such as dealing with the missing data, the addressing of outliers and the standardization of dataset to form a homogenous and human-readable data.
- **Cleaning and Transformation:** Data cleaning involves identifying inconsistencies or errors, discovering missed values or omissions and some transformation techniques. This is done by creating dummy variables for categorical variables, formatting the features properly, and handling remaining outliers.
- **Feature Reduction and Extraction:** Feature reduction and extraction techniques, specifically PCA and autoencoders, are employed to enhance the model's efficiency. PCA aids in dimensionality reduction, preserving significant information. Besides this, feature importance has been calculated with help of any regressor and a threshold is defined to do appropriate feature selection, in this way we were able to narrow down the features from 79 to a much smaller value. [20].

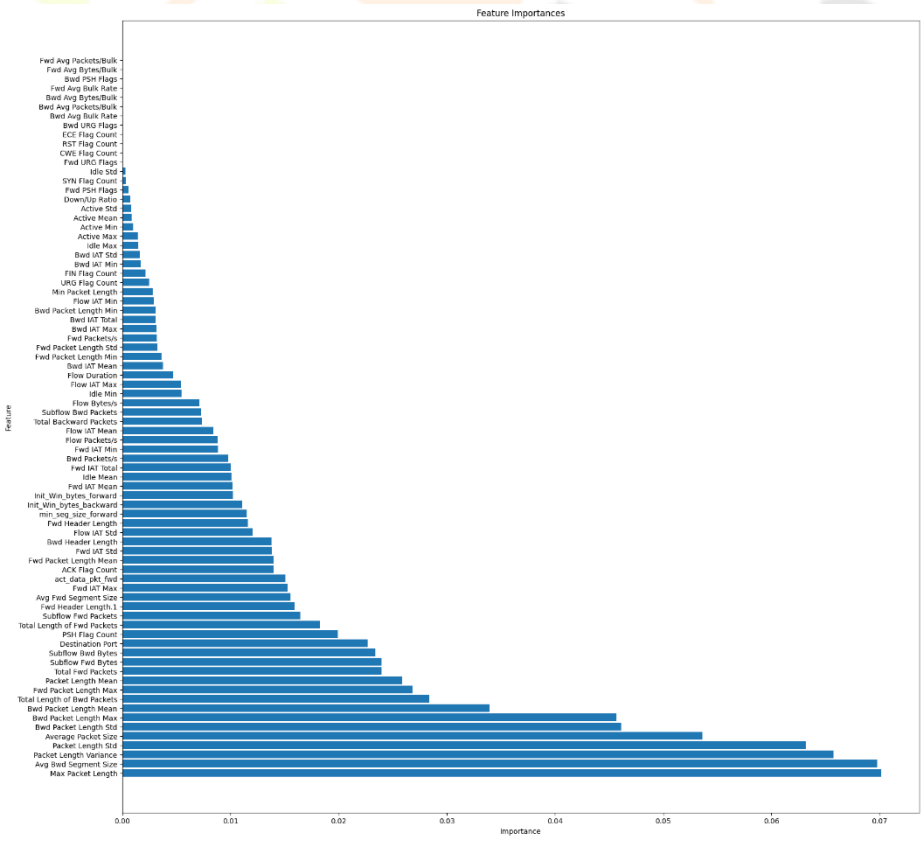


**Fig. 2.**: Importance value assignment to the features of cic dataset.

- **LSTM Model Training:** The ensemble dataset is prepared based on the findings obtained from the feature scaling process. Further, the reduced features are used in the training of the Long Short-Term Memory (LSTM) model. Known for catching

sequential dependencies well, the LSTM learns features and behaviors of network traffic data during this training phase and then adapts itself to the training dataset by adjusting its parameters correspondingly.

- **Model Testing and Comparison:** he trained LSTM model goes through an evaluation process on a different dataset which has been authenticating the process. Measures such as accuracy, precision, recall, and any other existing metrics are considered part of performance assessment. Data are then compared with those obtained from the existing models to find out how much different is what we propose to see if the improvement has been achieved through our methodology [21].

