University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)

ISSN: 2714-3627

A Journal of the Faculty of Computing, University of Ibadan, Ibadan, Nigeria

Volume 16 No. 1, January 2026

journals.ui.edu.ng/uijslictr
http://uijslictr.org.ng/
uijslictr@gmail.com

University of Ibadan
Journal of Science and Logics in ICT Research

NETPA-DLA: A Deep Learning–Based Network Packet Analyzer for DDoS Detection

[1,2]✉ Isiekwene C. C., [1]Azeez N. A., [1]Akinboro S. A. and [3]Asokere M. M.

[1]*University of Lagos, Akoka, Yaba, Lagos.*
[2]*Miva Open University, Utako, Abuja.*
[3]*Lagos State University, Ojo, Lagos.*
isiekwene.chioma@miva.university, nazeez@unilag.edu.ng, sakinboro@unilag.edu.ng, mauton.asokere@lasu.edu.ng.

Abstract
In the digital era of internetworked systems, understanding, analysing and filtering network traffic is crucial for maintaining security, optimal performance, conducting diagnostic routines and monitoring. This research developed a novel Deep Learning based Network Packet Analyzer (NETPA-DLA) which utilizes an optimal hyperparameter dynamic technique. An ensemble deep learning approach that integrates Deep Belief Networks (DBN), Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), Auto-encoders, Transformers and U-Net for robust and accurate classification of distributed denial-of-service (DDoS) was used. The ensembled model was trained on the CIC-DDoS2019 dataset. The study findings contribute to the continuous refinement and deployment of advanced measures to strengthen digital infrastructure against evolving threats. The experiment on the non-pretrained DBN model proved to be better than the pretrained counterpart for DDoS detection, with an accuracy of 99.72 % and false positives of 37 and false negatives of 13 on the validation dataset, with results for all metrics for the LSTM model at 0.9998, the least being validation specificity at 0.9855. Transformer had the highest accuracy level of 0.9998, closely followed by Autoencoder, which had an accuracy level of 0.9986, and ensemble weighted voting at 0.9984, while the RNN obtained a perfect score of 1.0000 for both Recall and Sensitivity across the three relative weights for each of the models. The study shows that DBN can accurately detect and predict DDoS while maintaining the security of the system and given access to the necessary user of the system without any form of denial.

Keywords: *DDoS attacks, Mitigation, Networks, IP packets, Hyperparameter Tuning, Deep learning*

1. Introduction

Smart cities leverage on technologies to improve resource use and reduce emissions, making cities more socially inclusive and creating open and transparent administration with active citizen participation [1-3]. Communication and information access have been made easier by the widespread use of internet-connected digital devices and the continuous advancement of network technologies, forming a smart city. However, these technological advancements have increased cybercriminals in the smart city, making customers' access to a wide range of services unavailable. Network security follows rules when handling malicious requests, but customers encounter problems such as unavailable web resources and loss of sensitive information [4, 5]. Therefore, several strategies, including decision theory, stochastic simulation and game theory, have been investigated to mitigate these attacks [6]. However, these approaches are not capable of identifying attacks on a global scale because the system is local and is not familiar with such attacks [7, 8]. Furthermore, the most vulnerable industries are those in healthcare, IT, finance, higher education, telecommunications, energy, and government [9-12]. Attackers may differ in their intentions to start an assault since there are five main reasons why they could launch one: financial gain, retaliation, ideological conviction, intellectual challenge, and cyber warfare. Given that these attacks are increasing, thus, it's critical to identify and stop assaults early on before they reach their target. It's getting harder to identify Distributed Denial of Service (DDoS) attacks [13, 14]. However, if the attack is successful, how does the system recover from such, and how much time is required to recover? Is the

Isiekwene C. C., Azeez N. A., Akinboro S. A. and Asokere M. M. (2026). NETPA-DLA: A Deep Learning–Based Network Packet Analyzer for DDoS Detection. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 16 No. 1, pp. 66 - 88

recovery total or partial? How much damage is done?

Machine Learning (ML) is a subset of artificial intelligence, subdivided further into Reinforcement, Unsupervised, and Supervised Learning, respectively. ML has enhanced data representation, thereby promoting data storytelling; however, the focus will be on its subset "Deep Learning (DL)". These techniques are frequently used to improve data processing, such as quickly forecasting an event's result, rather than to precisely duplicate human procedures. Additionally, DL algorithms may require certain conditions for effectiveness, thereby presenting a need for hyperparameter tuning [15].

Objective: This paper implemented the detection and mitigation of distributed denial of service (DDoS) attacks in the network model. the Network Packet Analyzer-Deep Learning Approach (NETPA-DLA) aims to capture, balance, analyse and predict network packets. The specific objectives of the study include:

i) Design an analyser on which the IP packets will be analysed for DDoS and non-DDoS attacks using a Deep learning approach.
(ii) Validate the NETPA-DLA model based on performance, time, and error rates.
(iii) The Level of intelligence at which the system would make decisions on its own, given certain conditions, will be viewed via a visualization board.

1.1 NETPA-DLA Architecture

NETPA-DLA stands for Network Packet Analyzer using Deep Learning Approach. It is a technique designed to automatically capture and store a variety of DDoS attacks from different IoT and IIoT devices globally to enable it recognize DDoS attacks in the Smart City Industrial Internet of Things (SC-IIoT) environment as shown in Figure 1. The approach involves several key steps:

Classification is one of the techniques in data mining to allocate objects to one of several predefined groups [1]. Data mining extracts interesting, non-trivial, implicit, previously unknown and potentially useful patterns or knowledge with the help of various techniques in the data gathered from the various sources. Data mining also involves selecting relevant data from the database, pre-processing and cleaning the relevant data, as well as transforming into a suitable form, mining and evaluating the data and afterwards online updating and visualisation. The actual task of data mining is a semi-self-regulating or mechanical investigation of large batches of the dataset for extracting the previously unknown, unusual records and dependencies.

The knowledge discovery process involves various selection steps which help in the efficient extraction of the useful data from databases. Furthermore, data mining is one of the essential steps in the KDD process [2].

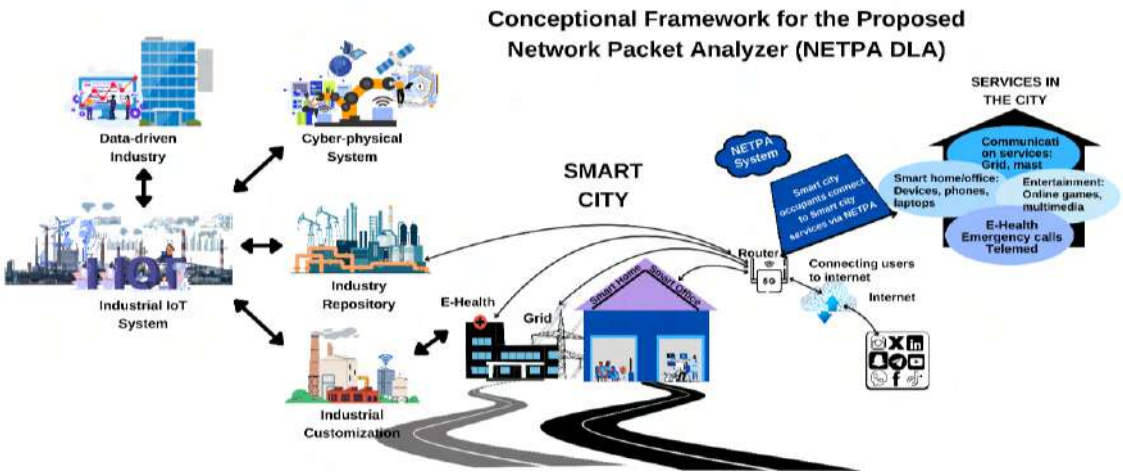


Figure 1: *Conceptual Framework of the Proposed Network Packet Analyzer (Driving a Secured Smart City)*

1. Data Normalization: Standard scaler was used, ensuring that the input data is scaled appropriately for enhanced performance, and it is more robust to outliers.

2. Feature Selection: trimmed down our dataset using feature selection by correlation, neural networks automatically select features during training, improving efficiency and accuracy.
3. Ensemble Deep Learning: Integrating Deep Belief Networks (DBNs), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Auto-Encoders, and Transformers for robust and accurate classification.
4. Hyperparameter Optimization: Using batch normalization, hidden dimension, dropout, and reconstruction weight.
5. DDoS Types: Using the CIC-DDoS2019 dataset. These include SYN floods, DNS, LDAP, MSSQL, TFTP, UDP, UDP-Lag.

The NETPA-DLA method aims to provide a comprehensive and effective framework for extracting, classifying and detecting DDoS-attacks, optimizing data processing and model performance to achieve improved accuracy and robustness in handling intrinsic classification threats. To achieve this, we divide the whole process into two modules and four sub-modules:

Module One: This is also known as the Application-Based Mitigation Strategy, where we employ the use of expert systems in the design and architecture of the IIoT infrastructure within the organization. This setup includes the use of firewalls, a load balancer, Imperva system for:

i. Packet monitoring and storage module
ii. Data collection process

Module Two: Here, we leverage Machine learning and Deep learning algorithms to perform analysis on the collected and stored data in the application-based mitigation strategy procedure. This process has two sub-modules, namely:

iii. Detection in figure 3b.
iv. Mitigation

1.2 Application-Based Mitigation Strategies as shown in figure 3a.

1. Load Balancing: Use load balancing techniques to distribute traffic across multiple servers.
2. Application Firewalls: Use application firewalls to filter out malicious traffic.

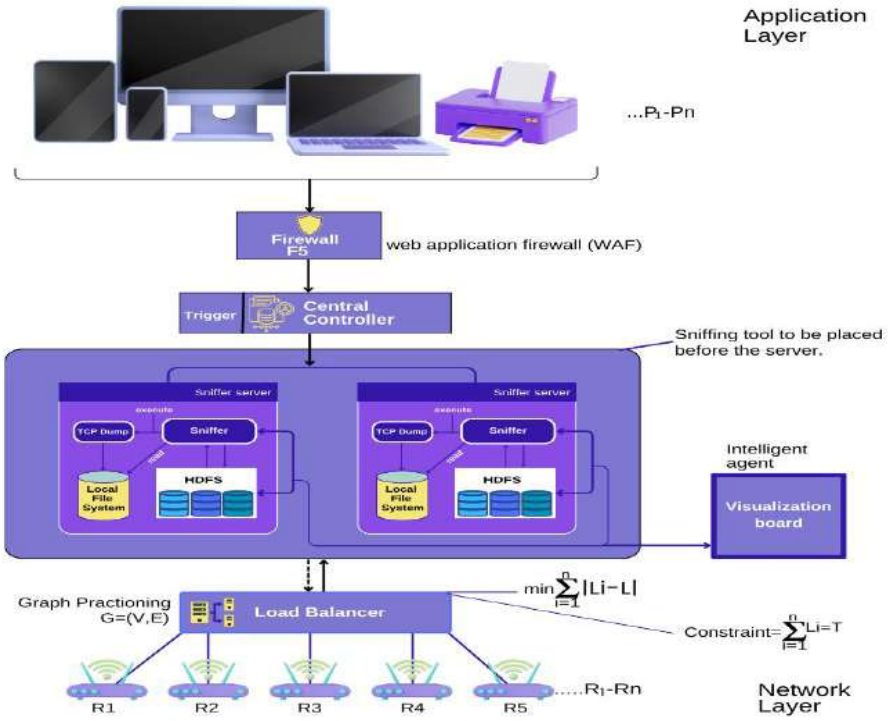


Figure 3a: *Framework for DDoS Mitigation on Application-Based*

Recovering from a DDoS attack requires a mix of immediate actions, such as traffic filtering, blocking IPs, or changing the compromised IP address, and long-term solutions. Analyzing the attack afterward is crucial to developing future mitigation strategies. Since IP addresses are assigned either dynamically (through DHCP) or statically, in the event of an attack, the compromised IP can be blocked, and a new one assigned. Having a proactive plan that includes incident response procedures is essential to effectively handle DDoS attacks upon detection as seen in figure 3b, and thereby ensuring network security. The focus will be on detection.

