Vol. 06, No. 02, pp. 427 –444 (2025)
ISSN: 2708-0757

JOURNAL OF APPLIED SCIENCE AND TECHNOLOGY TRENDS

www.jastt.org

Hybrid Feature Selection and Deep Neural Architectures for Real-Time Distributed Denial of Service Detection in Cybersecurity Systems

Raghupathi Manthena[1], Radhakrishna Vangipuram[2*]

[1]*Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad, India, mraghu30@gmail.com*
[2]*Department of Information Technology, VNR Vignana Jyothi Institute of Engineering and Technology Hyderabad, India, radhakrishna_v@vnrvjiet.in*

**Correspondence: radhakrishna_v@vnrvjiet.in*

Abstract

In today's digital age, timely and accurate detection is essential to maintain online service availability. Timely detection of network attacks is more crucial in network security. Adding effective feature selection minimizes computational requirements on security systems and increases accuracy, leading to more efficient mitigation. Hypothesis-driven feature selection and dense neural architecture address the dimensionality problem while preserving DDoS attack detection efficiency. The proposed method selects the significant features from the high-dimensional network traffic by integrating Z-tests and chi-square tests and detects the DDoS attacks using the DENSE Multi-Layer neural network (DNN) model. This work was evaluated on the benchmark publicly available datasets NSL-KDD, BoT_IoT, CICIDS2017, CICIDS2018, and CICDDoS2019, while it reduced an 80.13±2.38% feature space. The highlight of this work is predicting the network traffic in less than a second. The model performed well in generalizing, with detection rates of 99.34% (CICDDoS2017), 100% (CICIDS2018), and 99.69% (CICDDoS2019). However, the model performed moderately well on NSL-KDD (75.51%) and BOT-IoT (89.86%), showing that deep learning models can be influenced by the variety of datasets and how the features are organized. The proposed model outperforms the state-of-the-art comparison with existing works in terms of detection rate.

Keywords: *Low-rate and High-rate DDoS Detection, Intrusion Detection System, Hybrid Feature Selection, Deep Neural Architectures DENSE MLP, CICDDOS2019 Dataset, CICIDS Datasets, NSL-KDD, BOT-IoT.*

Received: July 10th, 2025 / Revised: November 25th, 2025/ Accepted: December 05th, 2025 / Online: December 11th, 2025

I. INTRODUCTION

Recent technological advancements have driven significant progress across various domains, including healthcare, industrial automation, and communications. The widespread digitization of critical service sectors such as insurance, energy, telecommunications, and banking has increasingly compelled individuals to depend on online and mobile banking solutions. This surge in digital adoption has fueled the expansion of various financial and transactional services, strengthening the interconnectivity and reliance on digital infrastructure more than ever before and parallel to this cyber threats also raised. Among the many types of cyber threats, Distributed Denial-of-Service (DDoS) attacks are commonly identified as one of the most severe due to their capability of interrupting service availability and bringing about extensive network outages. DDoS attacks affect reputable organizations such as Amazon, Google Cloud, GitHub, Dyn, among others [1-3]. All this information indicates the impact of the DDoS attacks in real time.

Intrusion detection system (IDS) is a security tool, keeps monitoring network traffic to detect any suspicious or malicious activity. Its main role is to identify and classify network traffic as either normal or potentially harmful. Traditional IDS techniques are not suitable for detecting modern types of DDoS attacks [4]. It indicates that an AI-enabled IDS system is needed to detect DDoS attacks [5]. The researchers [6] and [7] identified that AI-enabled IDS systems such as machine learning (ML)-based IDS and deep learning (DL)-based IDS, showed ineffective performance for the DDoS attacks detection due to the high dimensionality and nonlinearity in the data. Compared to the ML-IDS, the DL-IDS consumes more resources in terms of memory and time [8]. The ML-IDS has become a useful tool to build intelligent IDSs that can detect harmful activities [8-10].

doi: 10.38094/jastt62303

But for ML-based IDSs to work well, they need good input data. Network traffic data has too many dimensions, noise, and unimportant features. Such characteristics can negatively affect the performance of ML classifiers [11-13]. These problems can be mitigated by selecting the significant features from the data. Feature selection (FS) reduces the number of features in a dataset while preserving the most crucial information. It chooses the most valuable features and removes those that are irrelevant or redundant [14-15].

To address these issues, we proposed a DENSE MLP model integrated with hybrid feature selection method combination of Z-tests and chi-square statistical tests. The Z-test can select features having the discriminatory power to distinguish the normal and attack traffic based on mean deviation. From those features, the chi-square test selects the features which are highly interacting with class labels, unlike heuristic methods such as Information Gain or Correlation-based selection. In this work, the Z-test and Chi-Square test were used for feature selection, and the DENSE MLP model is used to detect the DDoS attacks. The major contributions of this research work are mentioned as follows.

Contributions:

- This research aims to develop a DENSE MLP DDoS detection model using an efficient feature selection algorithm.
- Using the hybrid feature selection method, the system can identify the most relevant features from the network traffic. This makes training quicker and improves the system's ability to detect threats accurately.
- The proposed method was evaluated on five publicly available datasets (NSL-KDD, BoT_IoT, CICIDS2017, CICIDS2018, and CICIDS2019).
- To further demonstrate its efficacy, the DENSE MLP model outperformed the baseline models in terms of precision, recall, accuracy, specificity, balanced accuracy, and F1 score.
- The DENSE MLP model predicted the network traffic flow accurately in less than 0.00001 second. Detecting cyber threats accurately in less time is crucial for the IDS systems.

The remaining part of the paper is organized as follows: the Section II discusses the recent works that contributed to developing an IDS using feature selection methods. In the Section III, the methodology explores data preprocessing, significant feature selection process, and DDoS attacks detection. The comprehensive experimentation results on five datasets and key observations of this research work are discussed in Section IV. In the Section V, the overview and conclusion of this work is done.

II. Related Study

Several research works contributed to detecting DDoS attacks using different types of feature optimization methods.

Thaseen et al. [16] developed an attack detection system that combines chi-square feature selection with a multi-class Support Vector Machine (SVM) to identify different types of threats more effectively. This model increases the effectiveness of classification by parameterizing the RBF kernel with heuristic optimization techniques. The aim was to develop a better multi-class SVM-based IDS that reduces training and testing times while improving effectiveness in classifying destructive actions. Additionally, the model has some limitations, like higher expenses for adjusting parameters, issues with overfitting in large data sets, and the need for a lot of retraining due to changes in long-term attack patterns. Dwivedi et al. [17] came up with an approach to pick out important features by combining ensemble feature selection (EFS) with the grasshopper optimization algorithm (GOA), aiming to improve the performance of SVM classification. EFS ranked the features, and GOA picked the most important ones, which improved detection rates and reduced false alarms on the NSL-KDD and KDD Cup 99 datasets. EFS ranked features, while GOA selected the most relevant ones, enhancing detection rates and lowering false alarms on NSL-KDD and KDD Cup 99 datasets. However, the method increases computational complexity due to the dual optimization process, which may impact real-time detection efficiency.

Kanna et al. [18] developed a hybrid IDS model using a MapReduce-based CONV-LSTM DL approach. The FS is performed with the ABC algorithm, while black widow optimization (BWO) fine-tunes hyperparameters. The model achieved high accuracy on NSL-KDD, ISCX-IDS, UNSW-NB15, and CSE-CIC-IDS2018 datasets. However, training (26,721.2 s) and testing (402.67 s) times remain high, demanding significant computational resources. Dora et al. [19] introduced a DDoS detection method that combines CNN with an optimized LSTM. They apply the CP-GWO technique for feature selection, CNN for feature extraction, and refine the LSTM for improved accuracy. Despite achieving better performance than traditional methods, the study does not evaluate the computational time required. Wei et al. [20] developed a hybrid DDoS detection model using an auto encoder and a multi-layer perceptron (AE-MLP). The auto encoder selects important features, while the MLP classifies different types of DDoS attacks. The model was evaluated on the CCIDDoS2019 dataset. It effectively addresses performance overhead and bias issues caused by large datasets. Kanna et al. [21] created the OCNN-HMLSTM model, which uses an improved CNN to pick out important spatial features, LSO to fine-tune settings, and HMLSTM to manage time-related features. This approach enhances threat detection by interpreting spatial-temporal features. However, the DL model is complex, requiring extensive training time (30,665 s) and high resource consumption. The authors plan to investigate FS techniques in future work.

Sarafaldin et al. [22] developed the CICDDoS2019 dataset with real-time DDoS attack traffic. They suggested that 24 features are the most important out of the 87 features, using feature importance and RadViz plots. Using the ID3 algorithm, they achieved 78% accuracy. Pontes et al. [23] implemented an Energy-based Flow Classifier (EFC) method to address the model adaptability. Their model was evaluated on the CIDDS-

001, CICIDS17, and CICDDoS19. They utilized the CICDDoS2019 dataset, which included all features except for FlowID, Source IP, Destination IP, and Timestamp. They mentioned that if the feature selection method is applied before the EFC method, the time complexity will be reduced.

Rajagopal et al. [24] proposed for selecting the feature subset using mutual information gain (MUI) and Fisher score. Their method was evaluated on the UNSW-B15, CICIDS2017, and CICDDoS2019 datasets and got good performance. Though it was shown to have good performance for classifying DDoS attacks, it did not consider some of the DDoS attack types for experimentation.

J. Halladay et al. [25] explore the use of 25 time-based features to detect and classify types of DDoS attacks with better detection rates. The time-based feature subset effectively reduced training time without sacrificing accuracy, making it suitable for near-real-time applications. However, this work neglected low-rate DDoS and high-rate DDoS attacks detection.

Raju et al. [26] selected five features using the SHAPLY value. For this purpose, they trained five ML models and collected the five common features. They developed a classification method, K-Nearest Oracles Eliminate (KNORA-E) and K-Nearest Oracles Union (KNORA-U), to detect DDoS attacks using these five features. Their method shows better performance, but it is a cost-effective method.