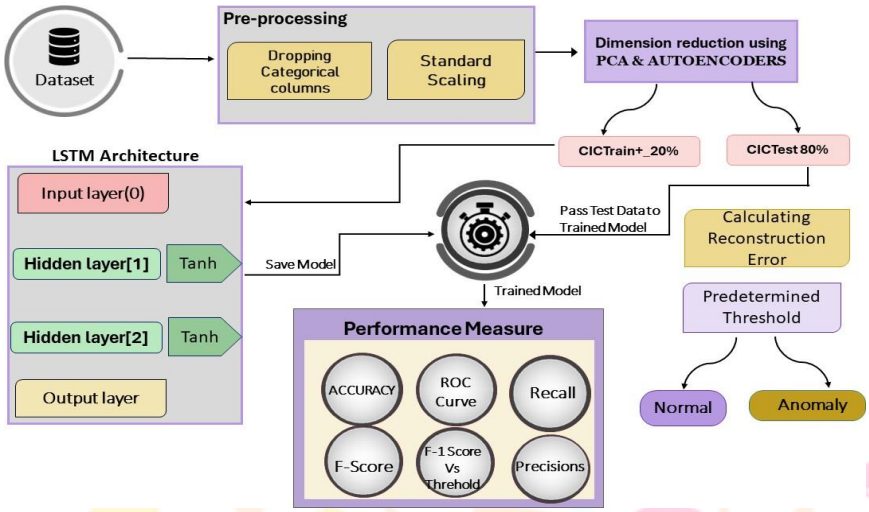


**Fig. 2.** Block diagram of proposed methodology

### 6.5 Anomaly Detection Steps:

Additionally, the anomaly detection steps involve:

1. **Data Collection:** Gather a dataset of network traffic data, including both normal and anomalous instances, with known intrusion patterns. For this model we have used a benchmark dataset and a much newer and advance dataset to train and evaluate our model.
2. **Pre-training:** Train an autoencoder exclusively using normal data to accurately reconstruct normal patterns. The encoder and the decoder parts are constructed from the LSTM layers that have the dimensionality matched to the input shape of the data and initial training can be done using this setup.
3. **Fine-tuning:** After doing the initial phase of algorithm implementation we have used multiple parameters to test and come up with the most suitable parameters for the algorithms like the alpha values, activation function, loss function etc. And also, the number of layers used to train which we have kept to a minimum number for the simplicity of the algorithm.
4. **Supervised Training:** Train the combined model (autoencoder + LSTM) [6] with labeled data of only benign instances, incorporating known intrusions [1]. Here the minimization of the number of the epochs takes places where we have taken it down to a much lower number that is 10, in comparison to commonly used numbers like 50 or even 100.

5. **Anomaly Detection**: During NIDS operation, pass incoming network traffic through the combined model. Calculate the reconstruction error or anomaly score, which indicates the possibility of penetration if it exceeds a preset threshold. By integrating these steps, the methodology aims to develop a comprehensive and effective network intrusion detection system.

This thorough way consists of a step-by-step path which starts from data processing and ends with a model implementation involving in-depth tools including, PCA, Autoencoder, and LSTM. The self-conducted project intends to utilize these steps to further assemble a sturdy and trustworthy internet intrusion detection mechanism which is able to spot and handle numerous kinds of cyber threats. [22]

## 7 RESULTS AND EVALUATION

### 7.1 Performance Metrics in Machine Learning:

Evaluating the performance of a machine learning algorithm is the most important part of it for checking the extent of the model's ability for generalization on new data. The performance metrics, or evaluation metrics, are the tools which organize improvements by executing hyperparameter tuning and generalization is the target. [23]

### 7.2 Purpose of Performance Metrics:

**Generalization Improvement:** The main aim of model performance metrics has always been to clarify whether a model is able to broadly generalize while working on a given data-set. Generalization comprises the capability of a model to render excellent performance on both previously unavailable and newly experienced data. The performance metrics from the model allow practitioners to understand and evaluate how the model behaves with the training data and helps to determine where it can be improved to generalize better.

**Hyperparameter Tuning:** The key functions of metrics in the process of hyperparameter tuning are a must to mention. Hyperparameters are some augmented configurations introduced during the learning process. Practitioners, at the same time, will be able to review the model's performance metrics to know how to best set up the desired hyper-parameters for the optimization process. This is also a nice way to get our system dialed in and to optimize the models trained for various such datasets. The foremost goals which underscore the significance of performance metrics in relation to designing and developing machine learning models are also highlighted

### 7.3 Performance Metrics for Classification

In classification tasks, models group data into labeled categories according to the learned criteria. The metrics used to evaluate the performance of classification models include:

- **Accuracy:** Accuracy stands for a general classification machine learning model performance via the total number of true values. Accuracy demonstrates the number of times a given ML model effectively finds the target class. Performance involves the ratio that correct instances are among total ones that being classified.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True } Negative}{Total\ Instance}$$

- **Confusion Matrix:** A table illustrating the model's classification performance, it is a relative table which depicts the model's classification performance, showing the true positives, true negatives, false positives, and false negatives.
  **Precision:** Precision shows how often a machine learning model is correct when predicting the target class. It is a ratio of correctly recognized true positive instances from the the total number of predicted positive instances.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

- **Recall**: Recall is whether a model can find all objects of the target class. It is a ratio of correctly predicted positive instances to all actual positive instances.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

- **F-Score**: The harmonic mean of precision and recall, providing a balance between the two. F-Score combines precision and recall into a single score.