2. Related Works

A distributed denial-of-service (DDoS) attack is a deliberate attempt to disrupt the normal operations of a server, service, or network by overloading the target or the infrastructure around it with excessive volumes of Internet traffic [16-18]. DDoS assaults are effective because they can generate attack traffic from many infected computer systems. One may classify computers and other networked resources like the Internet of Things devices as exploited machines. In essence, a denial-of-service assault is similar to unanticipated traffic jams that obstruct highways and prevent regular traffic from getting where it needs to go [19].

Three broad categories are used to classify attack detection techniques [20]: hybrid, anomaly, and signature-based. By comparing their signatures, the signature-based approach recognizes attacks that have already been reported [21, 22]. On the other hand, anomaly-based techniques recognize typical network behavior before identifying anomalies that deviate from typical traffic [23]. They are effective since they can identify attacks that aren't known about. The methodologies of both anomaly-based and signature-based approaches are combined in hybrid techniques.

The ML feature selection approach aims to find the optimal feature set to create optimized models of analyzed phenomena. Feature selection is the process of determining which is most relevant and non-redundant to use while building a model. It is important to reduce the size of the dataset [6]. The practice of effectively defending a targeted server or network against a DDoS assault is known as DDoS mitigation. A targeted victim can lessen the impact of the intruding threat by employing specifically made network equipment or a cloud-based protection service [24-26].

The term "hyperparameter" refers to a set of variables that are adjusted to control the behavior of an algorithm or model and generate an improved model with the best performance [27] as shown in Figure 2. Essentially, hyperparameter tuning is the act of fine-tuning the model's parameters over a period. Before model training, model hyperparameters are manually changed.

.

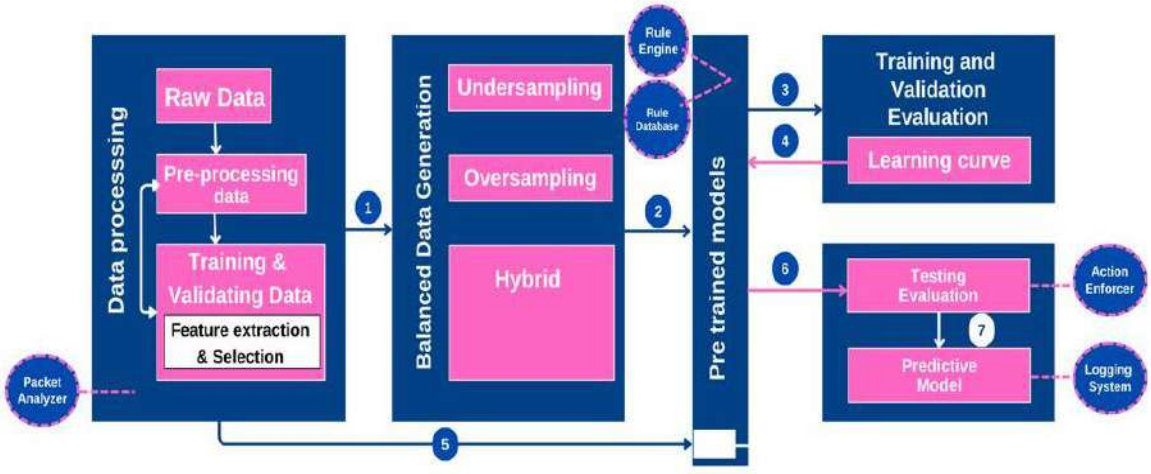


Figure 3b: Architecture for DDoS detection

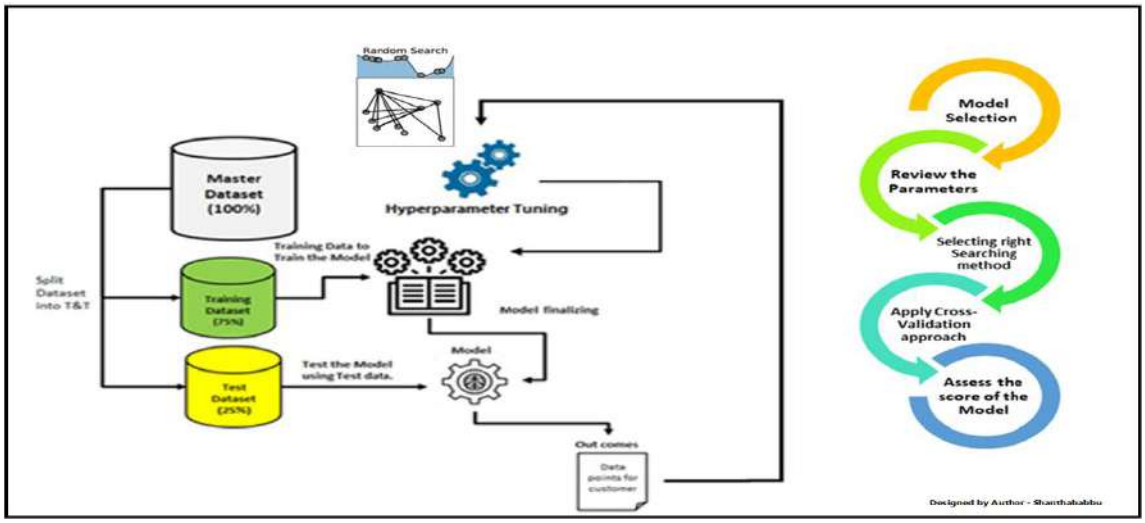


Figure 2: *Hyperparameter Stages and Integration* [28]

A contractive autoencder–based deep learning model was proposed for anomaly detection by learning compact representations of normal network traffic with a stochastic threshold employed for attack identification. The model was evaluated on three benchmark intrusion detection datasets which are CIC-IDS2017, NSL-KDD, and CIC-DDoS2019 and compared with a standard autoencoder and other deep learning approaches. Experimental results demonstrate effective intrusion detection achieving accuracy rates of 93.41%–97.58% on CIC-DDoS2019, 96.08% on NSL-KDD, and 92.45% on CIC-IDS2017 [29].

Comparative evaluation of Random Forest, LightGBM, XGBoost, and AdaBoost revealed that Random Forest consistently achieved the best performance for DDoS attack detection and classification across both datasets (Slowloris and CICDDoS2019). Its high accuracy (99.5%–99.8%) and strong F1-scores (92.4%–97.3%) highlight its effectiveness and robustness in distinguishing DDoS traffic from normal network behavior [30].

A CNN–LSTM based hybrid deep learning model was evaluated against five machine learning classifiers using the CICDDoS2019 dataset for binary and multiclass DDoS detection. The proposed model outperformed all baselines, achieving perfect classification for several attack types and overall weighted F1-scores of 90% [31].

Furthermore, A robust ensemble model combining autoencoder outputs and baseline classifiers with LightGBM significantly improves resilience to adversarial attacks, increasing accuracy by at least 13.2% and F1-score by over 110% without adversarial training [32].

Some ML and DL strategies for detecting and analyzing DDoS attacks. To help decide when to employ which of these techniques, this study also compares and evaluates the important differences between ML and DL techniques [33].

Hybrid of Gated Recurrent Unit and Convolutional Neural Network algorithms. The Canadian Institute of Cybersecurity Intrusion Detection System's benchmark cyber security dataset is used for simulations. With an overall accuracy rate of 99.7%, the simulation results show that the suggested approach performs better than the existing intrusion detection systems [34].

DL-2P-DDoSADF, a two-phase deep learning–based DDoS detection framework, was proposed and validated on the CICDDoS2019 and DDoS-AT-2022 datasets. The autoencoder achieved 99% accuracy in the first phase, while the DNN attained multiclass classification accuracies of 97% and 96% on the respective datasets [35].

A framework for malicious network traffic detection using two classification approaches: a

CNN–GRU deep learning model and an SVM optimized with the Slime Mould Algorithm. Evaluated on the KDD dataset, the optimized SVM achieved accuracies of 98.45% and 94.84%, with performance assessed in terms of accuracy, specificity, and computational efficiency [36].

The proposed ensemble-based approach with hybrid feature selection achieves near-perfect DDoS detection, delivering almost 100% accuracy, a 100% true positive rate, and zero error while improving robustness and reducing overfitting [37].

This suggested approach used a feature selection and Bi-LSTM-based honey badger optimization algorithm to forecast DDoS attacks in a cloud setting. The optimal feature is selected by minimizing the mean square error (MSE) of each feature. The Bi-directional Long short-term Memory (Bi-LSTM) classifier is then fed with the best features to anticipate DDoS attacks. Additionally, the suggested model is investigated concerning some of the current methods, such as ANN, DNN, LSTM, and DBN. The Bi-LSTM model attained 97% accuracy, 95% sensitivity, 90% specificity, 3% error, 94% precision, and so on when the performance was assessed using the current methodology. The suggested methodology works well for detecting DDoS in a cloud setting [38].

Experimented an all-new Optimized Dual IDS that was developed and deployed. When compared to the current IDSs, it is more accurate. The proposed IDS includes the recently created HRDPA data preprocessing technique, which uses the RFE approach, hyper-tuned parameters ML classifiers, and Repeated Stratified K-Fold process to select the best features that are beneficial in increasing the prediction accuracy of the classifier. In the end, a new Deep Grid Network was created that analyses the network dataset more efficiently and achieves better accuracy of 99.99 % by utilizing the machine learning classifiers LC, RF, DT, NB, Linear SVM, and Non-Linear SVM [39].

Standard performance measurements to show a notable improvement for both binary and multiclass classification using the suggested model. A 0.005% reduction in generalization error has been achieved, and an L2 regularisation strategy has been used to address overfitting. With increases in Precision, Recall, and F1-score, the model's overall Accuracy on different datasets is 99.99% for BOT-IoT, 99.08% for IoT23, 99.82% for UNSWNB15, and 99.96% for ToN_IoT, respectively [40].

Results of the experiment show that the Enhanced Random Forest algorithm, also known as ensemble random forest, successfully classifies attacks with an astounding accuracy rate of 99.98%. It will therefore be the initial stage classifier. Moving forward, the OneClass Support Vector Machine (SVM) algorithm will be our second-stage identifier due to its high degree of accuracy, which reaches 99.7% in detecting abnormalities [41].

The IDS accuracy of the models that were trained using the complete feature sets of the KDD-CUP-1999, BotIoT-2018, and NBaIoT-2021 datasets was 91.34%, 11.31%, and 84.39%, in that order. Nonetheless, the models trained using feature subsets that were chosen using feature selectors outperformed the models trained using the entire feature sets of the identical datasets. The models that were trained using feature subsets chosen by the backward sequential feature selector, on the other hand, had the highest accuracy levels, with 95.66% for KDD CUP 2018, 99.48% for BotIoT 2018, and 99.81% for N-BaIoT 2021 [42].

The proposed model shows an adaptable and scalable architecture that enables the meticulous analysis of network traffic data to detect intricate patterns suggestive of DDoS attacks. Performance metrics like detection accuracy and loss rates demonstrate how versatile our approach is across different datasets. With minimal loss rates and detection accuracy rates of 99.98%, 100%, and 99.99% for the InSDN, CICIDS2018, and Kaggle DDoS datasets, respectively, our DNN-based model demonstrates great capacity in combating contemporary DDoS assaults [43].

Illustrated the benefits of the concept through thorough assessments, showing how it can facilitate the portrayal of similarities, guide model classification/unknown assault identification, optimize defense measures, and expedite filtering reactions. For example, the results demonstrate that because of its uniform behavioral representation, only 15 different

types of attacks may be defended with just one rule. the suggestion is to create a DDoS family to address and overcome these problems. Characterizing traffic patterns, creating attack fingerprints, and executing cross-executed family partitions using community detection are all included in the specified technical roadmap [44]. Network packet analyzers with deep packet inspection (DPI) capabilities have enhanced stakeholders' ability to make informed decisions through advanced reporting and visualization features. Graphs, charts, and dashboards present network traffic patterns in a concise and clear manner, aiding in the interpretation of complex information. However, challenges remain, including managing high traffic volumes, processing encrypted data, and dealing with compatibility issues across diverse network configurations [45].