TABLE I. RELATED STUDY ON EXISTING WORKS USING FEATURE SELECTION

SL	Reference	Journal Publication	Datasets	Feature Selection Method	Model	Gaps
1	Dwivedi et al. [17] & 2021	Springer	NSL-KDD	EFSGOA	ML (SVM)	Increases computational complexity due to the dual optimization process, which may impact real-time detection efficiency
2	Kanna et al. [18] & 2022	Elsevier	NSL-KDD, ISCX-IDS, UNSW-NB15, and CICIDS2018	Artificial Bee Colony	CONV-LSTM	Training and testing times remain high, demanding significant computational resources
3	Dora et al. [19] & 2022	Springer	DARPA1998, DARPA LLS DDoS-1.0 dataset, dataset, NSL-KDD, KDD cup and CICIDS2017	CP-GWO	O-LSTM	The DL model is Complex and takes extensive resources.
4	Wei et al.[20] & 2021	IEEE Access	CICDDoS2019	Auto Encoder(AE)	MLP	While using AE for feature selection it may select irrelevant features, and that susceptibility to overfitting.
5	Kanna et al. [21] &	Elsevier	NSL-KDD, ISCX-IDS and UNSWNB15	HMLSTM	OCNN-HMLSTM	The DL model is complex, requiring extensive training time (30,665 s) and high resource consumption
6	Sarafaldin et al. [22] & 2019	IEEE Conference	CICDDoS2019	Feature importance	ID3	They did not address the low-rate, high-rate, reflection, and Exploitation attack detection in detail.
7	Pontes et al. [23] & 2021	IEEE Transactions	CIDDS-001, CICIDS17 and CICDDoS2019	Not Used	EFC	The time complexity of the model is high due to the high dimensionality of the dataset.
7	Rajagopal et al. [24] & 2021	IEEE Access	UNSWB, CICIDS2017, CICDDOS2019	MUI and Fisher Score	Meta Classifier	Reflection exploitation, low-rate and high-rate attacks not addressed.
8	JHalledy et al. [25] & 2022	IEEE Access	CICDDoS2019	Manually	ML (XGB)	This work neglects low-rate and high-rate attack detection.
9	Raju et al.[26] & 2022	Elsevier	CICDDoS2019	SHPELY	KNORA	The feature selection process is complex and has a high computational cost
10	Aswani et al. [27] & 2024	Elsevier	CIC_IoT2023 and CICDDoS2019	Manual	DL(GRU)	It was lacking in testing on real-time created IoT dataset.
11	Raza et. al. [28] & 2024	MDPI	CICDDoS2019	Correlation and Chi-square	CNN, SGD and RF	It was not tested on the available DDoS attack vector diversity
12	Thi-Thu-Huong Le et al.[29] & 2025	Elsevier	APA-DDoS, DDoS-SDN , CICDDoS2019, CRCDDoS2022 and BCCC-Packet-Cloud-DDoS-2024	Correlation, MUI and Univariate	DDoSBERT	This model's limitations are dataset specificity, computational resources, and interpretability.
13	Mahdavifar et al. [30] & 2024	IEEE Transactions	CICDDoS2019	MUI	CapseRule	It was not addressed the low-rate and high-rate DDoS attacks.
14	Davrim et al. [31] & 2022	Elsevier	CICDDoS2019	Information Gain	DL(CNN-inception)	Information gain selected features may lead to bias the model performance.
15	A.A. Najar et al. [32] & 2024	Elsevier	CICDDoS2019	Information Gain and Quasi-Constant	DL(CNN)	DL models complexity is higher than the ML models
16	Raghupathi et al. [33] & 2024	INASS	CICDDoS2019	Statistical T-Test	ML (SVM)	Not addressed the low-rate and high-rate DDoS attacks detection.

A DL model was suggested using GRU (Gated Recurrent Units) by Aswani et al. [27] to tackle the DDoS attacks in IoT-enabled healthcare systems. To train the DL model, they used 22 manually selected features. We evaluated this on the CIC_IoT2023 and CICDDoS2019 datasets. Raza et al. [28] developed an IDS system using FS and ML algorithms. In their study, 16 features were selected from the CICDDoS2019 dataset using chi-square and correlation FS methods. These features gave a better performance in terms of accuracy, recall, precision, and F1 score. In this approach, they considered only NetBIOS and SYN DDoS attacks.

Thi-Thu-Huong Le et al. [29] implemented the DDoSBERT model to detect DDoS attacks. In their process, they selected important features from the network traffic using correlation, MUI, and univariate FS methods. The model was evaluated on APADDoS, DDoS-SDN, CICDDoS2019, CRCDDoS2022, and BCCC-Packet-Cloud-DDoS-2024 datasets. This methodology showcased robust performance. Despite its drawbacks, which include issues with dataset specificity, processing capacity, and interpretability, it offers a sophisticated comprehension of the model's operational bounds.

S. Mahdavifar et al. [30] developed a CapsRule IDS to detect and classify DDoS attacks using DL techniques. Using the mutual information gain method, they selected 22 features out of 87. CapsRule IDS was tested using CICDDoS2019 datasets and achieved 99.24% of accuracy. Low-rate attacks were not taken into consideration by the model, which only examined seven of the thirteen DDoS attack categories present in the CICDDoS2019 dataset.

Davrim et al. [31], Intrusion Detection System (IDS) for identifying DDoS attacks using DL techniques. It evaluates multiple models, including DNN, CNN, and LSTM, on the widely used CICDDoS2019 dataset. On this dataset, using information gained, 40 relevant features were identified. Among the models tested, the CNN-based inception model performed the best, achieving 99.99% accuracy in binary classification and 99.30% in multiclass classification. However, this study did not discuss low-rate and high-rate attacks. A.A. Najar et al. [32] proposed an FS method to build a reliable intrusion detection system (IDS) for identifying and classifying DDoS attacks. They identified 43 features. Evaluated using the CICDDoS2019 dataset, the model achieved 96.82% accuracy, recall, and precision, along with a 96.50% F1 score and detected attacks in 0.189 milliseconds, surpassing baseline models. Raghupathi et al. [33] suggested a statistical T-test method to optimize the DDoS attack detection system. Their FS method evaluated the CICDDoS2019 dataset and obtained 59 important features with 95% confidence. These features gave better performance using the SVM classifier. However, there is scope to reduce the feature space.

The above-mentioned existing studies, given their good performance with their IDS models, add the feature selection methods depicted in Table I. However, some of these studies do not address the low-rate, high-rate DDoS attack detection, and some of the studies do not address the reflection and exploitation DDoS attacks.

III. METHODOLOGY

A. Preprocessing

Data Cleaning: The training dataset D contains instances $X = \{X_1, X_2, \ldots X_n\}$, feature set $F = \{F^1, F^2, \ldots F^m\}$, and target label $Y = \{Y_1, Y_2, \ldots Y_n\}$, where n and m represent the number of instances and features in the dataset. In the step 1, the dataset D is given as input to the Algorithm. In the dataset cleaning, we removed the samples that contained any one of the following: missing, Not a Number(NaN), or infinity. In CICDDoS2019, CICIDS2017, and BoT_IoT datasets, the samples containing the NaN and infinity values are removed.

Unwanted features play a minimal role in the detection or classification process; eliminating these features could enhance the model's performance. To eliminate the unwanted features from the dataset, we followed three methods. Socket-related features are removed at first; followed by redundant and highly correlated; and the third are invariant features.

Removing Socket-Related Features: Socket-related features are discarded based on domain knowledge to ensure that the model's performance will be efficient on various networks. The socket features ('FlowID', 'Destination IP', 'Source IP', 'Destination Port', 'Source Port', 'Protocol', 'TimeStamp', and 'SimilarHTTP') are removed from the F and the feature set is updated to, where $F^1 \subseteq F$. Step 2 from Algorithm presents socket feature removal process.

Removing Redundant Features: In the second method using Eq. (1), we calculated the correlation coefficient according to [34] for each feature with the rest of the features. A correlation value of 1 indicates that both features are contributing equally. Since preserving both the features leads to redundancy, from those feature pairs, we discarded the second feature and retained the first feature. At the step 3, the Algorithm handles the highly correlated feature elimination process. In step 3, the mean values x' and y' of each pair of the features $F_i^{1\,\text{th}}$ and $F_j^{1\,\text{th}}$, where $i,j=\{1,2,3,\ldots l \mid l \subseteq m\}$ are computed. Using Eq. (1), the correlation coefficient ρ value is calculated where x_k and y_k are the values of features F_i^1 and F_j^1, $k=\{1,2,3,\ldots n\}$. To reduce the redundant features in the dataset, we discard the feature F_j^1 from F^1 if($\rho==1$) and $i \neq j$ and update the feature set to F^2 where $F^2 \subseteq F^1$.

$$Correlation\ Coefficient\ (\rho) = \frac{\sum_{k=1}^{n}(x_k - x')\,(y_k - y')}{\sqrt{\sum_{k=1}^{n}(x_k - x')^2}\sum_{i=1}^{n}(y_k - y')^2} \qquad (1)$$

Algorithm: Hybrid Feature Selection using Z-test and Chi-Square

Input: Dataset *D* with features set $F = \{F_1, F_2 \ldots F_n\}$ and target variable *Y*

Output: Select the significant features set *S*.

Start

1. Load the IDS training dataset *D* with feature set F
2. Remove socket-related features
   $F = \{F_1, F_2 \ldots F_n\}$, F_s = { 'FlowID', 'Destination IP', 'Source IP', 'Destination

Port', 'Source Port', 'Protocol', 'TimeStamp', and 'SimilarHTTP' }

$F^1 \leftarrow F \setminus F_s$

3. Compute the correlation coefficient $\rho(F_i^1, F_j^1)$ *of each* pair of features F_i^1, F_j^1

   And remove the feature F_j^1 if $|\rho(F_i^1, F_j^1)| == 1$ and $i \neq j$

   $F^2 \leftarrow F_i^1$

4. Compute the standard deviation $\sigma(F_i^2)$ for each feature F_i^2

   And remove the feature F_i^2 if $\sigma(F_i^2) == 0$

   $F^3 \leftarrow \{F_i^2 \mid \sigma(F_i^2) != 0\}$

5. Dataset *D* with features F^3 and values $x \in F^3$

   If *x* is negative value

   $F^3(x) \leftarrow \{(10^{d(x)} - 1) \mid if\ (x < 0)\}$.

   where *d(x)* is the number of digits in *x*

6. Compute the z-score for each feature F_i^3

   $Z = \frac{x' - \mu}{\sigma\sqrt{n}}$, where x': is the mean of a feature F_i^3, σ: is the standard deviation of feature F_i^3, and μ: is the population mean

7. Compute the median value $Z'$of the Z scores

$$Z' = \begin{cases} \left(\frac{n+1}{2}\right)^{th}\ term\ of\ Z_{Score}, if\ n\ is\ odd\ number \\ \frac{\left(\frac{n}{2}\right)^{th} + \left(\frac{n}{2}+1\right)^{th}}{2}\ term\ of\ Z_{Score}, if\ n\ is\ even\ number \end{cases}$$