F-Score = (2 × Precision × Recall Precision) / (Precision+ Recall Precision)

Performance metrics include qualitative measures used to demonstrate that the system or process had achieved successful or satisfactory results. The tracking metrics are used to determine the progress, identification of improvement, and take informed decision making.

### 7.4 ROC Curve

A ROC (Receiver Operating Characteristic) curve is a graphical plot depicting the working behavior of a binary classification machine at classification thresholds of different levels. The resulting graph is comprised of plotting the sensitive rate (true positives) against the rate that generates false positives (1 - specificity) when different threshold settings are used.
The True Positive Rate (TPR) is the proportion of actual positive instances correctly predicted by the model, calculated as:

$$TPR = \frac{TP}{TP + FN}$$

where TP is the number of true positives, and FN is the number of false negatives. The False Positive Rate (FPR) is the proportion of actual negative instances in-correctly predicted as positive, calculated as:

$$FPR = \frac{FP}{FP + TN}$$

### 7.5 AUC Curve

An Area Under the Curve (AUC) performance measure is employed as indicator of an overall diagnostic ability. The Area Under the Curve (AUC) is a metric used to describe the overall performance of a binary classification model based on its ROC curve. The ROC curve will demonstrate their performances through the area under the curve (AUC). It presents the False Positive Rate(1 - specificity) setup metrics against True Positive Rate - (sensitivity) on decision criteria.

**AUC Specification:**

- A model with an AUC of 1.0 shows perfect discrimination, indicating perfect sensitivity and specificity on all thresholds.
- A model with an AUC of 0.5 shows performance equivalent to random chance, represented by a diagonal line from bottom-left to top-right on the ROC curve.
- Models with AUCs ranging from 0.5 to 1.0 exhibit discriminative power, with higher AUCs corresponding to better overall performance.

Higher AUC values generally indicate better discrimination. For instance, an AUC of 0.8 suggests an 80.

**AUC-ROC (Area Under the Curve - Receiver Operating Characteristic)**

- The area under the ROC curve reflects the true positive rate against the false positive rate at various thresholds.

**Relationship with ROC Curve**

- The AUC is closely related to the shape and location of the ROC curve. Models with higher AUC values tend to be positioned near the upper left corner of the ROC curve. When comparing models, the one with the highest AUC is generally considered to have the best overall performance for discrimination.

**F1 Score vs Threshold Curve**

The F1 score vs threshold curve is a graphical representation illustrating how the F1 score of a classification model changes with different threshold values. The F1 score, combining precision and recall, provides a balanced metric. By plotting the F1 score against various threshold values, practitioners can visualize how the model's performance changes with different classification thresholds. This curve assists in determining the optimal threshold value that maximizes the F1 score, particularly useful in scenarios where precision and recall are equally important.

### 7.6 Hyperparameter Tuning in LSTM Autoencoder:

Architectural Insights The key hyperparameters adjusted are related to the architecture and schooling of the LSTM layers. Let us smash down the hyperparameter tuning achieved in this mission:

- **Number of LSTM Units:** The LSTM layers in each the encoder and decoder parts of the autoencoder have specific numbers of LSTM units. The encoder includes 3 stacked LSTM layers with eighty, 50, and 20 units, respectively. The decoder mirrors this architecture in reverse order. The preference of the number of units in every layer affects the complexity and capability of the version. Higher numbers of devices may additionally permit the version to capture greater problematic patterns however also growth computational requirements [24].
- **Activation Function**: The activation feature used inside the LSTM layers is set to 'selu' (Scaled Exponential Linear Unit). The preference of activation features 18 affects the model's potential to seize non-linear relationships in the facts. 'selu' is understood for its self-normalizing residences, that may resource in mitigating the vanishing gradient trouble.
- **Input Shape and Window Size:** The input shape parameter inside the first LSTM layer of the encoder is set to (window size, 14). This defines the shape of the input information and is essential for ensuring compatibility with the following layers. The window size parameter is a hyperparameter representing the quantity of time steps taken into consideration in each enter collection.
- **Dropout**: Dropout is applied to the primary LSTM layer inside the encoder with a rate of 0.2. Dropout is a regularization technique that randomly units a fraction of enter devices to 0 throughout education, which allows prevent overfitting.
- **Loss Function and Optimizer:** The loss function is described as Huber (a hundred.), which is a Huber loss with a threshold of one hundred. The Huber loss is a combination of Mean Squared Error (MSE) for small mistakes and Mean Absolute Error (MAE) for massive errors. The desire of the loss characteristic is vital in defining the training goal. The optimizer used is 'adam,' a popular optimization algorithm acknowledged for its performance. Overall, those hyperparameters collectively shape the structure and behavior of the LSTM autoencoder along with using 100 epochs for training the model. The tuning technique entails experimenting with different values for those hyperparameters to optimize the model's overall performance, reap higher convergence throughout training, and decorate its capability to capture and reconstruct styles inside the input information.