3. Methodology

More than 200,000 rows are available for machine learning to distinguish between benign and DDoS data, collected with CICFlowMeter-V3 from the Canadian Institute for Cybersecurity. This is a subset of their generated data; the packet data, which totals more than 225,000 columns and comes from various devices, includes both benign and DDoS data.

The advancement of cloud computing and distributed systems, along with the rapid adoption of this service due to its low cost and easy accessibility, has presented a new layer of problems in cybersecurity, the most significant of which is availability, and a target for ddos attack perpetrators. the dataset contains features to predict various types of ddos attacks as shown in table 1, including dns, ldap, mssql, tftp, udp, and syn [19]

3.*1 Data preprocessing: Accurately Predicting DDoS Attacks (CIC-DDoS2019 Dataset):*

The original dataset was moderately clean but still required thorough cleaning to ensure it's suitable for model training. Data preprocessing included common data cleaning methods such as normalization, detecting and removing relevant features, reducing the data complexity, and increasing its reliability [46]. The Flow Bytes/s column had 30 and 4 infinite values, and null values, respectively, and the Flow Packets/s column had 34 infinite values, all of which were replaced with the column median value. The Flow ID had no attack-related significance in our modeling and was eliminated.

To handle redundancies in the dataset, 10 columns with constant figures were eliminated, 2 duplicate rows in the categories were removed, and 17 columns with the same values as an already existing column were iteratively eliminated. Using two basic correlation-based feature selection, highly correlated features were grouped using a correlation coefficient of 0.9 as the threshold, and the feature with the best correlation to the label was selected from the group, while others were dropped. Correlation of features was based on the Pearson correlation matrix Figure 4.

Table 1: Description of Datasets

DATASET	NAME	URL	SOURCE
DATASET1	SYN floods	https://www.kaggle.com/datasets/aymenabb/ddos-evaluation-dataset-cic-ddos2019	[19]
DATASET 2	UDP fragmentation and floods	https://www.kaggle.com/datasets/aymenabb/ddos-evaluation-dataset-cic-ddos2019	[19]
DATASET 3	DNS Amplification	https://www.kaggle.com/datasets/aymenabb/ddos-evaluation-dataset-cic-ddos2019	[19]
DATASET 4	LDAP floods	https://www.kaggle.com/datasets/aymenabb/ddos-evaluation-dataset-cic-ddos2019	[19]
DATASET 5	TFTP Amplification	https://www.kaggle.com/datasets/aymenabb/ddos-evaluation-dataset-cic-ddos2019	[19]
DATASET 6	MSSQL Amplification	https://www.kaggle.com/datasets/aymenabb/ddos-evaluation-dataset-cic-ddos2019	[19]

Figure 4: *Confusion Matrix of NETPA-DLA for the CIC-DDoS2019 Dataset*

The timestamp was normalized to a datetime format, and the whole dataset was reindexed using the timestamp as a reference. This helped to preserve the temporal order of the dataset as a time series, which works better with recurrent neural networks and reduces the impact of class imbalance. The time stamp column was dropped before.

The final dataset used for training had 225743 rows, 36 feature columns, and 1 label column, out of which 3 were binary columns (containing 2 unique items) and the rest were non-binary (containing more than 2 unique values) as shown in Table 2.

3.3 Data preprocessing Analysis:

The DDoS attack times lie within the 3 minutes 54 seconds and 4 minutes 16 seconds connection time, with the mode of approximately 3 minutes 57 seconds, while normal connection time ranges from 3 minutes 30 seconds to over 5 minutes Figure 5.

Table 2: Classification of Dataset Features After Data Cleaning

Feature Category	**Feature Names**
Binary Features	Fwd PSH Flags, FIN Flag Count, RST Flag Count, PSH Flag Count, ACK Flag Count, URG Flag Count, Label
Categorical Features	Source IP, Destination IP, Label
Numerical Features	Source Port, Destination Port, Protocol, Total Length of Fwd Packets, Fwd Packet Length Min, Fwd Packet Length Mean, Bwd Packet Length Min, Bwd Packet Length Mean, Flow Bytes/s, Flow Packets/s, Flow IAT Std, Flow IAT Min, Bwd IAT Total, Bwd IAT Mean, Bwd IAT Max, Bwd IAT Min, Bwd Header Length, Bwd Packets/s, Min Packet Length, Packet Length Mean, Down/Up Ratio, Init_Win_bytes_forward, Init_Win_bytes_backward, act_data_pkt_fwd, min_seg_size_forward, Active Std, Active Min, Idle Std

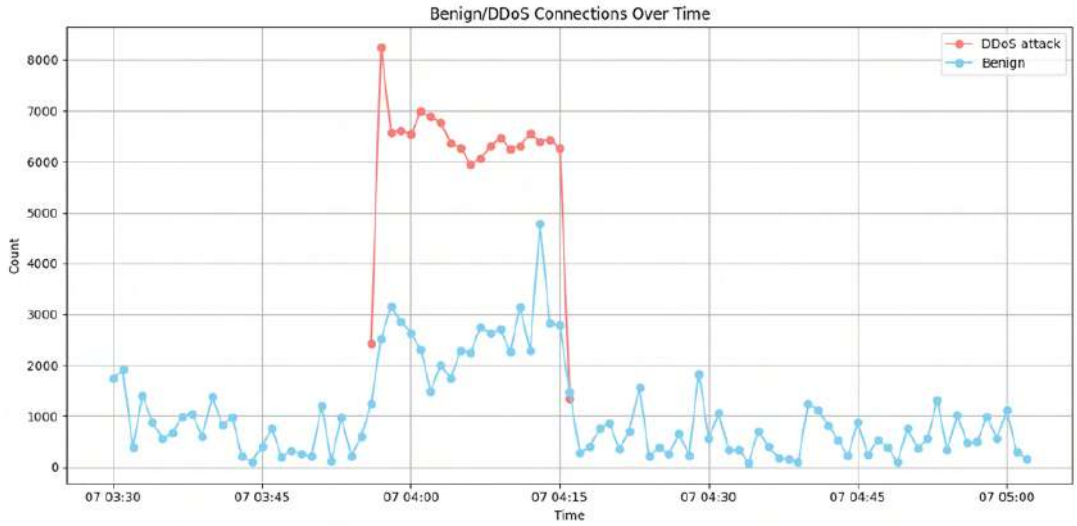


Figure 5: *Comparative Time Analysis*

Figure 6 shows that the dataset was slightly imbalanced, with 128027(56.7%) DDoS data points and 97716 (43.3%) Benign data points. By first identifying the majority and minority classes in the dataset and then randomly eliminating samples from the majority class like dropping the time stamp and selecting only distinct features as shown in (Table 2) until the class distribution is balanced and both classes have roughly equal numbers of samples. Therefore, by switching up the sample dataset's weights, the imbalance issue was resolved by random under sampling.

For most of the binary columns, the DDoS was not indicated, except for columns PCH flag count and ACK flag counts Figure 7.

Three protocols were available: Transmission Control Protocol (TCP) represented with 6, User Datagram Protocol (UDP) represented with 17, and Hop-by-Hop Option (IPv6 Extension Header)-(HOPOPT) represented with 0. HOPOPT had the lowest frequency (0.024%), and TCP had the highest frequency, comprising over 85 % of the dataset Figure 8.

The Figure 9 is showing how the TCP was also predominantly associated with a DDOS attack

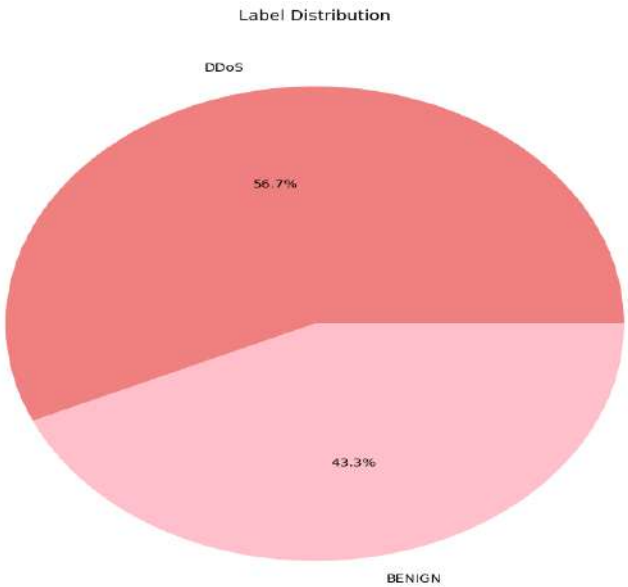


Figure 6: Benign Data Points

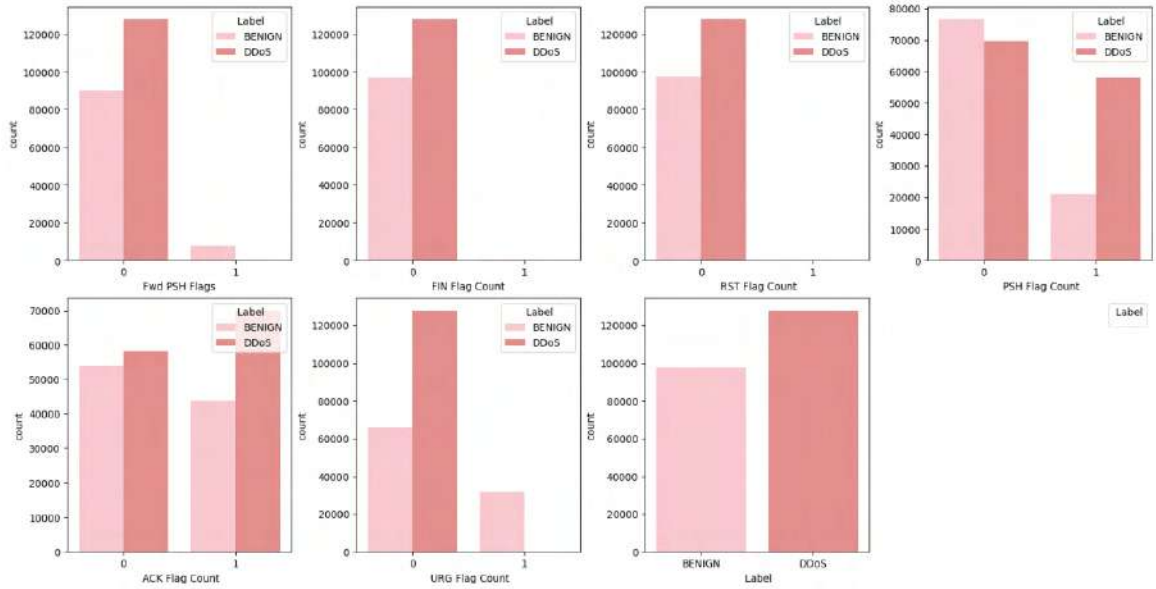


Figure 7: Binary columns representation

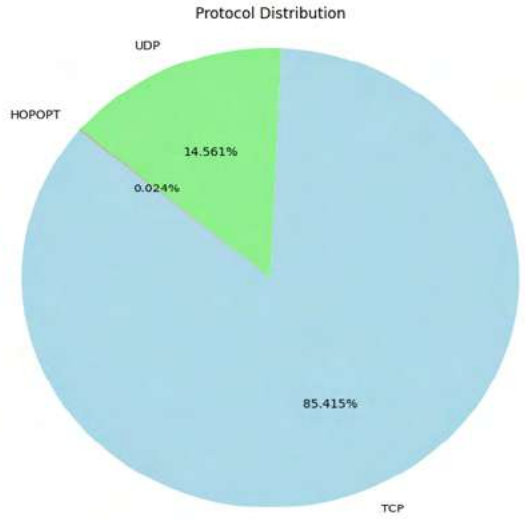


Figure 8: Available Protocols

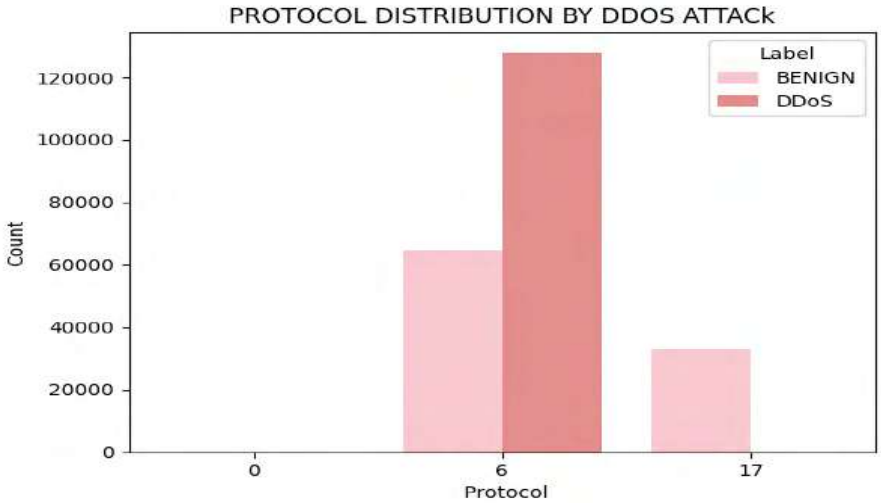


Figure 9: TCP-associated DDoS Attacks

3.4 Architecture for DDoS Detection & Mitigation using NETPA-DLA Approach:

This study evaluated five classifier methods within the NETPA-DLA system for DDoS detection. Input datasets were preprocessed and split using 5-fold TimeSeriesSplit, allocating 80% for training and 20% for testing. Classifier performance was assessed using confusion matrices, with metrics including accuracy, precision, recall, F1-score, ROC-AUC, precision–recall curves, false discovery rate, false negative rate, false positive rate, and negative predictive value. A summary of model algorithm used is presented in Algorithm 1.