   Select the feature F_i^3 if $Z \geq Z'$

   $F^4 \leftarrow \{F_i^3 \mid Z(F_i^3) \geq Z'\}$

8. Compute Chi-Score(χ^2) for each feature F_i^4

   $\chi^2(F_i^4) = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$ where O_i: is the obtained frequency, E_i: is the expected frequency, and $median\ (\chi^2(F^4))$: is the median of chi-score values

   $S \leftarrow \{F_i^4 \mid \chi^2(F^4) \geq median\ (\chi^2(F^4))\}$

9. S is the optimal feature set and retain from the dataset D

Stop

Removing Invariant Features: After the step 3, the updated feature set is F^2. In the step 4, calculate σ value for each feature F_i^2 from the F^2, and remove the features whose σ values are equal to zero, then the feature set is updated to F^3. If the data value of any feature is the same value throughout the column, it will not contribute to the process of detection or classification. These types of features can be found using the standard deviation method. The standard deviation (σ) of each feature F_i^2 is calculated using Eq. (2), where i={1,2,3,…h | h ⊆ *l* }. The x_k and x' are the value and mean of F_i^2, where k={1,2,3,…n}. The σ value is the square root of the sum of the squared difference of x_k and x' divided by (n-1). We removed the feature F_i^2, If (σ = 0), then updated the feature set to F^3 , where $F^2 \subseteq F^3$.

$$Standard\ Deviation(\sigma) = \sqrt{\frac{\sum_{k=1}^{n}(x_k - x')^2}{n-1}} \quad (2)$$

Data Imputation: In the datasets, we found that some of the features contained negative values. Those negative values are assumed to be outlier values in the dataset, and instead of removing those, they are replaced with higher integer values represented with some 9's, which is equal to the number of digits of that value. The dataset D with the feature set F^3 is updated as $F_i^3(x_k) = =$ number of 9's if $F_i^3(x_k) < 0$. For instance, if the value is -0.12, it will be replaced with the value 99. Step 5 from Algorithm represents the imputation process.

B. Hybrid Feature Selection

The process of selecting important features was conducted sequentially in two stages. In the first stage, the Z-test was applied.

Z-Test: The Z-test is a statistical test used to determine whether there is a statistically significant difference between a sample mean and a population mean of a feature or between two sample means when the population standard deviation is known and the sample size is large. At step 6, the Z-test was applied on individual F_i^3th feature and obtained Z_score^i using Eq. (3), where i={1,2,3,…g | g ⊆ h}. The x' and σ are the mean and standard deviation of ith feature of F^3, μ is the population mean, and n is the number of samples. After obtaining the Z-scores of all features, at step 7, the median of Z-scores (Z') is computed using Eq. (4). We chose Z' as the threshold to select the vital feature. The ith feature of F^3 can be considered as a vital feature if Z_score$^i \geq$ Z' , else it is discarded. After completing the first stage of feature selection, vital features are stored in feature set F^4.

$$Z_{Score} = \frac{x' - \mu}{\sigma/\sqrt{n}} \quad (3)$$

$$Z' = \begin{cases} \left(\frac{n+1}{2}\right)^{th}\ term\ of\ Z_{Score}, if\ n\ is\ odd\ number \\ \frac{\left(\frac{n}{2}\right)^{th} + \left(\frac{n}{2}+1\right)^{th}}{2}\ term\ of\ Z_{Score}, if\ n\ is\ even\ number \end{cases} \quad (4)$$

Chi-Square: In the second stage, we applied the chi-square test using Eq. (5). It is a non-parametric statistical test used to determine if there is a statistically significant association between two categorical variables. We discretize the continuous data with 2 bins before applying the chi-square test, as the IDS datasets contain both discrete and continuous values.

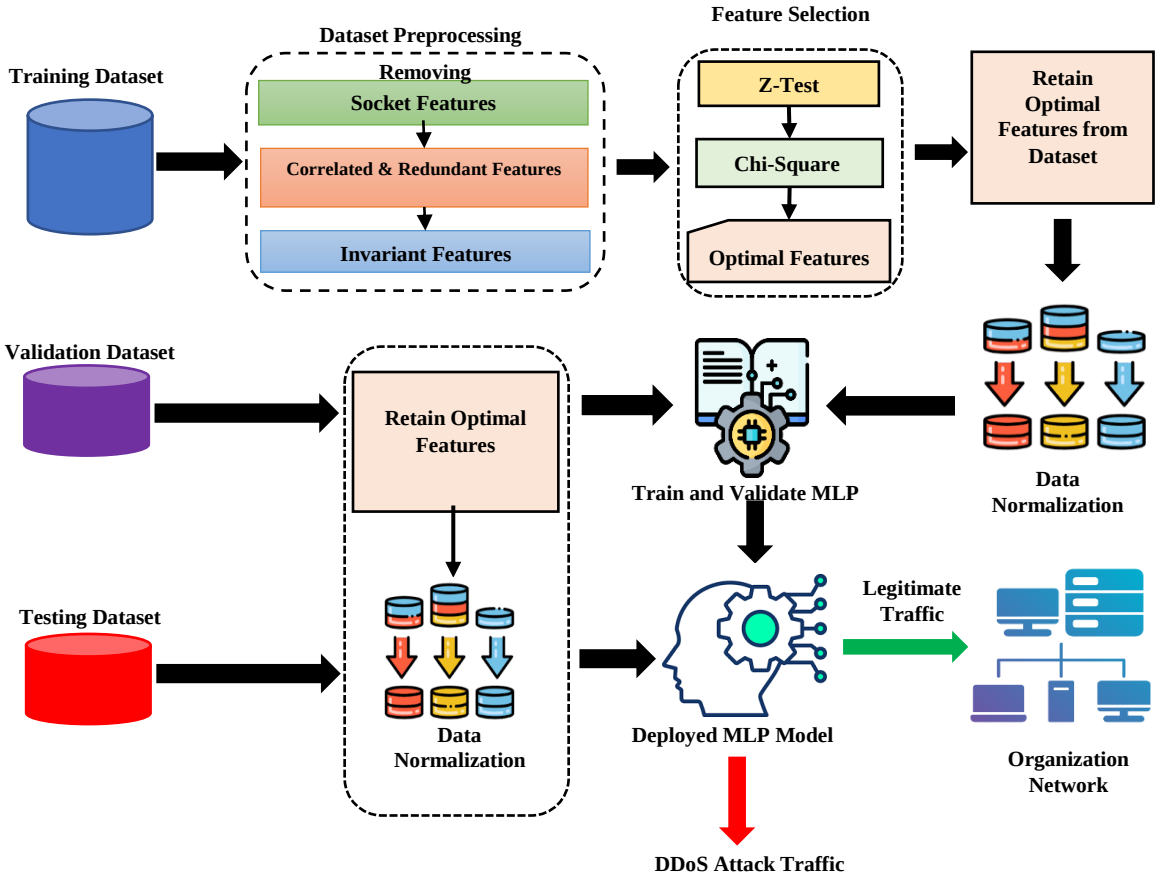

Fig. 1. The proposed IDS framework integrating hybrid feature selection and DNN

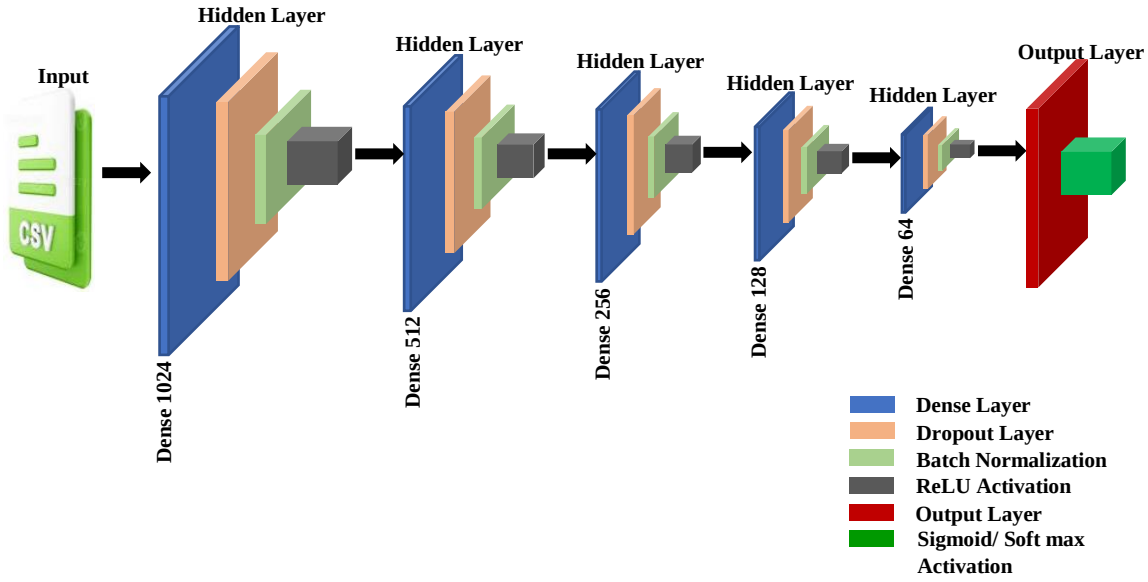

Fig. 2. DENSE MLP architecture

In the dataset D with feature set F^4, data x can be grouped into 2 bins as $F_i^4(x^1) \leq mean(x) and\ F_i^4(x^2) > mean(x)$, where i= {1, 2, 3,...e}. Then, we obtain the observed frequency count O_b and expected frequency count E_b for each bin, where b = {1, 2}. Using O_b and E_b information, create a contingency table for each F_i^4. From the contingency table, we calculate the

chi-square (χ_i^2) value for each F_i^4using Eq. (5). The threshold value was taken as the median value of chi-squares, which is calculated using Eq. (4).