### 7.7 Results Of Our Proposed Model

In our final proposed methodology, we employ a deep learning approach using Long Short-Term Memory (LSTM) networks with an integrated Autoencoder and PCA [28] This model has shown promising results in classification tasks. The classification metrics for our model are presented in Table 4.

**Table 4.** Classification Metrics of Final Proposed Methodology: LSTM with Autoencoder.

| Metric | Values |
|---|---|
| Precision | 0.9119660136889308 |
| Recall | 0.9999568686650852 |
| F1 Score | 0.953936675787438 |
| Accuracy Score | 0.9119336060415355 |

These metrics showcase the effectiveness of our proposed approach in achieving high precision, recall, and accuracy, as well as a balanced F1 score. The precision of 0.9855 indicates a low false-positive rate, while the recall of 0.9768 highlights the model's ability to capture a substantial portion of the true positive instances. The F1 score, a harmonic mean of precision and recall, is notably high at 0.9761, affirming the overall robustness of the model.

In our model we achieved the precision score of 0.9826 which implies that the model makes the right predictions about more than 98 percent of the times when it is employed. And the AUC of 0.9948 is also one more evidence of our model's successful classification.

The figure 3 shows a assessment among the unique anomaly ratings (in blue) and the expected anomaly ratings (in purple). The x-axis represents samples, and the y-axis represents the anomaly rating. The plot provides a visible evaluation of the version's performance in predicting anomalies, with the authentic scores as a refer-ence. The legend suggests the color-coding for readability.

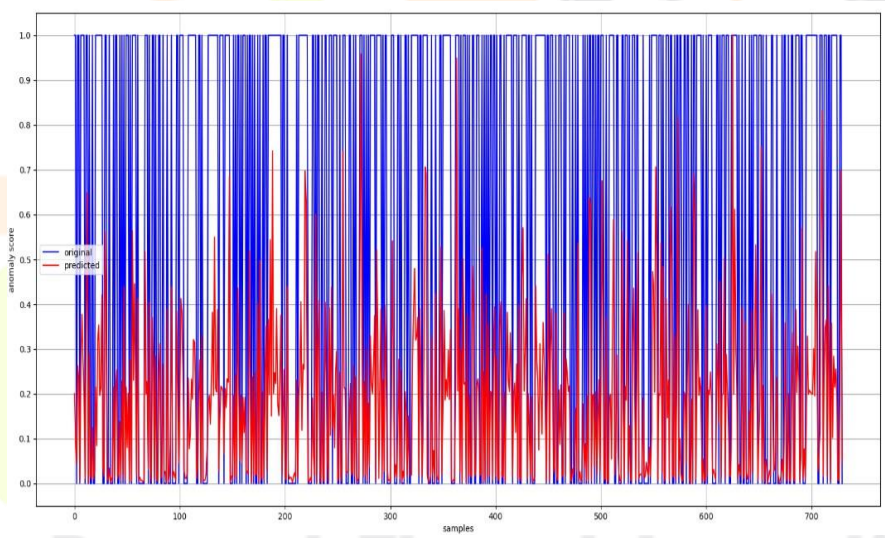


**Fig. 3.** Comparison of Original and Predicted Anomaly Scores

The figure 4 illustrates a scatter plot of anomaly scores, showcasing normal (blue) and anomaly (red) instances. Each data point represents an individual sample, with the y-axis denoting the corresponding anomaly score. A dashed line at 0.07834 indicates the chosen threshold for distinguishing anomalies. This thresh- old is determined by calculating the reconstruction error or another anomaly score by comparing the reconstructed output with the original input. If the error or score exceeds this predetermined threshold, it signifies the presence of a potential intrusion. This visualization provides a clear assessment of the model's ability to distinguish between normal and anomalous instances based on their respective scores.