3.5 Performance Metrics

To evaluate the performance of the traditional classifiers and ensemble learning on the test data where the true values are known a confusion matrix was used. The performance measures considered in this project include accuracy, recall, precision, and the f1 measure which is determined from the confusion matrix. See Table 3 for the structure of the confusion matrix.

Algorithm 1: Hybrid Prediction Algorithm of NETPA-DLA

Input: (i) $X \in R^{B\ x\ T\ x\ F}$: Batch of sequences (e.g., time-series or flattened image patches) where B = batch size, T = time steps, F = feature dimension

Output: Y: Final feature representation or prediction

1: Initialize Model Components:

2: Instantiate submodules: RNN, DBN, LSTM, Auto encoder, and Transformer.

3: Set necessary hyperparameters: input size, hidden size, latent size, number of layers, etc.

4: Temporal Feature Extraction with RNN:

5: Input X is passed into an RNN block.

6: Output: $H^{(1)} = RNN(X)$

7: Hierarchical Representation Learning with DBN:

8: Pass $H^{(1)}$ through a DBN (modelled with stacked dense layers).

9: Output: $H^{(2)} = DBN(H^{(1)})$

10: Long-Term Dependency Modeling with LSTM:

11: Feed $H^{(2)}$ into the LSTM layer to capture long-term sequence relationships.

12: Output: $H^{(3)} = LSTM(H^{(2)})$

13: Denoising and Dimensionality Reduction with Auto encoder:

14: Compress and reconstruct $H^{(3)}$ using an Auto encoder.

15: Output: $H^{(4)} = Decoder(Encoder(H^{(3)}))$

15: Global Context Encoding with Transformer:

17: Process $H^{(5)}$ with a Transformer encoder for attention-based modeling.

18: Output: $Y = Transformer(H^{(5)})$

19: Return Output:

20: Final representation Y can be passed to task-specific layers (e.g., classification head, decoder, etc.)

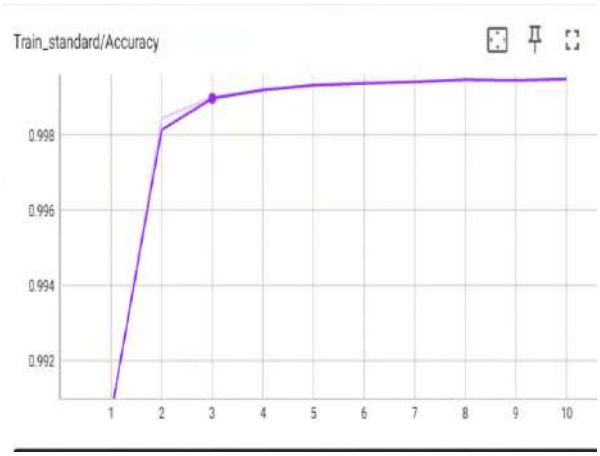
Table 3: THE CONFUSION MATRIX

	PREDICTED POSITIVE	**PREDICTED NEGATIVE**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	True Negative (TN)	False Positive (FP)

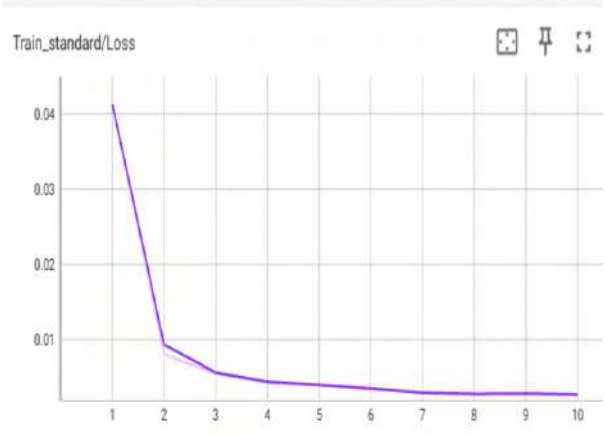
4. RESULTS AND DISCUSSION

Our non-pretrained DBN model proved to be better than the pretrained counterpart for DDoS detection, with an accuracy of 99.72 % and false positive of 37 and false negative of 13 on the validation dataset, with result for all metrics for our LSTM model (Tables 5 and 6) 98%, The least being validation specificity at 98.55 %.

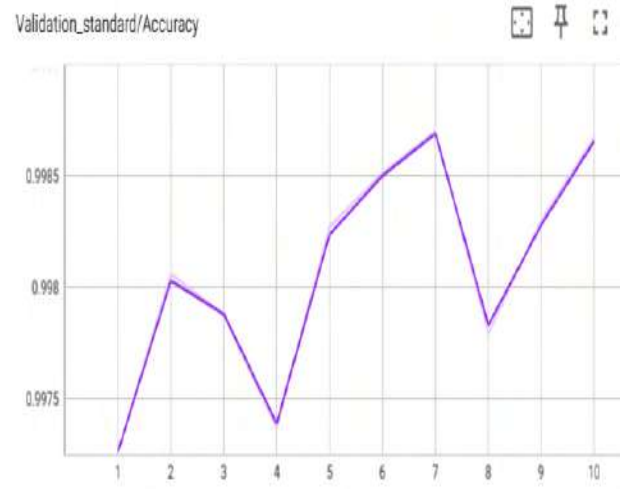
The loss and accuracy plot for the training phase shows smooth progression (a and b), and the validation phase (c and d) shows relatively less smooth plots, which was likely due to the small batch size we used resulting in frequent readjustment as shown in figure 10.



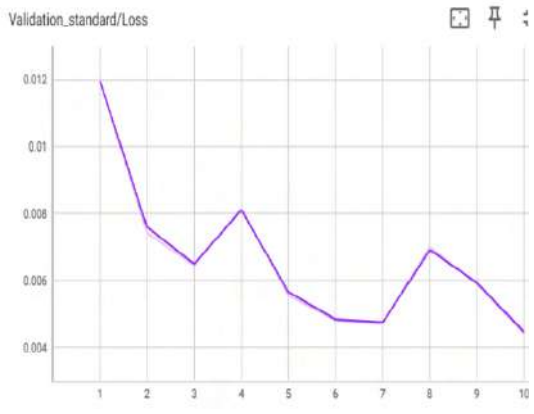
a. : Non-pretrained DBN training accuracy



b: Non-pretrained DBN training loss



c.: Non-pretrained DBN validation accuracy



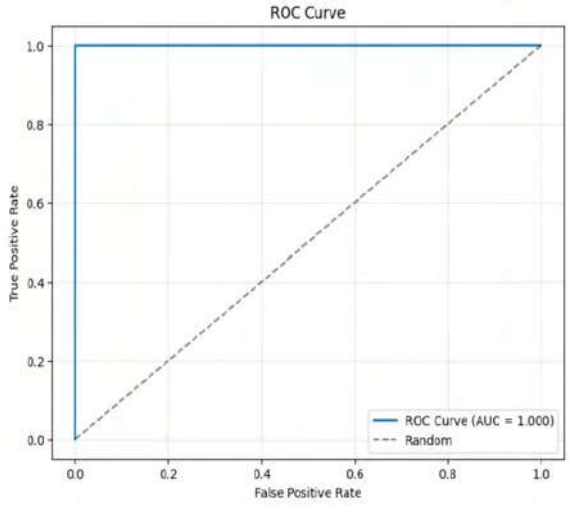
d.:Non-pretrained DBN validation loss

Figure 10: Results of Non-pretrained DBN training and validation.

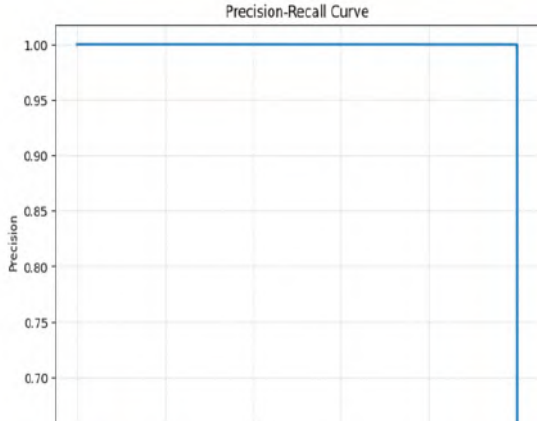
The ROC plot shows a perfect AUC score of approximately 1 in (a-d) in figure 11, which shows that the model understands the data and was not guessing values for any of the two classes. The P-R curve showing the tradeoff between recall and precision shows consistent high score (approximately 1) from the start to the end as shown in (b-c) in the same figure 11 respectively. The right-angle curve shows a high recall and high precision value demonstrating no compromise by imbalanced dataset.

Limitation to model performance was mainly due to the small data sample for training and validation i.e. we used a sample of $120000/150000\,x\,100 = 80\%$ for training and $30000/150000\,x\,100 = 20\%$ for validation as shown in both confusion matrix in (a.) and (b.) in figure 12. But with large high-quality data, our model can learn to generalize and also record better performance metrics.

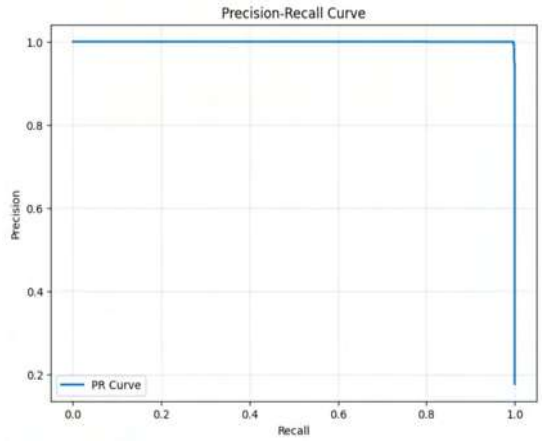
The training and validation duration took about 3 minutes for our best model, but testing out the numerous hyperparameters on the DBN model took several hours of continuous training and validation.