The F_i^4 is considered an optimal feature if $F_i^4(\chi_i^2) \geqslant$ median value of chi-squares; otherwise, it is discarded. Optimal features are stored in S and retained from dataset D, where S ⊆ F^4. This process is presented at step 8 and step 9 in Algorithm.

$$Chi - Square\ (\chi^2) = \sum_{b=1}^{2} \frac{(O_b - E_b)^2}{E_b} \tag{5}$$

C. Data Normalization

Data normalization is a task to put all the data values on the same scale. For instance, the flow duration feature values lies between 0.44 and 1.25 milliseconds, and the flow packets/s feature values lie between 20 and 1000024. This type of data may lead to bias in the learning model. For this reason, after obtaining the optimal features dataset, we normalized the data using the min-max method using Eq. (6) to put the data values of each feature on the same scale, where X is the original data and X' is the normalized data.

$$X' = \frac{X - \text{minimum}(X)}{maximum(X) - minimum\ (X)} \tag{6}$$

D. Dense Multi-Layer Perceptron Architecture (DENSEMLP)

The DENSE MLP model takes the preprocessed dataset $D_{X,S,Y}$ as input to the training model, where X'= {X_1, X_2, … X_n}, S={F_1, F_2, …. F_m} and Y={C_1, C_2, …. C_n}. Fig. 1, depicts the proposed IDS frame work integrating hybrid feature selection and DNN. As shown in Fig. 2, it consists of one input, one output, and five hidden layers; each layer is a fully connected layer. The input layer contains the number of neurons equal to the size of the optimal feature set S.

The first to fifth hidden layers consist of 1024, 512, 256, 128, and 64 neurons, respectively. For each hidden layer, we apply activation function, dropout, and batch normalization operations. Each neuron performs the function f as mentioned in the Eq. (7). The function f is the net weighted sum of the product of weights W, input X′ and bias b for each neuron at layer *l*, where Wand b are the weights vector and bias of the layer *l* and φ is the activation function.

$$\begin{aligned} f^{(l)} &= W^{(l)T} * X' + b^{(l)} \\ f &= \varphi^l\left(\sum_{l=1}^{5} f^{(l-1)} * W^{(l)T} + b^{(l)}\right) \end{aligned} \tag{7}$$

$$\varphi(x) = max(0, x) \tag{8}$$

The DENSE MLP model calculates the weighted sum plus bias from the 1024 neurons from the first layer and gives it as input to the next layer. This process continues till the fifth hidden layer, and finally, 64 neurons are fully connected to the output layer. To make the computations simple and efficient at the hidden layers, the ReLU activation function introduce non-linearity into the data using Eq. (8). The output layer gives $\hat{y}$ for each given input. At the output layer, we used the activation function sigmoid for the binary class and soft-max for the multi-class classification. The $\hat{y}$ is calculated using Eq. (9), where C is the number of class labels and f is the input to the output layer.

$$\hat{y} = \begin{cases} \dfrac{1}{1+e^{-f}} & if\ classification\ is\ binary - class \\ \dfrac{e^f}{\sum_{j=1}^{C} e^f} & if\ classification\ is\ multi - class \end{cases} \tag{9}$$

To fine-tune and improve the DENSE MLP model efficiency, we used the Adaptive Moment Estimator (ADAM) optimizer along with two other hyper parameters: a learning rate (η) of 0.001 and an error rate (θ) of 0.001. The ADAM optimizer minimizes the training loss, using the gradient descent method. The loss value L is computed using cross-entropy [35] as mentioned in Eq. (10), where y is the actual output, and $\hat{y}$ is the predicted output for the ith training instance and jth target class label. Using the L, calculate the error term (δ^l) by partial derivative to the L with respect to predicted value f^l as mentioned in Eq. (11).

After obtaining the δ^l, find the gradients for $W^{(l)T}$and $b^{(l)}$ using Eq. (12), and update the weights ($W^{(l)T}$) and bias ($b^{(l)}$) as mentioned in Eq. (13), where η is a learning rate, *l* represents the layer.

Using the updated $W^{(l)T}$ and $b^{(l)}$ information, the DENSE MLP model will carry the next iteration. In this model, the number of epochs is set as 50. The iterative execution continues till the 50 epochs if the loss function is minimized anywhere before the 50th epoch. To address this, we used an early stopping method with patients =10, which will monitor continuous change in the validation loss value. It will stop the DENSE MLP model execution, when the loss value is not changing continuously in 10 epochs. Then, it restores the best weights to the model. Table II, depicts the hyperparameters, which are used to build the model.

$$L = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C} y_{ij} * \log(\hat{y}_{ij}) \tag{10}$$

$$\delta^l = \frac{\partial L}{\partial f^l} \tag{11}$$

$$\begin{aligned} \frac{\partial L}{\partial W^{(l)T}} &= \delta^l * f^{(l-1)} \\ \frac{\partial L}{\partial b^{(l)}} &= \delta^l \end{aligned} \tag{12}$$

$$\begin{aligned} W^{(l)T} &= W^{(l)T} - \eta * \frac{\partial L}{\partial W^{(l)T}} \\ b^{(l)} &= b^{(l)} - \eta * \frac{\partial L}{\partial b^{(l)}} \end{aligned} \tag{13}$$

TABLE II. THE DENSE MLP MODEL HYPERPARAMETER INFORMATION

SN	Parameter	Value
1	Number of layers	1 Input, 1 Output, and 5 Hidden
2	Input layer	16 neurons
3	Hidden layers	1024,512,256, 128, and 64 neurons
4	Output layer	1 neuron for binary and 3 neurons for three classes
5	Batch size	1024
6	Activation Function at Hidden Layers	ReLU
7	Drop out	0.5
8	Batch normalization	After each layer
9	Activation Function at Output Layer	Sigmoid for binary classification SoftMax for multi-class classification
10	Optimizer	ADAM
11	Learning rate	0.01

12	Error rate	0.001
13	Loss function	Cross-Entropy
14	Number of Epochs	50

IV. Experimentation And Results Discussion

In this section, we discuss the experimental results of the proposed method on the five datasets mentioned in Table III. Experimentation was carried out on the Intel i5 processor with a 1.9 GHz clock speed and 198 GB of RAM LENOVO Think Server system. To simulate the entire process, a Python script has been executed on the Jupyter Notebook. The performance of the proposed method was evaluated on the five datasets NSL-KDD, CICIDS2017, CICIDS2018, CICDDoS2019, and BoT-IoT.

A. Datasets

In this paper, we utilized five publicly available IDS datasets (NSL-KDD, CICIDS2017, CICIDS2018, CICDDoS2019, and Bot_IoT2022).

1) NSL-KDD: The NSL-KDD dataset [36], an improved version of KDD-99, is widely used in intrusion detection research. It contains 125,973 training and 22,544 test flows with 41 features — 38 continuous and 3 categorical. The dataset includes 40 labels: 23 are used in the training set and 17 in the test set, which creates label discrepancies. Labels are grouped into Normal and Attack categories for analysis. The training and validation datasets are considered in an 80% and 20% split from the training dataset.

2) CICIDS2017: The CICIDS2017 [37] is an improved dataset that satisfies the 11 properties of IDS dataset [38]. It overcomes the drawback of its previous IDS datasets (NSL-KDD, ISCX-2012, etc.). The dataset is formed by capturing, from Monday to Friday, a total of five days of network data and consists of benign and 9 different types of attacks. From these five days, Friday's captured network data is considered and used in this paper, which contains the benign and DDoS traffic flows of 97686 and 128025 samples with 79 features and 1 target class. We split these traffic flows into training, validation, and testing datasets according to an 80-10-10 percent rule.

3) CICIDS2018: The CICIDS2018 dataset was covered with benign and 14 different types of attacks, and it is publicly available at https://www.unb.ca/cic/datasets/ids-2018.html in CSV and PCAP format. This dataset captured the network traffic data with 79 features and one target class over a period of ten days. In a total of ten days, they executed the network attacks and captured the network traffic data with 79 features and one target class. Among these 14 network attacks, we utilize DDOS attack-HOIC and DDOS attack-LOIC-UDP traffic flows. Since the CICIDS2018 dataset isn't available for separate training and testing, these traffic flows are divided into training, validation, and testing datasets in an 80-10-10 percentage split.

4) CICDDoS2019: The CICIDS2017 and CICIDS2018 datasets covered various types of network attacks, and in those, there were fewer DDoS attack categories. These two datasets do not have the training and testing of distinct traffic flows. To overcome these problems, the CICDDoS2019 [22] dataset was generated by the Canadian Institute of Cyber Security and is publicly available at https://www.unb.ca/cic/datasets/ddos-2019.html in CSV and PCAP format. This dataset was captured on the test bed network, which mimics the real-time network. The CICDDoS2019 dataset was captured on two different days. They executed 12 different types of DDoS attacks (DNS, MSSQL, LDAP, TFTP, SNMP, SSDP, UDP, UDPLag, Syn, NetBIOS, NTP, and WebDDoS) on the first day and 7 different types of DDoS attacks (MSSQL, LDAP, UDP, UDPLag, Syn, NetBIOS, and PortMap) on the other day. From the first day traffic flows, we considered 399998 for training, which covers all the DDoS attack classes and benign classes. From the second day, we formed four distinct testing datasets (D1, D2, D3, and D4) with 112,611 flows. Among these four distinct sets, use the three sets for testing and one for validation.

5) BoT_IoT: The BoT-IoT [39] dataset was made by setting up a real-life network in the Cyber Range Lab of UNSW Canberra, with labels showing different attack patterns. This dataset covered the 4 different types of attack flows along with benign flows. Among the four types of attacks, we considered DDoS attack flows in this paper. This dataset is publicly available in CSV and PCAP formats, but the PCAP format is used in this paper evaluation. The PCAP data was converted into CSV format using CICFlowmeter [40 - 43], which will convert the PCAP to CSV format with 83 features and one target class. After conversion, we collected 242289 flows from DDoS attacks and 37711 flows from benign. We divided these samples into training, validation, and testing datasets according to the 80-10-10 percent rule. Table III displays the information from the aforementioned five datasets.

TABLE III. Dataset Samples Used for Experimentation

SN	Dataset	Training Samples	Validation Samples	Testing Samples	Number of Features
1	NSL-KDD	100778	25195	22544	41
2	CICIDS2017	180568	22571	22572	79
3	CICIDS2018	320000	40000	40000	79
4	Bot_IoT	280000	60000	60000	83
5	CICDDoS2019	399998	112611	112611	87

B. Evaluation Metrics

In this work, we first use the confusion matrix layout as shown in Fig. 3. From this, we calculated evaluation metrics like such as accuracy (Acc.) = TP/(TP+TN), recall (Rec./Sns.) =TP/(TP+FN), precision (Pre.) = TP/(TP+FP), F-score = 0.5*((Pre. * Rec.)/ (Pre. + Rec.)), specificity (Spe) = TN/(TN+FP), balanced accuracy (BA) = (Pre. + Rec.)/2, in addition to these, receivers operating system (ROC) curves are also obtained.