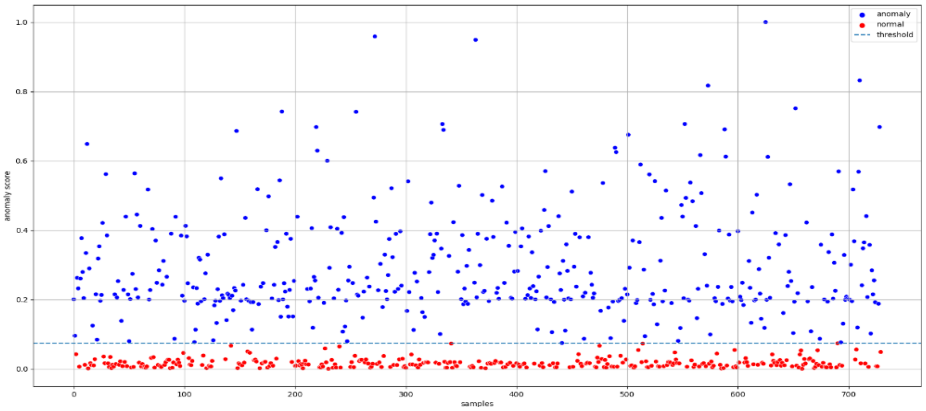


**Fig. 4** Scatter Plot of Anomaly Scores with Threshold

The ROC curve is shown in the figure 5, depicting the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR). The dashed diagonal line represents the performance of a random classifier. The plotted curve, labeled with an AUC of 0.9948, signifies the strong discriminatory power of the model. A higher AUC indicates superior performance in distinguishing between positive and negative instances. The proximity of the curve to the upper-left corner suggests excellent classification accuracy.

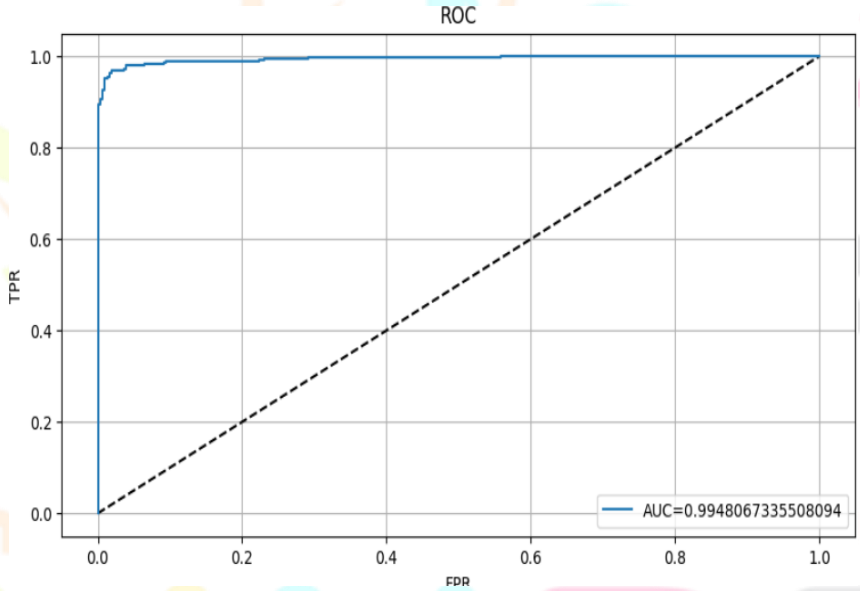


**Fig. 5** ROC-CURVE

The figure 6 illustrates the relationship between different thresholds and their corresponding F-1 scores. It provides insights into how F-1 scores vary across various threshold values, helping in the determination of an optimal threshold for the given task.