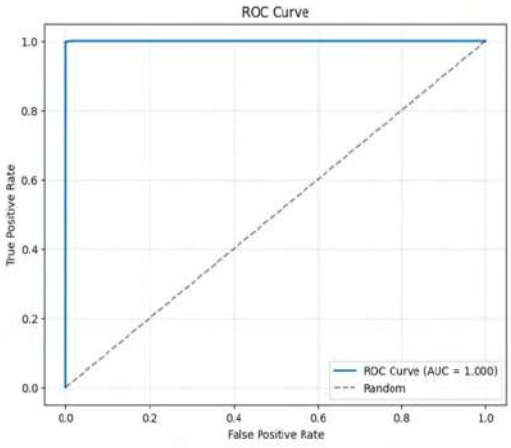
a.: Non-pretrained DBN training ROC curve



b.: Non-pretrained DBN training PR curve

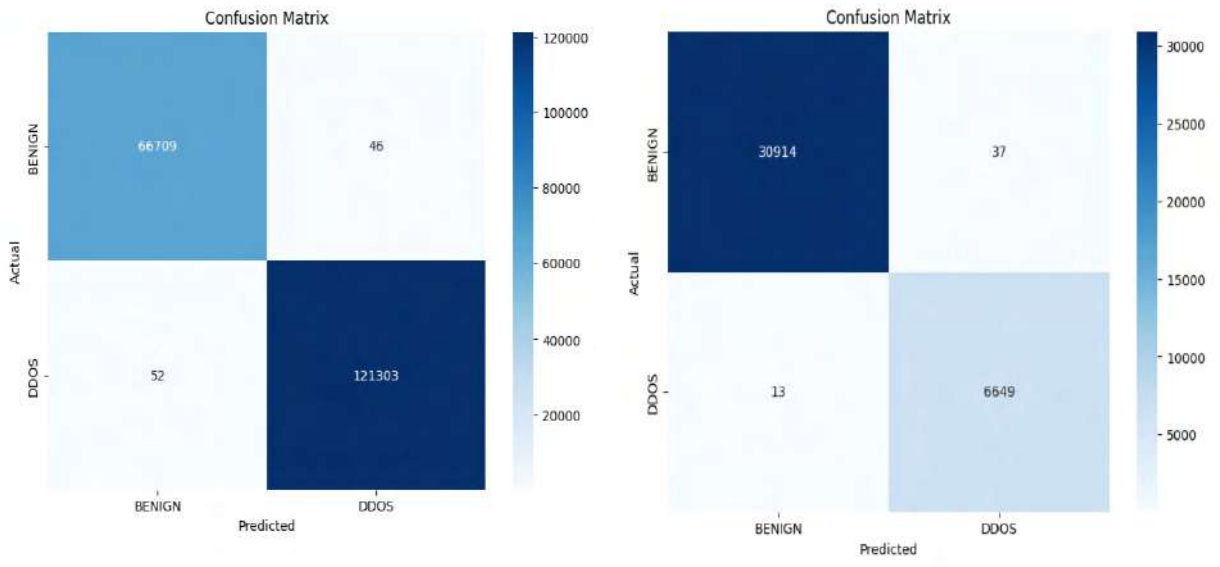


c.: Non-pretrained DBN validation PR curve



d.: Non-pretrained DBN validation ROC curve

Figure 11: (a-d) ROC of 70%:30% and (b-c) PR curve of 70%:30%.



(a.) Training CM (b.) Validation CM

Figure 12: Non-pretrained DBN training and validation confusion matrix

4.1 Results:

The model was tuned with training carried out using the Cartesian product of hidden dimensions=[[64, 32], [128, 64], [128, 64, 32]], dropout=[0.2, 0.3], reconstruction weight=[0.1, 0.3], pretrain epochs=[5, 10], finetune epochs=[10, 15], pretrain learning rate=[0.001, 0.01], finetune learning rate=[0.0001, 0.001], batch normalization=[True, False] giving a total of 384 trials. The binary cross-entropy is used to measure classification loss. The pre-trained DBN model was unsupervised for 5–10 epochs before fine-tuning, using a batch size of 128, dropout 0.2, hidden layers [64, 32], batch normalization, and a 0.001 learning rate. Performance was evaluated via accuracy, loss, precision, recall, specificity, F1-score, ROC, and precision–recall curves, with metrics logged per epoch.

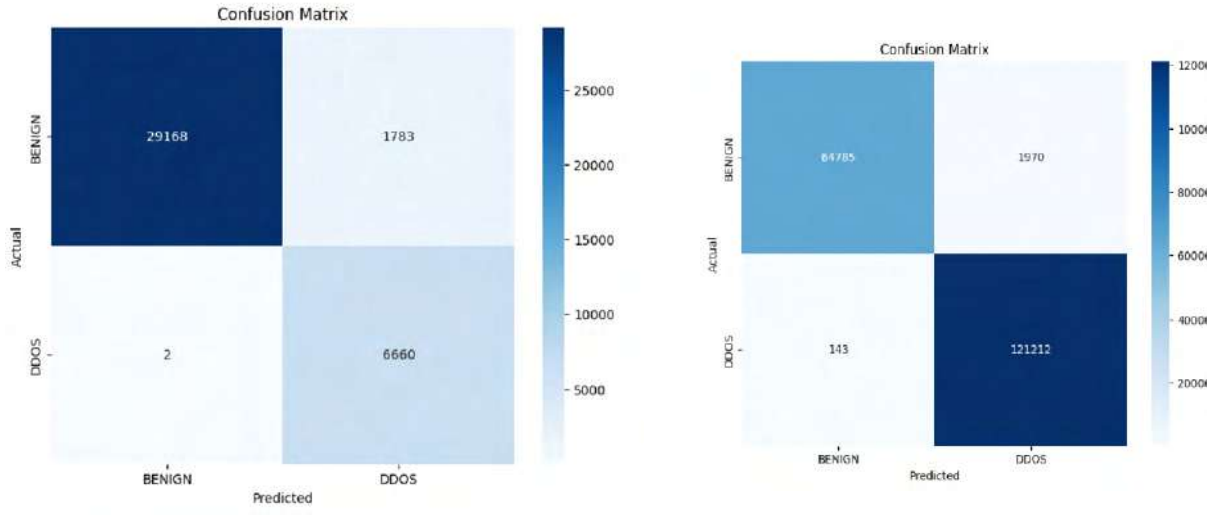
The pretrained DBN model performed below other models when tried for DDOS detection, with an accuracy of 95.25% and false positive of 1783, and false negative of 2 on the validation dataset shown in figure 13, with the result for all the metrics performing below other architecture in (table 5 and 6) repectively and the precision score being the lowest at 78.88%.

Table 5: Training Data

Model	Accuracy	loss	F1 score	precision	Recall	specificity
LSTM	99.98	0.0012	99.98	99.98	99.98	99.97
Pretrain DBN	98.87	0.1334	99.14	98.40	99.88	97.05
DBN	99.95	0.0027	99.96	99.97	99.96	99.94
RNN	98.16	0.4482	98.61	97.27	99.99	94.90
Autoencoder	99.97	0.0021	99.98	99.98	99.98	99.96
Transformer	99.97	0.0022	99.96	99.97	99.97	99.95

Table 6: Validation Data

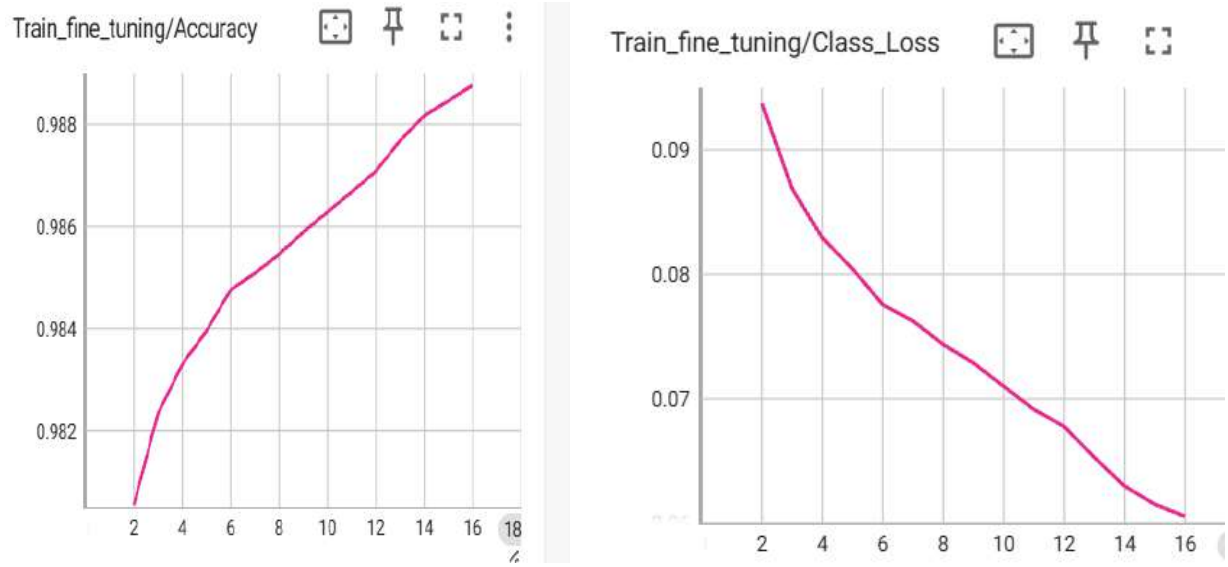
Model	Accuracy	loss	F1 score	precision	Recall	specificity
LSTM	99.65	0.0131	99.02	98.48	99.58	99.67
Pretrain DBN	95.25	0.3552	88.18	78.88	99.97	94.24
DBN	99.76	0.0073	99.31	98.81	99.82	99.74
RNN	82.08	0.6265	66.64	49.97	100	78.45
Autoencoder	99.86	0.0064	99.62	99.40	99.83	99.87
Transformer	99.99	0.0012	99.96	99.94	99.99	99.99



(a) Validation CM (b.) Training CM

Figure 13: Pretrained DBN validation and training confusion matrix

The loss and accuracy plot for the training phase shows a smooth progression (a-b) figure 14, but the validation phase in (c-d) in the same figure 14 shows a rapidly changing value, which was likely due to the small batch size we used as a result of limited available data compared to other studies on the same task.



a.: Pretrained DBN training accuracy b.: Pretrained DBN training loss

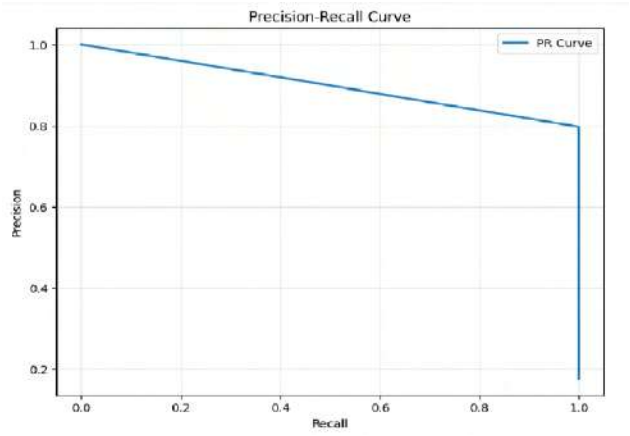
d.: Pretrained DBN validation loss

c.: Pretrained DBN validation accuracy

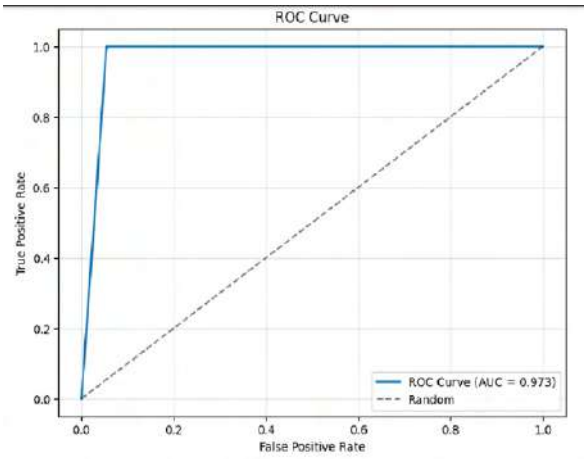
Figure 14: Pretrained DBN validation and training (accuracy and loss)

The ROC plot shows an AUC score of approximately 0.973 in (b-c) in the figure 15, which shows that the model has better understanding of the data and was not guessing values for any of the two classes, but the reduced score compared to other models shows there was some level of influence from the larger class. The P-R curve showing the tradeoff between recall and precision shows a relatively drifting score as we approach score higher scores in (a-d) in the same figure 15.