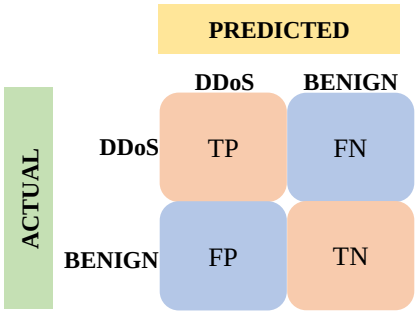

Fig. 3. Confusion matrix

TABLE IV. THE PROPOSED DENSE MLP MODEL EVALUATION RESULTS FOR DDOS DETECTION OVER THE FIVE DATASETS

Dataset	NFS[a]	Sns[b]	Spe[c]	Pre[d]	Acc[e]	FS[f]	BA[g]	Training Time (Sec)	Memory (MB)
CICDDoS2019	15	99.67	99.95	99.95	99.63	0.9963	99.63	377.25	193.58
		99.74	99.59	99.59	99.66	0.9966	99.66		
		99.85	99.59	99.59	99.72	0.9972	99.72		
		99.87	99.59	99.59	99.73	0.9973	99.73		
CICIDS2018	15	100	100	100	100	1.00	100	368.68	182.59
CICIDS2017	15	99.93	98.75	99.06	99.42	0.9949	99.34	117.90	77.45
NSL_KDD	10	57.81	93.21	91.84	73.06	0.7096	75.51	238.66	160.73
Bot_IoT	16	96.06	83.66	97.44	94.40	0.9675	89.86	249.66	172.67

[a] Number of features, [b.] Sensitivity, [c] Specificity, [d.] Precision, [e] Accuracy, [f.] F-Score, [g] Balanced Accuracy.

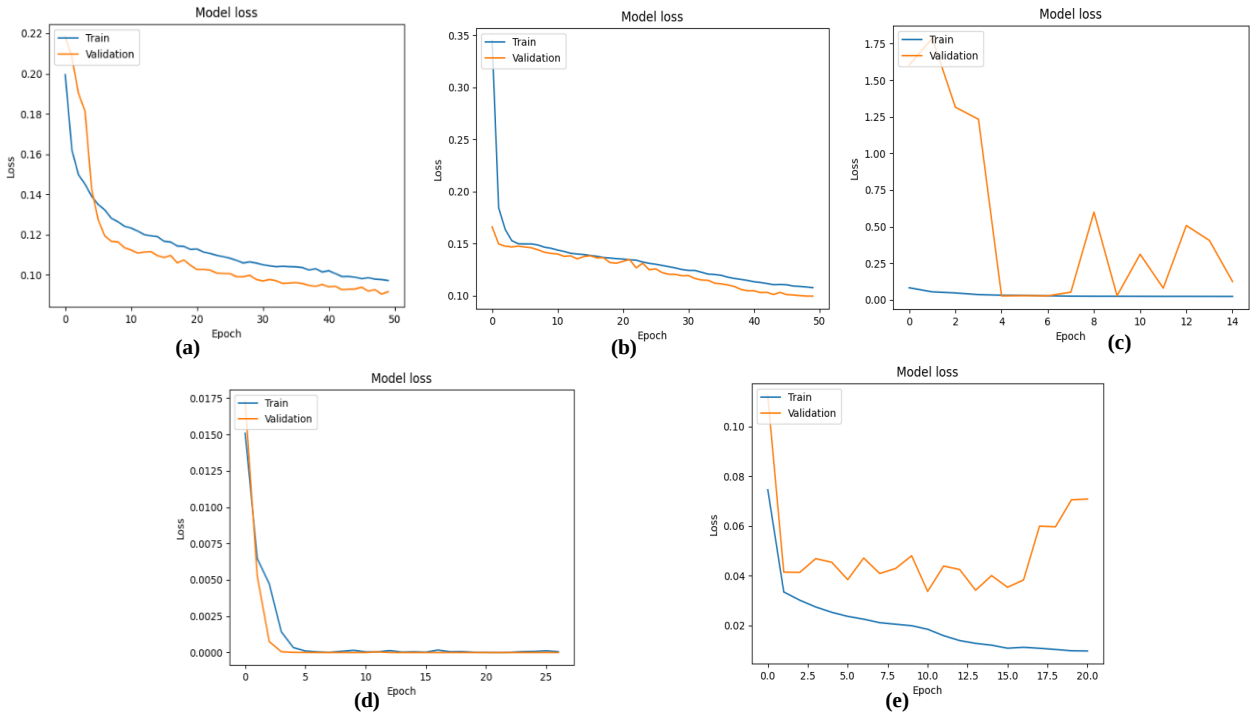

Fig. 4. The DENSE MLP model loss curves of binary class classification: a. NSL-KDD dataset, b. Bot_IoT dataset, c. CICIDS2017 dataset, d. CICIDS2018 dataset, e. CICDDoS2019 dataset.

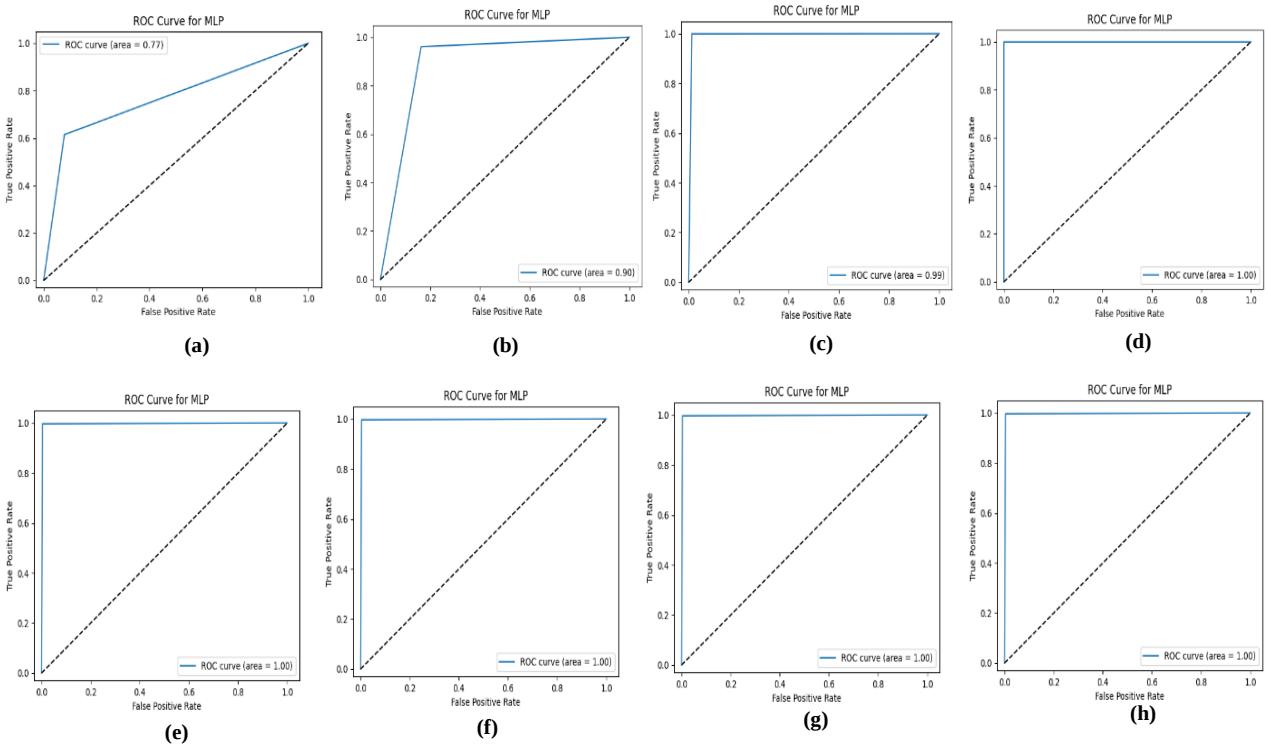

Fig. 5. Binary class classification ROC curves: a. NSL-KDD dataset, b. Bot_IoT dataset, c. CICIDS2017 dataset, d. CICIDS2018 dataset, e. CICDDoS2019 (D1) dataset, f. CICDDoS2019 (D2) dataset, g. CICDDoS2019(D3) dataset, and h. CICDDoS2019 (D4) dataset

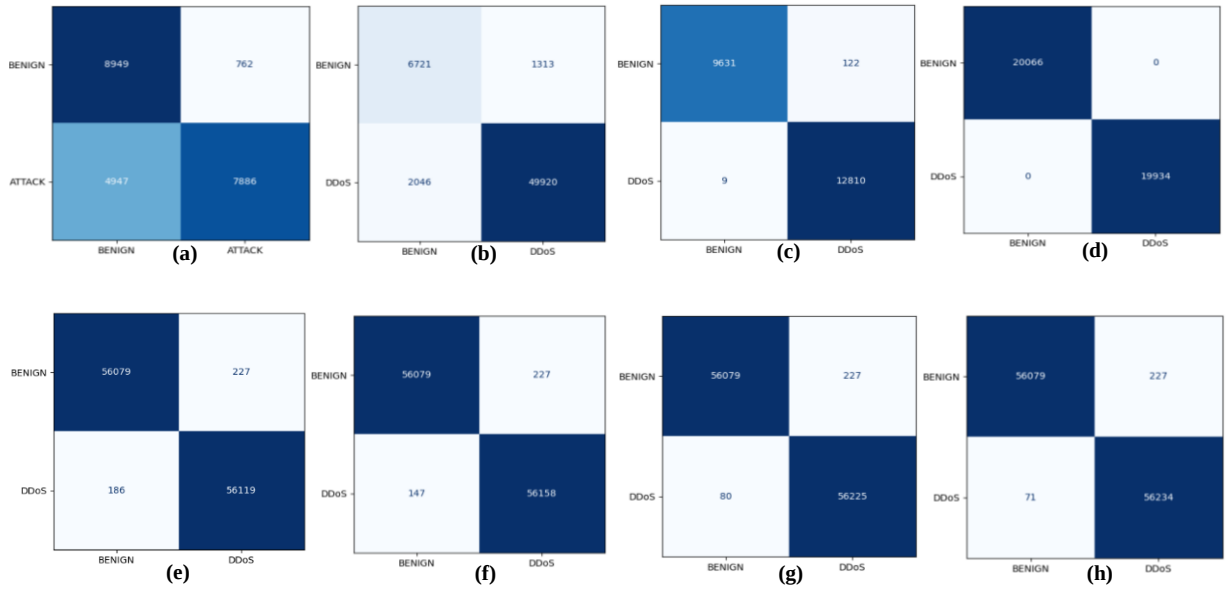

Fig. 6. Binary class confusion matrix a. NSL-KDD dataset, b. Bot_IoT dataset, c. CICIDS2017 dataset, d. CICIDS2018 dataset, e. CICDDoS2019(D1) dataset, f. CICDDoS2019(D2) dataset, g. CICDDoS2019(D3) dataset, and h. CICDDoS2019(D4) dataset.