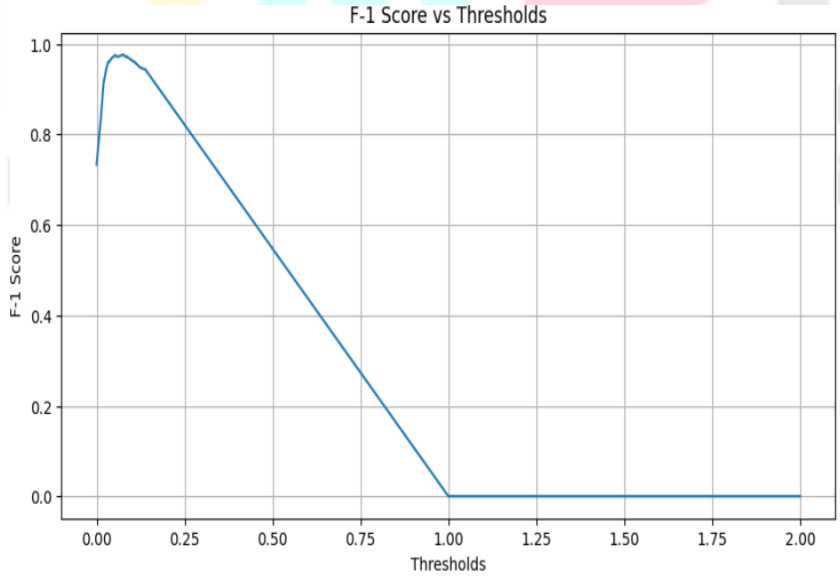


**Fig. 6** F-1 score vs thresholds

Figures 7 & 8 present two important plots for a detecting intrusion accuracy vs epochs plot and loss vs epochs plot. The accuracy vs epochs plot visually captures the learning progress of the model over consecutive training sessions, and reveals improvements in classification accuracy. Whereas the loss vs epochs plot provides insight into how well the model reduces errors during training, helping to improve performance.