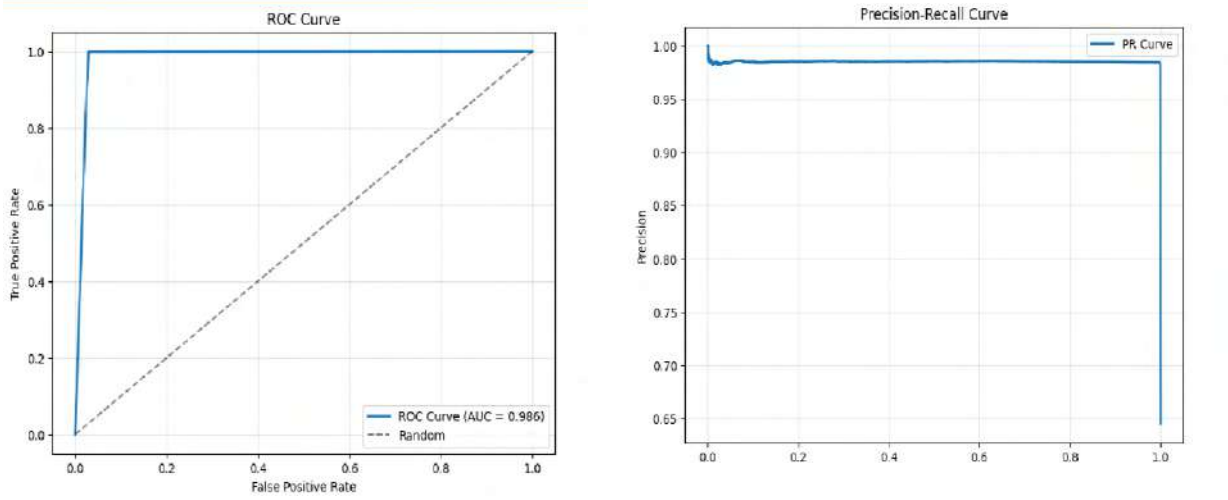
The angle of curve in P-R was higher than 90 degrees with the precision suffering the most.



a.: Pretrained DBN validation PR curve



b.:Pretrained DBN validation ROC curve



c.: Pretrained DBN training ROC curve d.: Pretrained DBN training PR curve

Figure 15: Pretrained DBN validation and training ROC and PR curves respectively.

The data sample for training and validation were smaller than what was used in other studies. But with large high-quality data, our model can learn to generalize better. This pretraining approach, which was focused on unsupervised training of the RBMs, did not improve the performance as anticipated, and the result shows signs of underfitting and took too long to converge. The precision-recall curve shows an analysis of the weight and bias in the layers shows a continuous improvement in the weight and bias of DBN layers, but the RBM layer, which is the building block, was stagnant throughout the steps, which had a negative impact on the training.

4.2 Discussions:

For the best LSTM model, the batch size was set to 64, meaning that the model sees 64 samples at a time and it is shuffled to prevent overfitting, the dropout was 0.2, the hidden dimension was set to 32 and the number of layers was 2. Interchanging between a learning rate of 0.001 and 0.0001, the learning rate of 0.001 shows a better progression. Other LSTM models with good score were eliminated because it's either the model learnt too fast and overfit easily or the change in accuracy, which was the primary metric of comparison, was erratic. The required metrics were train and validation accuracy, train and validation loss, train and validation recall, train and validation precision, train and validation specificity and the train and validation F1 Score. The ROC curve and the precision–recall (P-R) curve was to determine the actual performance in both attack classes (Benign and DDOS). The customized training and validation function allowed to save the metrics as a step-based data to the tensor board object, allowing to view all these metrics and plots in each epoch.

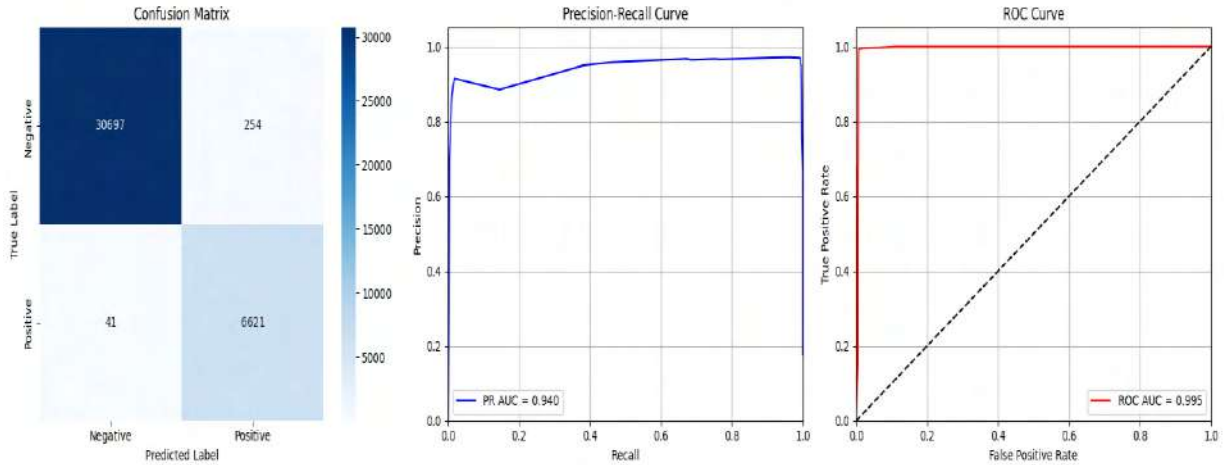


Figure 16: LSTM Confusion Matrix, training and validation curve

The training phase shows a smooth progression from the confusion matrix in figure 16, along with the precision-recall curve and ROC curve placed side-by-side, but the loss and accuracy plot for the validation phase in figure 17, shows a rapidly changing value, which was likely due to the small batch size we used. The choice of batch sizes was influenced by the total dataset available.

The ROC plot shows a perfect AUC score of approximately 1, which shows our model understand the data and was not guessing values for any of the two classes. The P-R curve showing the tradeoff between recall and precision shows a consistently high score (approximately 1) from the start to the end. The right-angle curve shows a high recall and high precision value, demonstrating no compromise by the imbalanced dataset. The training and validation duration took about 4 minutes for our best model as captured in figure 18, but testing out the numerous hyperparameters on the LSTM model, which was a complex task, took many hours of continuous training and validation

.

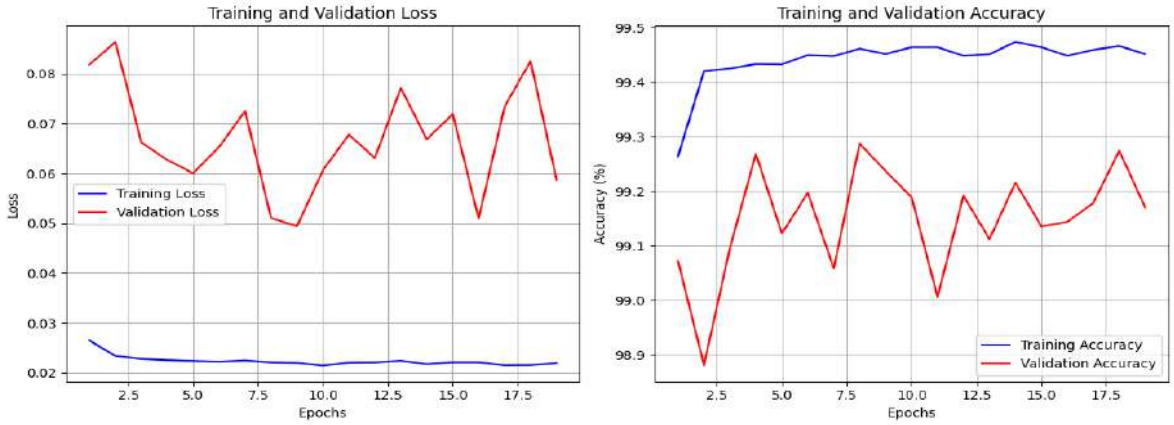
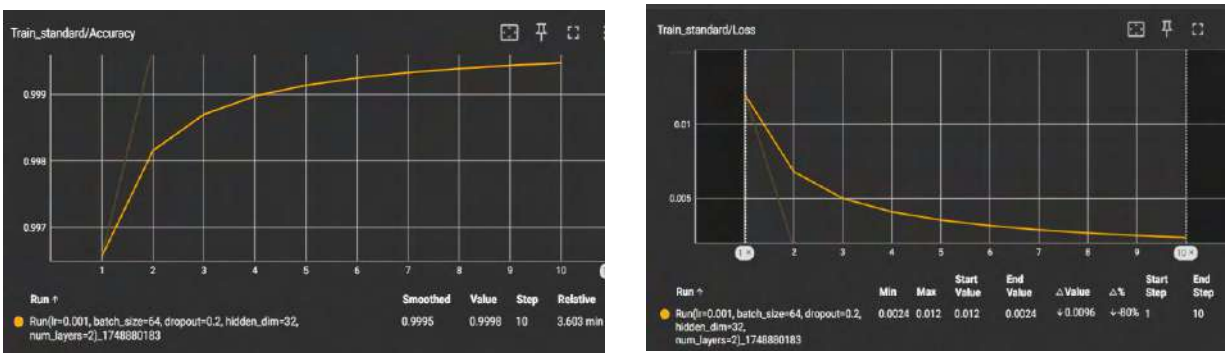
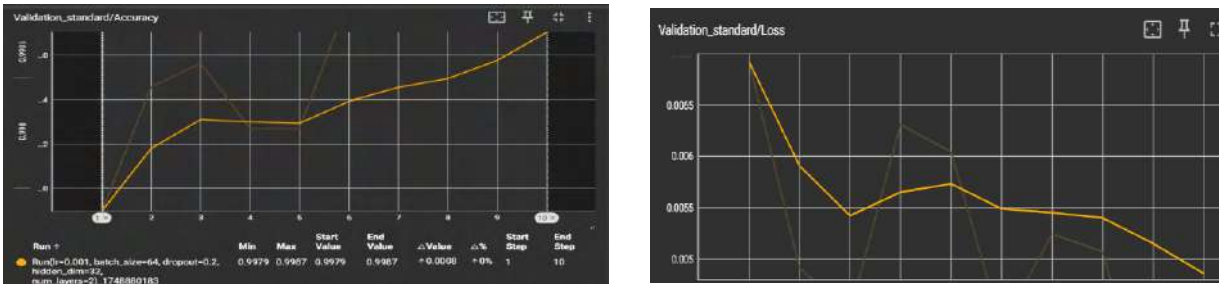


Figure 17: LSTM Validation Loss and accuracy



(a) (b)



(c) (d)

Figure 18: (a-c) Accuracy of 80%:20% and (b-d) Loss of 80%:20

Transformers employ self-attention to determine the relative relevance of various input sequence elements, in contrast to conventional recurrent neural networks (RNNs), which process sequences sequentially. This enables them to more successfully capture contextual relationships and long-range interdependence. As shown in the results of the Transformer model, which proved to be one of the best models for DDOS detection, it consistently outperformed other models. After running multiple validations, the transformer was still the best with an accuracy of 99.99 % and false positive of 9, and false negative rate of 1 on the validation dataset, with results for all metrics for the LSTM model and Autoencoder (Tables 5 and 6) above 99%. The LSTM, on the hand, was not consistent and changed position with DBN frequently.

During the experimental validation study of the NETPA-DLA methods, the results show that the NETPA-DLA model efficiently classified and recognized all class labels. Figure 19 stated the classifier outcome of the NETPA-DLA approach at 80:20 percent. Figures 19a and 19c depict the accuracy study of both the model training and validation. The figure shows that the NETPA-DLA method achieved increased accuracy performance over growing epoch counts as recorded in Figure 20.

The progressive validation training accuracy over training accuracy shows that the NETPA-DLA model learns proficiently from the dataset. Additionally, figures 19b and 19d show the loss study of the NETPA-DLA technique. These results indicate that the NETPA-DLA approach achieves adjacent training and validation loss outcomes. It is identified that the NETPA-DLA approach studies the test datasets well.