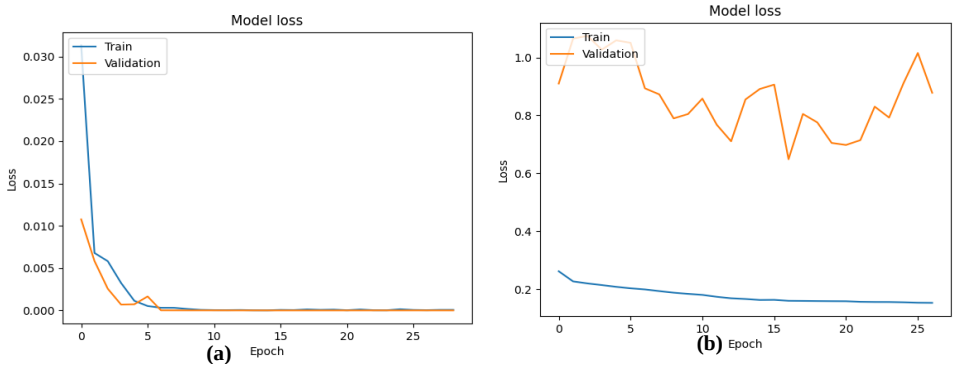

Fig. 7. Three class classification DENSE MLP loss curves a. CICIDS2018 dataset and b. CICDDoS2019 dataset.

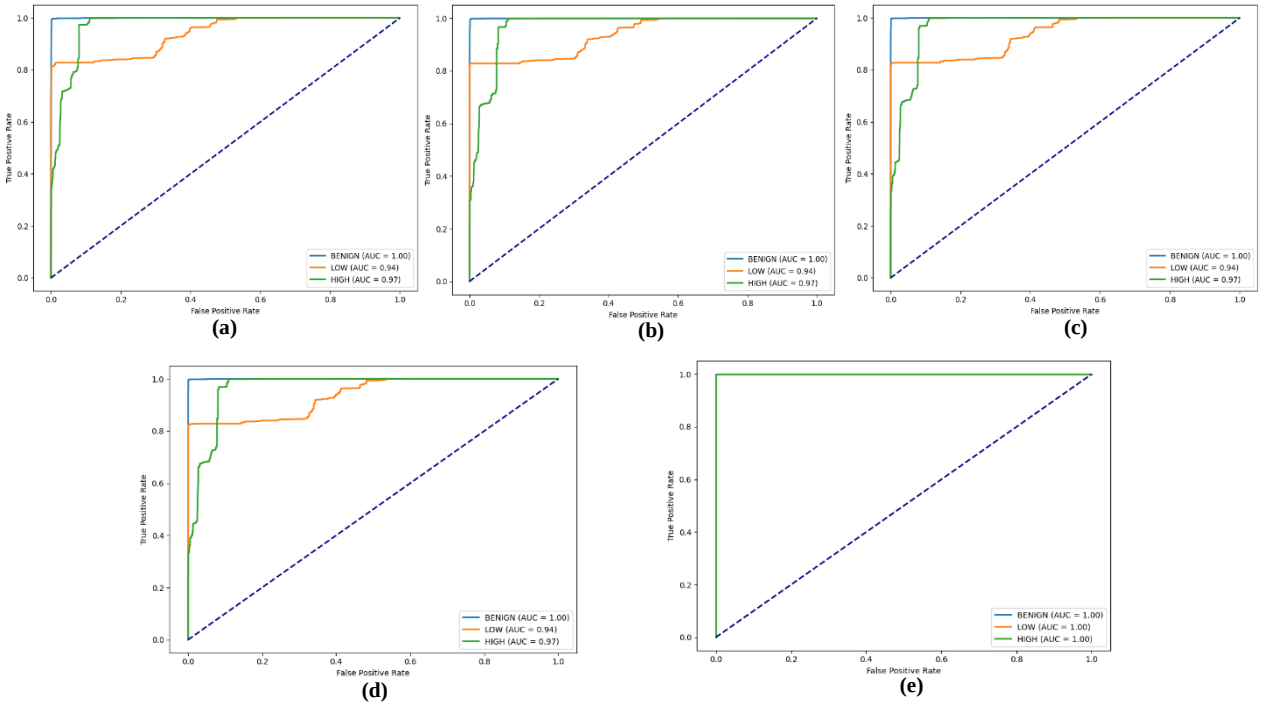

Fig. 8. Low-rate DDoS and High-rate DDoS attack classification ROC curves: a. CICDDoS2019 (D1) dataset, b. CICDDoS2019 (D2) dataset, c. CICDDoS2019 (D3) dataset, d. CICDDoS2019(D4) dataset, e. CICIDS2018 dataset.

TABLE V. THE PROPOSED MODEL BALANCED ACCURACY METRIC RESULTS OVER THE FIVE DATASETS

SL	Model	2019[a]			2018[b]	2017[c]	2009[d]	2022[e]
		D1	D3	D4				
1	ADB	94.06	94.34	94.35	100	50.00	76.99	89.95
2	DT	95.37	95.37	95.38	100	58.84	75.83	94.75
3	EXT	97.78	97.80	97.81	99.56	81.85	74.53	92.48
4	KNN	98.02	98.04	98.04	100	81.84	75.59	95.44
5	LDA	89.29	89.77	89.77	99.63	96.63	75.32	91.13
6	LR	99.24	99.35	99.35	99.63	97.82	77.40	78.55
7	NB	90.92	91.45	91.45	99.63	50.00	75.49	85.17
8	QDA	95.09	96.20	96.22	74.48	50.00	76.83	90.23
9	RF	97.53	97.54	97.54	75.67	59.11	73.45	94.14
10	Ridge	89.19	89.67	89.67	99.62	96.63	75.32	82.11
11	XGB	98.41	98.43	98.43	99.99	68.05	74.92	96.88
12	**DENSE MLP**	**99.63**	**99.72**	**99.73**	**100**	**99.34**	**75.51**	**89.86**

[a] CICDDoS2019 Dataset, [b.] CICIDS2018 Dataset, [c] CICIDS2017 Dataset [d.] NSL-KDD datset, [e] BoT_IoT Datasets

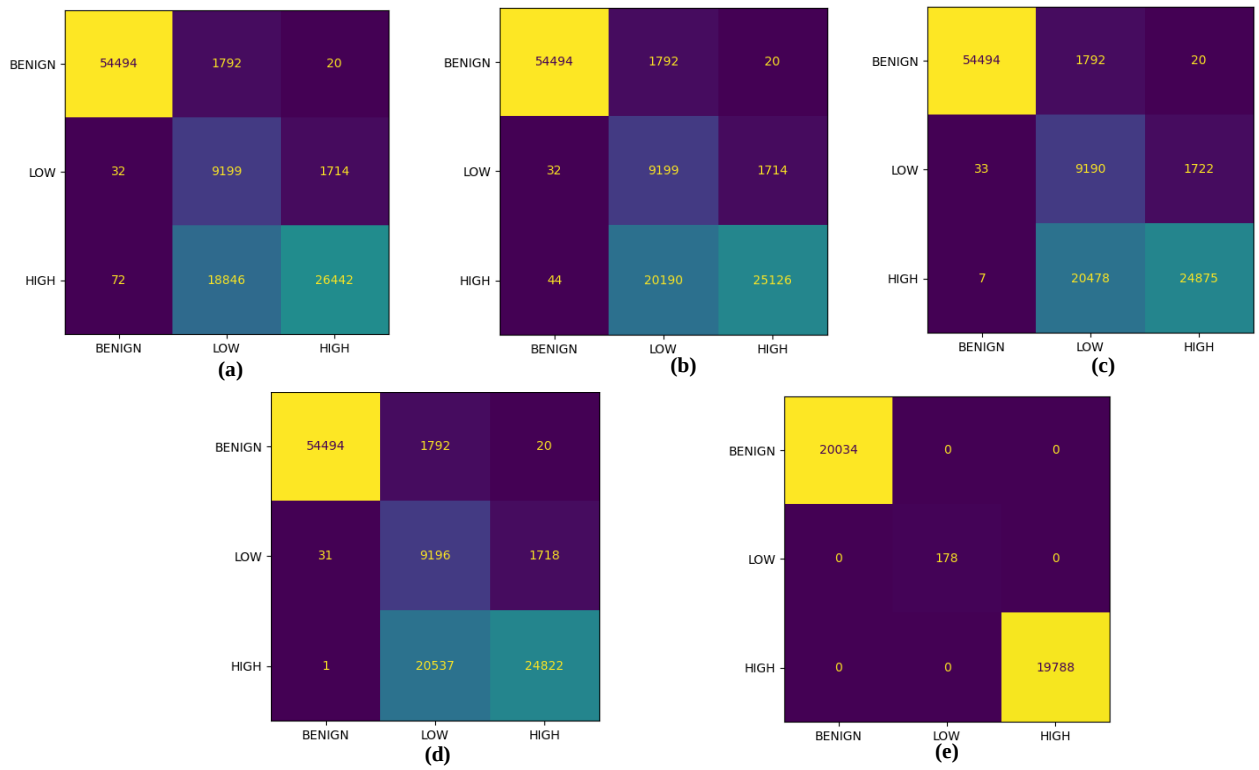

Fig. 9. Low-rate DDoS and High-rate DDoS attack classification confusion matrices: a. CICDDoS2019 (D1) dataset, b. CICDDoS2019 (D2) dataset, c. CICDDoS2019 (D3) dataset, d. CICDDoS2019 (D4) dataset, e. CICIDS2018 dataset.