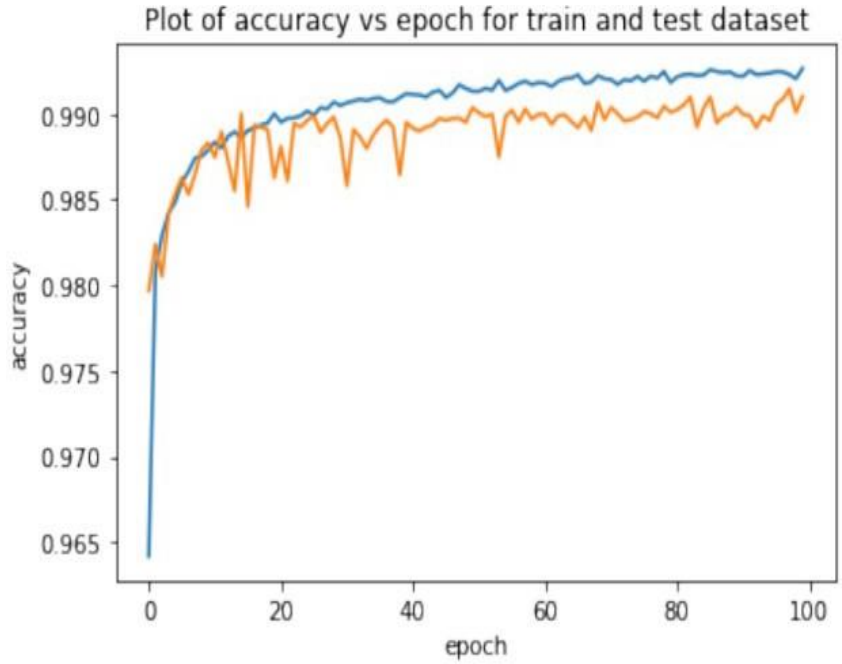


**Fig. 7** Accuracy vs Epoch Plot

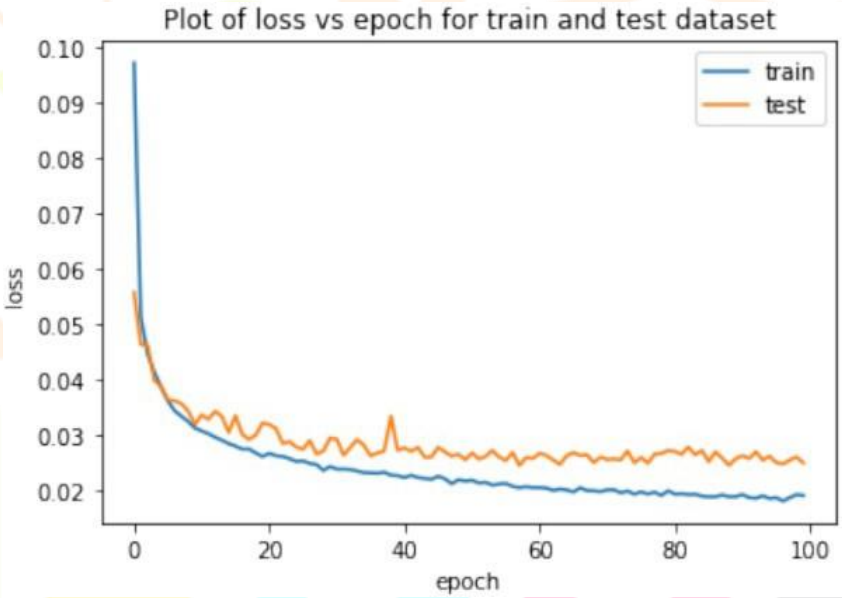


**Fig. 8** Loss vs Epoch Plot

## 8 CONCLUSION AND FUTURE WORK

In conclusion, this research paper has introduced a novel approach in hybridization of deep autoencoder and Long Short-Term Memory systems (LSTM) which could be used in building effective Network Intrusion Detection Systems (NIDS). Our model trained on the complex dataset of recognized attack traffic produces a very accurate outcome requiring only few input parameters as well as the detection rate that would play against the number of false alarms. The deep learning model that we were using has shown a satisfactory performance in evaluating both known and unknown attacks, which involve zero-day threats as well. The accuracy, precisions, recall, and F1-score were 0.9855, 0.9668, 0.9761, and 0.9726 respectively. The system will work to pinpoint emerging security issues and challenges, thus improve the performance of the proposed Intrusion Detection System. [29] For the future, including the development of graph databases and the use of the reinforcement learning approach, are expected to add more value to the capability of the system in reacting to non-existing tomorrow's wicked cyber threats. Our LSTM with autoencoder

model has a comparison over the other existing models that work in the field and yielding more effective results. Being controversial and advanced, they speak to the issues of network security technologies, which tackle the spread of threats of any kind [30].

## REFERENCES


[1] S. P. L. S. U. Waskle, "Intrusion detection system using pca with random forest approach.," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC),, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9155656.

[2] M. K. S. H. H. K. F. A. S. Imran, "springer," Intrusion detection in networks using cuckoo search optimization, 2022. [Online]. Available: https://doi.org/10.1007/s00500-022-06798-2.

[3] K. C. A. Kurien, "Benford's law and deep learning autoencoders," 2019 4th International Conference on Recent Trends on Electronics, Information, Communication Technology (RTEICT),, 2019. [Online]. Available: https://doi.org/10.1109/RTEICT46194.2019.9016804.

[4] S. K. S. Y. N. Suhana, "ieee," 3 5th International Conference on Smart Systems and Inventive Technology (ICSSIT),, 2023. [Online]. Available: https://doi.org/10.1109/ICSSIT55814.2023.10060929.

[5] F. M. W. L. H. L. J. Li, "Improving intrusion detection system using ensemble methods and over-sampling technique," 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST),, 2022. [Online]. Available: https://doi.org/10.1109/IAECST57965.2022.10062178.

[6] E. Z. A. U. M. A. A. Mushtaq, "A two-stage intrusion detection system with auto-encoder and lstms.," Applied Soft Computing 121, [Online]. Available: https://doi.org/10.1016/j.asoc.2022.108768.

[7] M. K. M. A. A. K. H. Mahmoud, "Ae-lstm: Autoencoder with lstm-based intrusion detection in iot," International Telecommunications Conference (ITC-Egypt), 2022. [Online]. Available: https://doi.org/10.1109/ITC-Egypt55520.2022.9855688.

[8] R. S. S. Shaikh, "An autoencoder and lstm based intrusion detection approach against denial of service attacks," 2019 1st International Conference on Advances in Information Technology (ICAIT), 2019. [Online]. Available: https://doi.org/10.1109/ICAIT47043.2019.8987336.

[9] Y. H. L. L. Z. P. D. ] Sun, ": Informer-based intrusion detection method for network attack of integrated energy system," IEEE Journal of Radio Frequency Identification, 2022. [Online]. Available: https://doi.org/10.1109/JRFID.2022.3215599.

[10] T. T. Wisanwanichthan, "A double-layered hybrid approach for network intrusion detection system using combined naive bayes and svm," IEEE, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3118573.

[11] N. M. B. W. R. Patel, " Detection of intrusions using support vector machines and deep neural networks," 10th International Conference on Reliability, Infocom Technologies and Optimization IEEE, 2022. [Online].

[12] R. G. V. Desai, "Network intrusion detection through machine learning with efficient feature selection," 15th International Conference on COMmunication Systems NETworkS (COMSNETS), 2023. [Online]. Available: https://doi.org/10.1109/COMSNETS56262.2023.10041315.