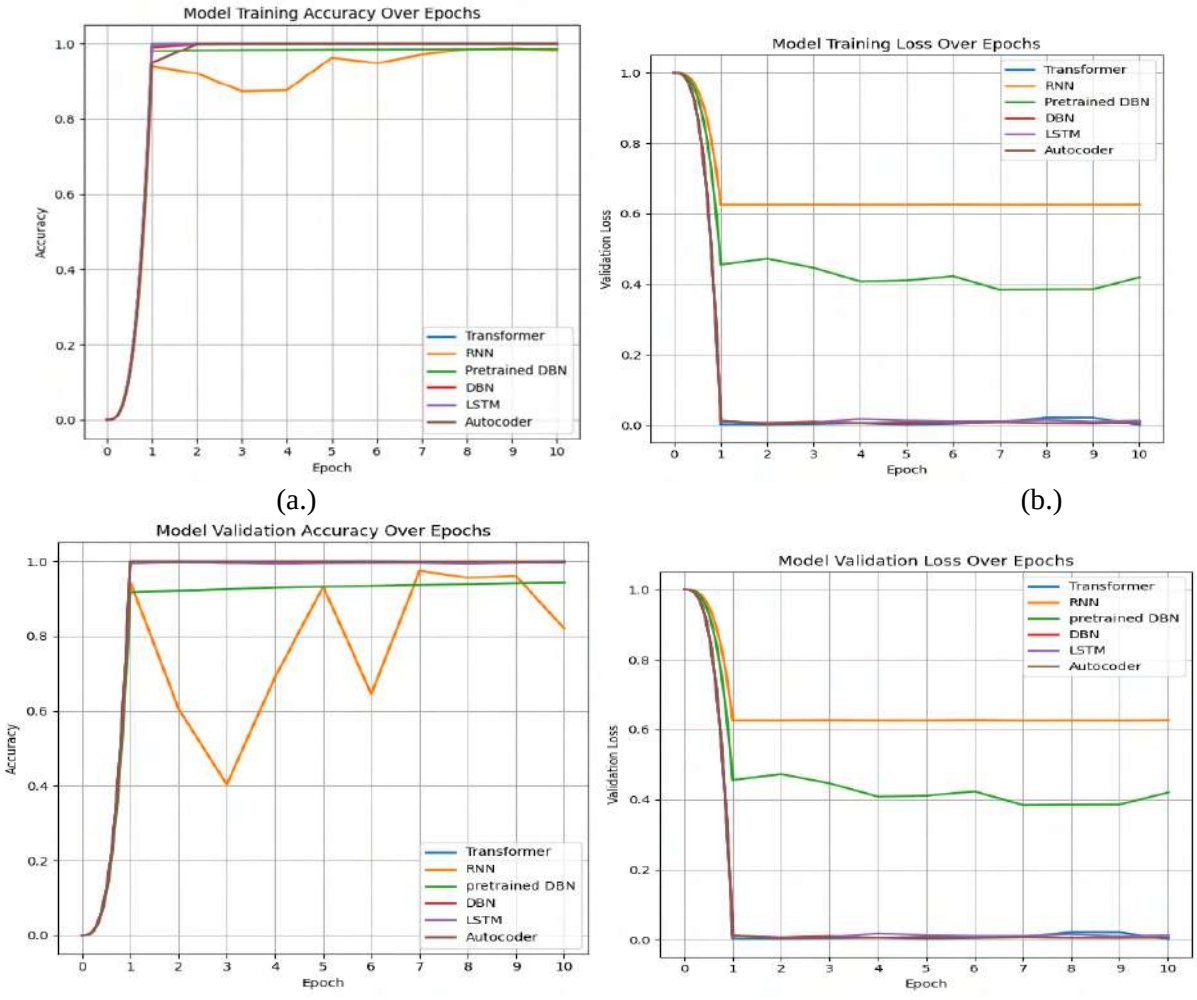


Figure 19: The plots showing the comparison between the accuracy and loss for both training and validation over epochs.

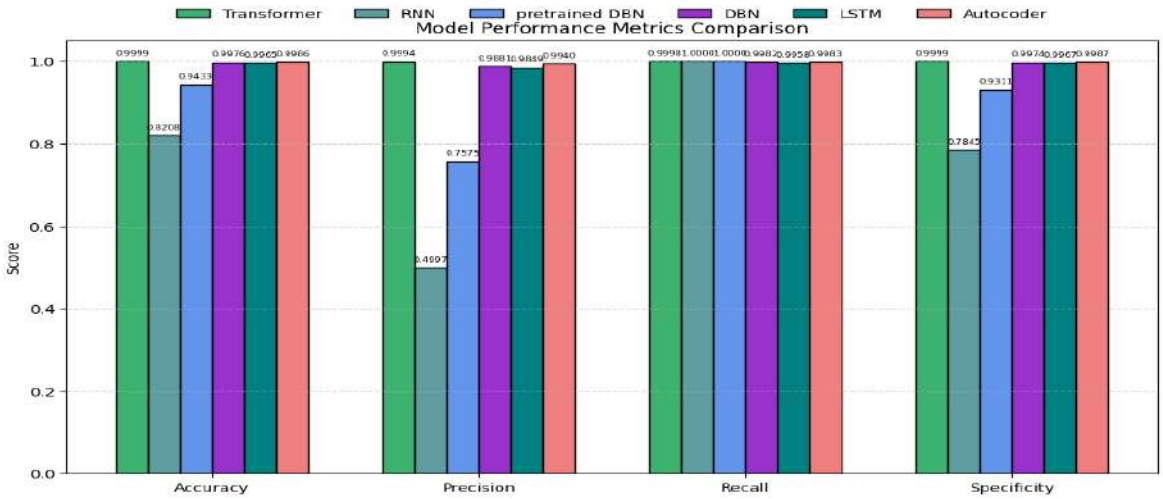


Figure 20: Graphical representations of Evaluation Metrics for Deep Learning Algorithms.

Figure 20 depicts complete attack recognition results of the NETPA-DLA method under 80% TRAIN and 20% TEST. The results show that the NETPA-DLA method has properly identified and classified six classes as against the ensemble, which were identified and classified against 12 different classes. With 80% TRAP, the NETPA-DLA methodology offers an accuracy of 99.98% in LSTM.

The weighted voting ensemble learning model was used as ADLRT_20-20-20-20-20, ADLRT_25-16-16-8-33, and ADLRT_25-15-10-5-45, where ADLRT means Autoencoder weights 20, 25, 25; DBN 20,16,15; LSTM 20, 16, 10; RNN 20, 8, 5; and Transformer 20, 33, 45 respectively across three different runs in this study, and results were obtained from each of the algorithms were recorded alongside the results of the deep learning model as shown in Figure 21 below. In the ensemble, all weights were normalized to 1.

Each algorithm considered in section 2.5 performance measures and generated an output result that include Accuracy, Precision, Recall, Specificity, Sensitivity, Cohen's KAPPA, F1 Score, ROC_AUC, TP, TN, FPR, and FNR. The accuracy levels allow us to identify the ensemble and deep learning classifiers that perform the best at identifying DDoS attacks. Transformer had the highest accuracy level of 0.9998, closely followed by Autoencoder, which had an accuracy level of 0.9986, and ensemble weighted voting at 0.9984, while the RNN obtained a perfect score of 1.0000 for both Recall and Sensitivity across the three relative weights for each of the models. i.e., ADLRT_20-20-20-20-20, ADLRT_25-16-16-8-33, and ADLRT_25-15-10-5-45.

Additionally, Figure 21 gives a clear graphical representation of the Evaluation Metrics for Deep Learning Algorithms and ensemble learning. The Bar graph represents each run of the ensemble learning weighted voting and the deep learning algorithms, and the results they produced individually.

Weights are important parameters in deep learning that control how strongly neurons in a network link to one another. To reduce mistakes and increase accuracy, these weights and biases are modified during training as shown in the dashboard in Figure 18. The network basically determines the ideal weight values to appropriately map input data to the appropriate output classes. The accuracy across board shows that the relative weights in ADLRT_25-16-16-8-33 and ADLRT_25-15-10-5-45 produce better accuracy compared to ADLRT_20-20-20-20-20, which has equal values. However, in the comparison of the weight versus the accuracy, the ADLRT_20-20-20-20-20produced a more stable distribution as seen in figure 22.

The reason for the comparison of F1 Score and Cohen Kappa Score is that the two help to exclude class imbalance. They are good metrics for checking data imbalance.

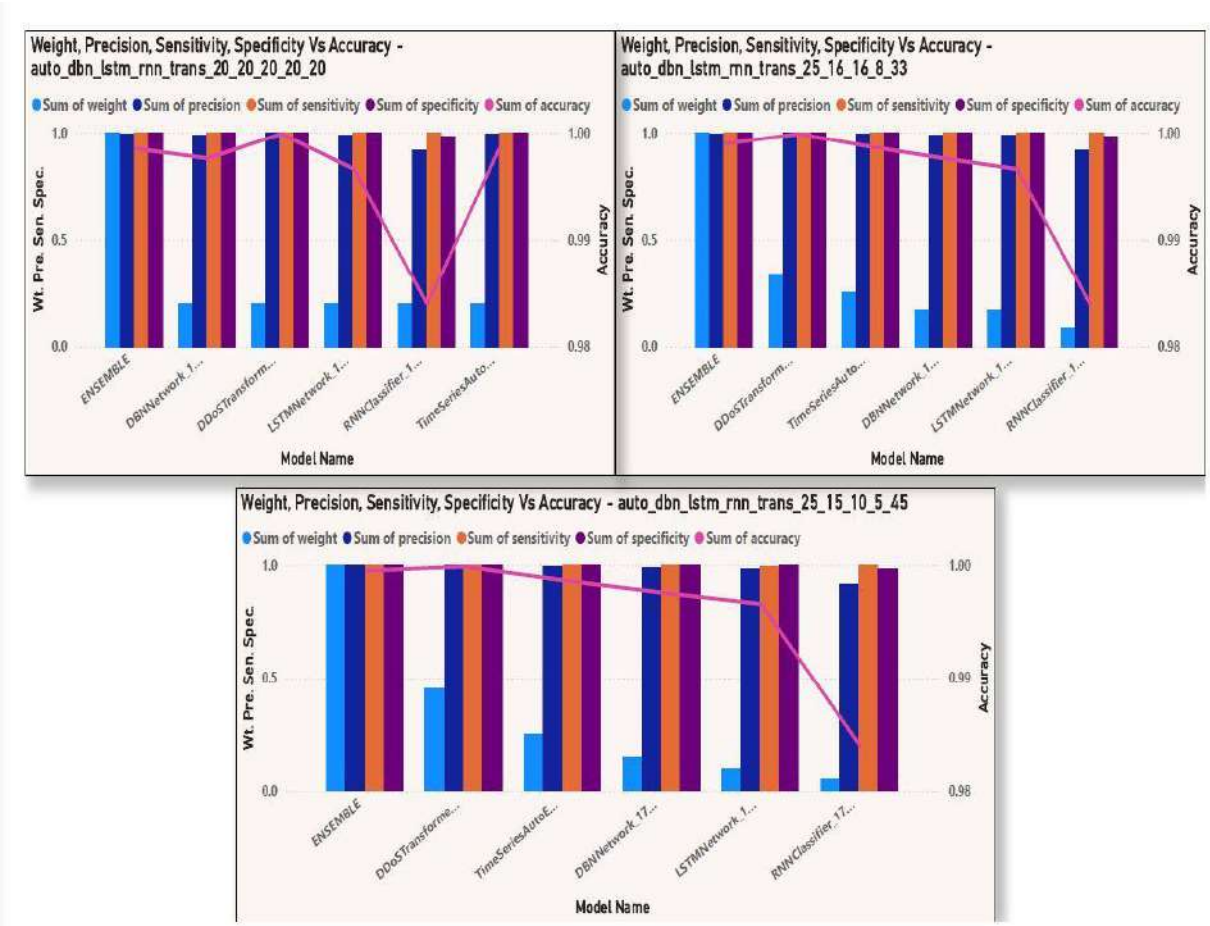


Figure 21: Comparative Ensemble Analysis of the NETPA-DLA technique with recent models.