TABLE VI. THE DENSE MLP MODEL EVALUATION METRICS FOR LOW-RATE AND HIGH-RATE DDOS DETECTION OVER THE CICIDS2018 AND CICDDOS2019 DATASETS

Test Dataset		Class Label	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F_Score	Balanced Accuracy (%)	Training Time (Sec)	Memory (MB)
CICDDoS2019	D1	BENIGN	96.78	99.82	99.81	98.30	0.9827	98.30	493.35	169.13
		LOW	84.05	79.70	30.83	80.12	0.4511	81.87		
		HIGH	58.29	97.42	93.85	81.66	0.7192	77.86		
	D2	BENIGN	96.78	99.87	99.86	98.32	0.9830	98.32		
		LOW	84.05	78.38	29.50	78.93	0.4367	81.21		
		HIGH	55.39	97.42	93.54	80.49	0.6958	76.41		
	D3	BENIGN	96.78	99.93	99.93	98.36	0.9833	98.36		
		LOW	83.97	78.09	29.21	78.67	0.4334	81.03		
		HIGH	54.84	97.41	93.46	80.26	0.6912	76.12		
	D4	BENIGN	96.78	99.94	99.94	98.36	0.9834	98.36		
		LOW	84.02	78.04	29.17	78.62	0.4331	81.03		
		HIGH	54.72	97.42	93.46	80.22	0.6903	76.07		
CICIDS2018		BENIGN	100	100	100	100	1.0	100	472.83	195.94
		LOW	100	100	100	100	1.0	100		
		HIGH	100	100	100	100	1.0	100		

C. *Results and Discussions*

1) Evaluation on NSL-KDD:

We split the NSL-KDD training dataset into 80-20% training and validation datasets. Training and validation datasets consist of 100778 and 25195 samples with 41 features and 38 different types of attacks. Those 38 attacks were categorized into benign and attack. Using the proposed feature selection method, among the 41 features, ten significant features 'dst_host_srv_serror_rate',‘dst_host_serror_rate','srv_serror_rate',‘serror_rate’,‘srv_count','dst_host_same_src_port_rate',‘dst_host_diff_srv_rate’,‘logged_in’,‘same_srv_rate','dst_host_same_srv_rate' are identified. These features were retained from training, validation and testing datasets. The DENSE MLP model was trained using training dataset and validated using the validation dataset. While building the model, it took 238.68 seconds and used 160.73 MB memory.

Fig. 4 shows a gradual decrease in both training and validation losses. The testing dataset model evaluation metrics are accuracy: 75.51%, sensitivity: 57.81%, precision: 91.84%, specificity: 93.21%, F-score: 0.7096, and balanced accuracy: 75.51%, as depicted in Table IV. Using the ten significant features, performance of the eleven machine learning models

was observed, as depicted in Table V. From these results, we observed that the DENSE MLP model performance is adequate with the eleven baseline machine learning models using the significant features.

2) Evaluation on Bot-IoT Dataset:

As we discussed in section 3, a training dataset with 83 input features was given to the proposed feature selection method. In the pre-processing phase, we discarded the 23 features. Following, using the Z-test reduced the feature count to 31. The chi-square test was applied on these 31 features in the training dataset and selected the 16 significant features: Max Packet Length,' 'Packet Length Std,' 'Source Port,' 'PSH Flag Count,' 'Fwd IAT Total,' 'Flow Duration,' 'Fwd IAT Max,' 'Flow IAT Mean,' 'Fwd IAT Mean,' 'Idle Max,' 'Flow IAT Max,' 'Idle Min,' 'Idle Mean,' 'Fwd Packet Length Max,' 'Fwd Packet Length Mean,' and 'Avg Fwd Segment Size'. Using this method, 80.7% of feature space was optimized. The attack target class labels were encoded as DDoS, and normal traffic as BENIGN. The significant features training and validation datasets gave to the DENSE MLP for the training; while training, the loss was decreased gradually, and the validation loss was converged at the 19th epoch. After the 19th epoch, we restored the optimal model weights and conducted testing on the DENSE MLP model. While training the model, it took 249.66 seconds and occupied 172.67 MB memory. The proposed method's performance evaluation metrics are accuracy: 94.40%, sensitivity: 96.06%, precision: 97.44%, specificity: 83.66%, F-score: 0.9675, and balanced accuracy: 89.86%, depicted in Table IV. The DENSE MLP model using BoT_IoT dataset showed accuracy performance higher than LR, NB and Ridge models, whereas it showed lower accuracy performance than DT, EXT, KNN, LDA, QDA, RF, and XGB models.

3) Evaluation on CICIDS2017 Dataset:

The proposed method for choosing features was used on the training dataset, which has 79 features. During the preprocessing, one socket feature, nine extra features that were too similar, and ten features that didn't change were removed. After this cleaning up, the number of features went down from 79 to 60. After pre-processing, the features are reduced from 79 to 60. From the 60 features, the Z-test has identified 30 relevant features. Finally, the chi-square method selected the 15 significant features 'Bwd Packet Length Max,' 'Init_Win_bytes_forward,' 'Destination Port,' 'URG Flag Count,' 'min_seg_size_forward,' 'Average Packet Size,' 'Packet Length Mean,' 'Max Packet Length,' 'Down/Up Ratio,' 'Packet Length Std,' 'Bwd Packet Length Std,' 'Bwd Packet Length Mean,' 'Fwd Packet Length Mean,' 'PSH Flag Count,' and 'Packet Length Variance' from the 30 features.

The DENSE MLP model was trained using a preprocessed training dataset, and it was evaluated using the testing dataset. While training the model, it took 117.90 seconds and occupied 77.45 MB memory. The model evaluation metrics are accuracy: 99.42%, sensitivity: 99.93%, precision: 99.06%, specificity: 98.75%, F-score: 0.9949, and balanced accuracy: 99.34%. These metrics show that the proposed hybrid feature selection method helps to detect the DDoS attack traffic with lower feature space. The DENSE MLP model on the CICIDS2017 dataset outperformed the baseline eleven ML models. Table IV provides a detailed description of the DENSE MLP model evaluation metrics, while Fig. 5(c) and Fig. 6(c) display the confusion matrix and ROC curve.

4) Evaluation on CICIDS2018 Dataset:

The training set of the CICIDS2018 dataset, which consists of 79 features, is given as input to the proposed feature selection method. In the pre-processing phase, a total of 20 features (two socket, 8 redundant, and 10 invariant) were discarded, and then the Z-test was applied to 59 features. The Z-test identified 30 features as relevant. We then applied the chi-square method to the 30 features. The output of the feature selection was 15 significant features. The proposed feature selection method was evaluated on the binary class and ternary class.

a) Binary Class: For the binary class classification, target class labels were encoded as DDoS and BENIGN. The DENSE MLP model was trained using the pre-processed training dataset. While training the model, validation loss converged at the 14th epoch and restored the best weights. While training the model, it took 368.68 seconds and occupied 182.59 MB memory. The model's performance was evaluated using testing dataset. The evaluation metrics are accuracy: 100%, sensitivity: 100%, precision: 100%, specificity: 100%, F-score: 1.0, and balanced accuracy: 100%. These metrics are depicted in Table IV, ROC, and the confusion matrix in Fig. 5(d) and Fig. 6(d). The model demonstrated equal to or higher performance to eleven ML models, as depicted in Table V.

b) Ternary Class: The CICIDS2018 pre-processed dataset consists of low-rate and high-rate DDoS attacks; those are encoded, such as the DDOS attack-LOIC-UDP attack class being LOW and the DDOS attack-HOIC attack class being HIGH. The DENSE MLP model was trained using the encoded dataset. While training the model, the validation loss converged at the 14th epoch, and the loss curve is depicted in Fig. 7(a). The best weights of the model were restored and tested with the testing dataset. While training the model, it took 472.83 seconds and occupied 195.94 MB memory. It shows the evaluation metrics such as accuracy (100%), sensitivity (100%), precision (100%), specificity (100%), F-score (1.0), and balanced accuracy (100%). These metrics are depicted in Table VI, the ROC, and the confusion matrix in Fig. 8(e) and Fig. 9(e).

The DENSE MLP model showed better performance for detecting and classifying low-rate and high-rate DDoS attacks using significant features. Compared to the baseline 11 ML models, the DENSE MLP model shows better performance.

5) Evaluation on CICDDoS2019 Dataset:

The training dataset consists of 399998 samples with 87 features and was given to the proposed feature selection method. In the pre-processing phase, we discarded a total of 28 features (8 socket, 8 redundant, and 12 invariant). After pre-processing 59 features as input to the Z-test, 30 relevant features were selected from them. The chi-square test was applied on the 30 features, and 15 significant features were identified: 'ACK Flag Count', 'Fwd Packet Length Min', 'Average Packet Size', 'Min Packet Length', 'Flow Bytes/s', 'min_seg_size_forward', 'Fwd Header Length', 'URG Flag Count', 'Fwd IAT Total', 'Flow

Duration,Init_Win_bytes_forward,Inbound','Init_Win_bytes_backward', 'Idle Max', and 'Flow IAT Max'. This proposed method was evaluated on binary and ternary class classification over the three testing datasets (D1, D3, and D4).

a) Binary Class: For the binary class classification, training, and testing datasets, class labels were encoded as BENIGN for normal traffic and DDoS for attacks. The pre-processed training dataset and validation dataset are given as input to the DENSE MLP model. While training the model, the validation loss was converged at the 21st epoch, and it was depicted in Fig. 4(e). From there, the best weights were restored, and the model was tested using three testing datasets. While training the model, it took 377.25 seconds and used 193.58 MB memory. The evaluation metrics of the DENSE MLP model over the testing datasets were depicted in Table IV. The average model accuracy is 99.69%, precision is 99.71%, sensitivity is 99.80%, specificity is 99.71%, F-score is 0.9969, and balanced accuracy stands at 99.69%. ROC curves of the D1, D3, and D4 testing datasets are depicted in Fig. 5(e), Fig. 5(g), and Fig. 5(h) and the confusion matrices were depicted in Fig. 6(e), Fig. 6(g), and Fig. 6(h).