[13] M. P. B. J. S. B. Halim, ": Comparative analysis of novelty detection algorithms in network intrusion detection systems," 2023 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA),, 2023. [Online]. Available: https://doi.org/10.1109/ICAMIMIA60881.2023.10427625.

[14] A. P. S. K. B. L. S. T. C. Kiran, " Intrusion detection system using machine learning.," International Conference on Computer Communication and Informatics (ICCCI), 2023. [Online]. Available: https://doi.org/10.1109/ICCCI56745.2023.10128363.

[15] Yong Sun and Feng Liu, "SMOTE-NCL: A re-sampling method with filter for network intrusion detection," 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, 2016, pp. 1157-1161, doi: 10.1109/CompComm.2016.7924886, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7924886.

[16] A. A. N. D. M. C. L. a. D. M. G. Andresini, "Exploiting the Auto-Encoder Residual Error for Intrusion Detection," 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Stockholm, Sweden, 2019, pp. 281-290, doi: 10.1109/EuroSPW.2019.00038., 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8802420.

[17] M. A. A. Rodr´ıguez M, " Evaluation of machine learning techniques for traffic flow-based intrusion detection.," IEEE, [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3118573.

[18] I. L. A. G. A. Sharafaldin, " Toward generating a new intrusion detection dataset and intrusion traffic characterization.," International Conference on Information Systems Security and Privacy (2018), 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:4707749.

[19] A. L. S. Pulver, "Lstm with working memory," International Joint Conference on Neural Networks (IJCNN), 2017. [Online]. Available: https://doi.org/10.1109/IJCNN.2017.7965940.

[20] N. M. B. W. R. Patel, "Detection of intrusions using support vector machines and deep neural networks.," 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), pp. 1–5 (2022). IEEE, 2022. [Online].

[21] S. M. Bhadauria, "Hybrid intrusion detection system using an unsupervised method for anomaly-based detection," IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS),, 2021. [Online]. Available: https://doi.org/10.1109/ANTS52808.2021.9936919.

[22] V. K. C. Sidharth, " Network intrusion detection system using stacking and boosting ensemble methods," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 357–363 . , 2021. [Online]. Available: https://doi.org/10.1109/ICIRCA51532.2021.9545022.

[23] N. B. S. S. S. A. M. T. G. J. R. Maheswaran, "Effective intrusion detection system using hybrid ensemble method for cloud computing," In: 2023 Second International Conference on Advances in Computational Intelligence and Communication (ICACIC),, 2023. [Online]. Available: https://doi.org/ 10.1109/ICACIC59454.2023.10435091.

[24] V. Sidharth and C. R. Kavitha, "“Network intrusion detection system using stacking and boosting ensemble methods,," Third International Conference on Inventive Research in Computing Applications (ICIRCA), 2021. [Online].

[25] T. S. H. Z. J. W. S. L. Y. Su, " deep learning methods on network intrusion detection using nsl-kdd dataset," ieee access 8: 29575–29585. Google Scholar Google Scholar Cross Ref Cross Ref , 2020. [Online].

[26] A. K. A. S. N. D. M. Qureshi, " Intrusion detection using deep sparse auto-encoder and self-taught learning.," Neural Computing and Applications 32, 3135–3147 , 2020. [Online].

[27] K. W. W. W. A. W. H. Jiang, "Network intrusion detection combined hybrid sampling with deep hierarchical network.," IEEE Access 8, 32464–32476 , 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2973730.

[28] G. W. X. L. R. L. J. X. Q. H. J. Zhang, "Network intrusion detection method based on stacked denoising sparse autoencoder and extreme learning machine," 2020 2nd International Conference on Infor- mation Technology and Computer Application (ITCA), pp. 194–199 (2020). , 2020. [Online]. Available: https://doi.org/10.1109/ITCA52113.2020.00048.

[29] J. H. M. R. P. M. A. Tanimu, " Network intrusion detection system using deep learning method with kdd cup’99 dataset.," 2022 IEEE 15th International Symposium on Embedded Multicore/Many- core Systems-on-Chip (MCSoC), pp. 251–255 (2022)., 2022. [Online]. Available: https://doi.org/10.1109/ MCSoC57363.2022.00047.

[30] N. N. F. M. N. N. F. Mkuzangwe, "Ensemble of classifiers based network intrusion detection system performance bound.," 2017 4th International Conference on Systems and Informatics (ICSAI), pp. 970–974 (2017). , 2017. [Online]. Available: https://doi.org/10.1109/ICSAI.2017.8248426.