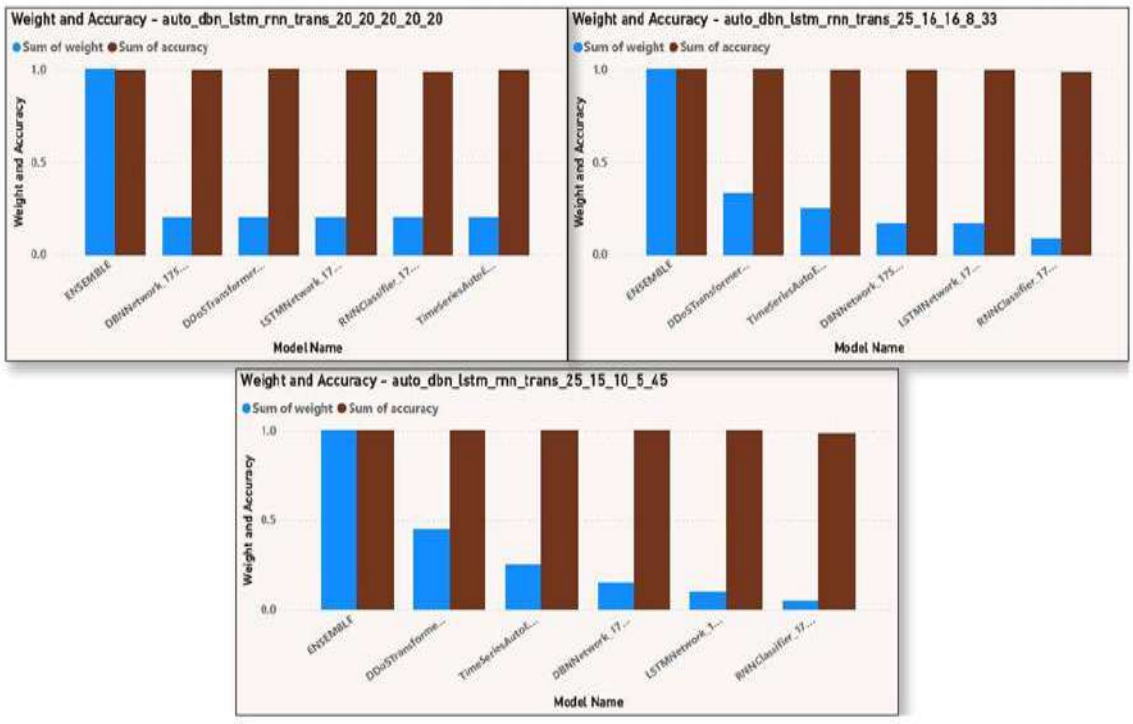


Figure 22: *Ensemble Evaluation of the NETPA-DLA approach with accuracy of deep models.*

5. Conclusion

DDoS assaults are becoming more varied in form and volume to exhaust the target resources across whole network systems. Therefore, we deployed an automatic detection and mitigation system that implements a policy of protection from the Application – Systems - Network. Ensuring security at all levels, as well as a monitoring system peculiar to each organization. The ML module, built using the Flask web framework, processes and classifies the incoming flows as benign or malicious based on time-accumulated data over a defined window size. The integrated platforms through the Faucet SDN system, facilitate real-time monitoring, visualization, and data scraping for performance analysis. This integration of CICFlowMeter in ML detection, and SDN control provides an adaptive, automated, and efficient framework for DDoS mitigation in software-defined networks. The comparison also showed that weights are important parameters in deep learning, it reduces mistakes while increasing accuracy, this directly influences the control of neurons in a network link to one another. In future work, mitigation decisions, such as blocking or rate limiting, will be implemented dynamically via configuration files with thresholds adjustable according to the confidence score of predictions. Attacks with confidence levels above 0.7 are blocked automatically, while those between 0.5 and 0.69 are rate-limited.

References

[1] Almulhim, A.I. and T. Yigitcanlar, Understanding smart governance of sustainable cities: A review and multidimensional framework. Smart Cities, 2025. 8(4): p. 113.

[2] Caputo, F., et al., Rethinking the role of technology for citizens' engagement and sustainable development in smart cities. Sustainability, 2023. 15(13): p. 10400.

[3] Azeez, N., et al., Evaluation of a flexible column-based access control security model for medical-based information. Journal of Computer Science and Its Application, 2015. 22(1): p. 14-25.

[4] Shalaginov, A., et al. Modern cybercrime investigation: technological advancement of smart devices and legal aspects of corresponding digital transformation. in 2020 IEEE International Conference on Big Data (Big Data). 2020. IEEE.

[5] Tok, Y.C. and S. Chattopadhyay, Identifying threats, cybercrime and digital forensic opportunities in Smart City Infrastructure via threat modeling. Forensic Science International: Digital Investigation, 2023. 45: p. 301540.

[6] Kim, M., Supervised learning - based DDoS attacks detection: Tuning hyperparameters. ETRI Journal, 2019. 41(5): p. 560-573.

[7] Chou, E., F. Tramer, and G. Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. in 2020 IEEE Security and Privacy Workshops (SPW). 2020. IEEE.

[8] Kurt, M.N., Y. Yılmaz, and X. Wang, Distributed quickest detection of cyber-attacks in smart grid. IEEE Transactions on Information Forensics and Security, 2018. 13(8): p. 2015-2030.

[9] Katti, A., A Brief Visit to the Landscape of Cloud DDoS Attacks, in Recent Advances in Computer Based Systems, Processes and Applications. 2020, CRC Press. p. 15-26.

[10] George, A.S., T. Baskar, and P.B. Srikaanth, Cyber threats to critical infrastructure: assessing vulnerabilities across key sectors. Partners Universal International Innovation Journal, 2024. 2(1): p. 51-75.

[11] Ayofe, A.N., A.R. Ajetola, and A.S. Oyewole, Assessment of existing gap between industrial IT skill requirements and computer science curriculum in tertiary institutions. The Pacific Journal of Science and Technology, 2009. 10(2): p. 326-336.

[12] Azeez, N.A. and C. Van Der Vyver. Digital education: assessment of e-learning and m-learning adoption in tertiary institutions in South Africa. in 2018 IEEE Conference on e-Learning, e-Management and e-Services (IC3e). 2018. IEEE.

[13] Li, Q., et al. DDoS attacks detection using machine learning algorithms. in International Forum on Digital TV and Wireless Multimedia Communications. 2018. Springer.

[14] Azeez, N.A. and I.M. Venter, Towards ensuring scalability, interoperability and efficient access control in a multi-domain grid-based environment. SAIEE Africa Research Journal, 2013. 104(2): p. 54-68.

[15] Mishra, D., et al., Light gradient boosting machine with optimized hyperparameters for identification of malicious access in IoT network. Digital Communications and Networks, 2023. 9(1): p. 125-137.

[16] Gupta, B.B. and A. Dahiya, Distributed Denial of Service (DDoS) Attacks: Classification, Attacks, Challenges and Countermeasures. 2021: CRC press.

[17] Salim, M.M., S. Rathore, and J.H. Park, Distributed denial of service attacks and its defenses in IoT: A survey. Journal of Supercomputing, 2020. 76(7).

[18] Kaur Chahal, J., A. Bhandari, and S. Behal, Distributed denial of service attacks: a threat or challenge. New Review of Information Networking, 2019. 24(1): p. 31-103.

[19] Sharafaldin, I., et al. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. in 2019 international carnahan conference on security technology (ICCST). 2019. IEEE.

[20] Kadri, M.R., et al., Survey and classification of Dos and DDos attack detection and validation approaches for IoT environments. Internet of Things, 2024. 25: p. 101021.
[21] Kwon, H.-Y., T. Kim, and M.-K. Lee, Advanced intrusion detection combining signature-based and behavior-based detection methods. Electronics, 2022. 11(6): p. 867.
[22] Díaz-Verdejo, J., et al., On the detection capabilities of signature-based intrusion detection systems in the context of web attacks. Applied Sciences, 2022. 12(2): p. 852.
[23] Moustafa, N., J. Hu, and J. Slay, A holistic review of network anomaly detection systems: A comprehensive survey. Journal of Network and Computer Applications, 2019. 128: p. 33-55.
[24] Chang, V., et al., A survey on intrusion detection systems for fog and cloud computing. Future Internet, 2022. 14(3): p. 89.
[25] Aslan, Ö., et al., A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics, 2023. 12(6): p. 1333.
[26] Azeez, N. and H. Iliyas, Implementation of a 4-tier cloud-based architecture for collaborative health care delivery. Nigerian Journal of Technological Development, 2016. 13(1): p. 17-25.
[27] Pandian, S., A comprehensive guide on hyperparameter tuning and its techniques. Analytics Vidhya, 2022.
[28] AnalyticsVidhya, Techniques to Solve Imbalanced Classes in Machine Learning. 2023.
[29] Aktar, S. and A.Y. Nur, Towards DDoS attack detection using deep learning approach. Computers & Security, 2023. 129: p. 103251.
[30] Rani, S.J., et al., Detection of DDoS attacks in D2D communications using machine learning approach. Computer Communications, 2023. 198: p. 32-51.
[31] Mahadik, S.S., P.M. Pawar, and R. Muthalagu, Edge-HetIoT defense against DDoS attack using learning techniques. Computers & Security, 2023. 132: p. 103347.
[32] Sarıkaya, A., B.G. Kılıç, and M. Demirci, RAIDS: Robust autoencoder-based intrusion detection system model against adversarial attacks. Computers & Security, 2023. 135: p. 103483.
[33] Al-Shareeda, M.A., S. Manickam, and M.A. Saare, DDoS attacks detection using machine learning and deep learning techniques: analysis and comparison. Bulletin of Electrical Engineering and Informatics, 2023. 12(2): p. 930-939.
[34] Diaba, S.Y. and M. Elmusrati, Proposed algorithm for smart grid DDoS detection based on deep learning. Neural Networks, 2023. 159: p. 175-184.
[35] Mittal, M., K. Kumar, and S. Behal, DL-2P-DDoSADF: Deep learning-based two-phase DDoS attack detection framework. Journal of Information Security and Applications, 2023. 78: p. 103609.
[36] Ahmad, I., Z. Wan, and A. Ahmad, A big data analytics for DDOS attack detection using optimized ensemble framework in Internet of Things. Internet of Things, 2023. 23: p. 100825.
[37] Hossain, M.A. and M.S. Islam, Enhancing DDoS attack detection with hybrid feature selection and ensemble-based classifier: A promising solution for robust cybersecurity. Measurement: Sensors, 2024. 32: p. 101037.
[38] Pandithurai, O., et al., DDoS attack prediction using a honey badger optimization algorithm based feature selection and Bi-LSTM in cloud environment. Expert Systems with Applications, 2024. 241: p. 122544.
[39] Nalayini, C., et al., A novel dual optimized IDS to detect DDoS attack in SDN using hyper tuned RFE and deep grid network. Cyber Security and Applications, 2024. 2: p. 100042.
[40] Mishra, A.K., S. Paliwal, and G. Srivastava, Anomaly detection using deep convolutional generative adversarial networks in the internet of things. ISA transactions, 2024. 145: p. 493-504.
[41] Ioannou, I., et al., Gemlids-miot: A green effective machine learning intrusion detection system based on federated learning for medical iot network security hardening. Computer Communications, 2024. 218: p. 209-239.
[42] Azimjonov, J. and T. Kim, Designing accurate lightweight intrusion detection systems for IoT networks using fine-tuned linear SVM and feature selectors. Computers & Security, 2024. 137: p. 103598.
[43] Hnamte, V., et al., DDoS attack detection and mitigation using deep neural network in SDN environment. Computers & Security, 2024. 138: p. 103661.
[44] Zhao, Z., et al., DDoS family: A novel perspective for massive types of DDoS attacks. Computers & Security, 2024. 138: p. 103663.
[45] Madhava, R., Network packet analyzer. 2024.
[46] Rickert, C.A., M. Henkel, and O. Lieleg, An efficiency-driven, correlation-based feature elimination strategy for small datasets. APL Machine Learning, 2023. 1(1).