In Table V, we observed that the DENSE MLP model showed better performance than the 11 baseline ML models. The low-rate and high-rate DDoS detection results are depicted in Table VI. Table VII presents a comparison with existing feature selection methods, highlighting how the proposed approach performs against the latest techniques using the CICIDDoS2019 dataset. The proposed method outperformed the existing works of Rajagopal et al. [24], Raju et al. [26], Davrim et al. [31], S. Mahdavifar et al. [30], Thi-Thu-Huong Le [29], J. Halladay et al. [25], Pontes et al. [23], A. A.A. Najar et al. [32], and Aswani et al. [28]. When compared to the Raza et al. [28] works, its performance is 0.02% lower.

b) Ternary Class:

i) Low-Rate and High-Rate DDoS attacks classification: To evaluate the performance over the low-rate and high-rate DDoS attack classifications, the training class labels (DNS, LDAP, MSSQL, NetBIOS, SSDP, and UDP) were considered low-rate DDoS attacks and encoded as LOW. And the WebDDoS, NTP, SNMP, Syn, TFTP, and UDPLag class labels were considered as high-rate DDoS attacks and encoded as HIGH. As such, the testing dataset class labels were encoded, and the PortMap class was encoded as HIGH.

The DENSE MLP model was trained using the pre-processed training and validation datasets. The model validation loss converged at the 27th epoch while training, and its loss curve is depicted in Fig. 7(b). The DENSE MLP model was restored to the best weights, and it was tested using the three testing datasets. While training the model, it took 493.35 seconds and occupied 169.13 MB memory. The performance over the three testing datasets is depicted in Table VI. The ROC curves of the D1, D3, and D4 test datasets are depicted in Fig. 8(a), 8(c), and 8(d), respectively. And the confusion matrices are depicted in Fig. 9(a), 9(c), and 9(d), respectively. The average balanced accuracy of the model is 79.03%.

The proposed method was compared with existing IDS approaches and showed performance on par with the latest advancements like Rajagopal et al. [24], S. Mahdavifar et al.[30], Thi-Thu-Huong Le [29], J. Halladay et al. [25], and Aswani et al. [27]. When compared to Raza et al. [28], Raju et al. [26], Davrim et al. [31], Pontes et al. [23], and A.A. Najar et al. [32], the performance is lower.

ii) Reflection and Exploitation DDoS attacks classification: To assess how well the system identifies reflection and exploitation DDoS attacks, the attack types were labeled as REFLECTION for DNS, LDAP, MSSQL, NetBIOS, SSDP, WebDDoS, NTP, SNMP, TFTP, and PortMap, and as EXPLOITATION for Syn, UDP, and UDPLag, as noted by Sarafaldin et al. [23]. After labelling, the DENSE MLP model was trained with one dataset, while another dataset, D2, was used for validation. After the label encoding, the DENSE MLP model was trained using a training dataset, while training the dataset D2 was given for validation. The validation loss converged at the 11th epoch during training. The model was tested using D1, D3, and D4 test datasets, with an average of 89.44% balanced accuracy over the three datasets. Table VII depicts a comparison with existing works based on state-of-the-art performance. The proposed method's performance is better than all existing works except Raju et al. [26]. Though it is excellent at comparing with existing works, it shows lower performance while detecting the exploitation of DDoS attacks.

D. Key Findings

- The proposed hybrid feature selection reduced the feature space w.r.t. datasets NSL-KDD, BoT_IoT, CICIDS2017, CICIDS2018, and CICDDoS2019 by 73.80%, 80.72%, 81.01%, 81.01%, and 82.75%. The hybrid feature selection method reduced 80.13±2.38% of feature space over the five datasets.
- When the learning model is trained using selected features obtained using the proposed feature selection, the trained model outperforms w.r.t. CICIDS2017, CICIDS2018, and CICDDoS2019 test datasets.
- State-of-the-art comparison: The DDoS attacks detection performance is better than the recent existing works using feature selection methods.
- An average detection time for a single network traffic is recorded as less than a second; it is a more important attribute in IDS systems.
- Our DENSEMLP model, with 5 layers and less than 16 input features, is lightweight, and using a batch size of 1024 enables efficient vectorised and parallel computation on GPU. The model achieves sub-microsecond per-sample prediction while maintaining high detection accuracy, making it suitable for real-time deployment.
- To calculate the real-time inference of the models, we have run the prediction of one sample network flow for 100 times and calculated the average inference time. The average inference time of all models for each sample is depicted in Table VIII.

TABLE VII. STATE-OF-THE-ART COMPARISION WITH EXISTING WORKS OVER THE CICDDOS2019 DATASET

SL	Author & Year	Number of Features	Feature Selection Method	Classification		
				BINARY	LOW HIGH BENIGN	REFLECTION EXPLOITATION BENIGN
1	Sarafaldin et al. [22] & 2019	24	Feature Importance	66.38	93.80	71.88
				66.45	94.01	73.58
				66.13	94.02	73.09
				66.13	94.08	73.21
2	Pontes et al. [23] & 2021	82	Manually	98.72	82.18	82.41
				98.81	80.93	81.36
				98.84	81.30	83.41
				98.84	81.27	83.52
3	Rajagopal et al. [24] & 2021	15	MUI, Fisher Score, And Correlation	87.66	64.38	78.99
				88.13	61.10	79.39
				88.34	65.59	80.93
				88.38	65.62	80.95
4	J. Halladay et al. [25] & 2022	25	Manually	88.26	66.01	70.55
				88.70	63.80	69.32
				88.15	64.65	70.82
				88.20	64.71	70.96
5	Raju et al. [26] & 2022	5	Shapely	99.49	97.21	96.41
				99.51	97.59	96.12
				99.61	97.73	97.91
				99.62	97.74	97.93
6	Aswani et al. [27] & 2024	22	Manually	77.16	62.25	68.20
				77.45	55.88	70.28
				77.04	62.78	68.79
				77.05	62.84	68.90
7	Raza et al [28] & 2024	16	Correlation and Chi-square	99.77	85.96	81.31
				99.74	83.60	79.73
				99.80	85.46	81.58
				99.80	85.62	81.78
8	Thi-Thu-Huong Le[29], & 2025	5	MUI	77.03	76.14	54.44
				77.12	74.79	54.47
				77.05	75.29	54.47
				76.95	75.27	54.48
9	S. Mahdavifar et al. [30] & 2024	22	MUI	86.30	71.23	40.70
				87.56	67.63	41.11
				86.40	68.94	40.99
				86.40	68.92	40.98
10	Davrim et al. [31] & 2022	40	Information Gain	76.74	81.47	75.18
				76.77	80.64	75.06
				76.67	80.86	76.24
				76.67	80.85	76.26
11	A.A. Najar et al. [32] & 2024	43	Information Gain	97.76	85.22	75.43
				98.29	84.13	76.06
				98.31	84.06	76.08
				98.29	84.03	76.10
12	**Proposed**	**15**	**Z-test and Chi-Square**	**99.63**	**80.04**	**88.99**
				99.66	**78.87**	**88.13**
				99.72	**78.64**	**89.65**
				99.73	**78.60**	**89.69**

E. Practical Implications

The NIDS functions as a shield, protecting the network from DDoS attacks. While it operates in real-time, time complexity plays a crucial role. The optimized feature space, utilizing our proposed feature selection method, will help reduce the time complexity.

F. Interpretation

To address interpretability, we incorporated SHAP-based analysis to identify the top contributing features influencing the model's predictions. As shown in in Fig. 10, a global SHAP bar plot of CICDDoS2019 is added to illustrate how each feature impacts the model predictions. This provides clear evidence that the model relies on meaningful and domain-relevant features, thus strengthening the practical applicability of our method. The CICDDoS2019 dataset consist of various DDoS attack classes, the interpretability of most influencing features of this dataset will explore the other directions of research in terms of low-rate and high-rate DDoS attacks detection.

G. Limitations

The proposed method has shown excellent performance on CICIDS2017, CICIDS2018, and CICDDoS2019 datasets to detect DDoS attacks. Whereas, on the NSL-KDD and BoT_IoT datasets, performance is lower due to NSL-KDD not containing the DDoS attack samples. The proposed method was evaluated on the CICDDoS2019 dataset using nearly four lakh samples. The dataset size may be increased to enhance the DENSE MLP's ability to classify low-rate, high-rate, reflection, and exploitation DDoS attacks. In this paper, we focused on hybrid feature selection and developing the DNN. While, we didn't focused on model explainability. In our future work, will focus on model explainability.

Although the proposed DNN model incurs higher inference time than traditional classifiers, this behavior is typical due to the increased model depth and non-linear transformations. Despite this, the average inference latency of 40.4986 ms per sample remains sufficiently low for real-time deployment, where sub-second decision times are considered acceptable. Moreover, the improved detection performance offset the additional computational overhead, representing a high accuracy compare to rest of models.

V. Conclusion

While providing information technology services to legitimate users, organizations are struggling with DDoS attacks. Using an IDS system, this problem can be mitigated by monitoring the network and detecting DDoS attacks continuously. The IDS system should process the network data and give the results in less time than is required, which can be possible by selecting the significant features instead of using all the network features. Our hypothesis-driven feature selection method selected the significant features and reduced the feature space by 80.13±2.38% over the NSL-KDD, BoT-IoT, CICIDS2017, CICIDS2018, and CICDDoS2019 IDS datasets. The proposed method using the DENSE MLP model showed 99% to 100% DDoS detection performance on the CICIDS2017, CICIDS2018, and CICDDoS2019 datasets, whereas 75% to 85% on the NSL-KDD and BoT_IoT datasets. The optimized feature set also showed excellent performance in detecting the low-rate and high-rate DDoS attacks.

In real-time IDS systems, there is a trade-off between computational cost and detection accuracy, as complex models can introduce unacceptable latency. Our approach, using Z-test and chi-square feature selection with DENSEMLP, balances high detection accuracy with 40.4986 ms/sample latency, making it suitable for real-time deployment.

Moreover, the latency time of the DENSE MLP model for a single network traffic instance across the datasets NSL-KDD, BoT_IoT, CICIDS2017, CICIDS2018, and CICDDoS209 is 40.4986 ms when using the optimal features. In our work, we evaluated the balanced accuracy metric, which will give the model an unbiased detection performance. The DENSE MLP model showed better performances over the recent IDS datasets (CICIDS2017, CICIDS2018, and CICDDoS2019). In contrast, over the NSL-KDD and BoT-IoT datasets it showed lower performance. NSL-KDD contains only DoS samples and lacks DDoS instances, while BoT-IoT includes DDoS samples, but the traffic is limited to IoT environments. We concentrated on the hybrid feature selection method to optimize the feature space, rather than model explainability. The lack of real-time streaming or IoT-native experiments, where high-velocity traffic, constrained resources, and dynamic conditions may influence performance, limits our study. As future work, we plan to deploy the model in streaming frameworks (e.g., Kafka/Flink or MQTT-based testbeds) for continuous low-latency inference and validation under realistic IoT workloads.

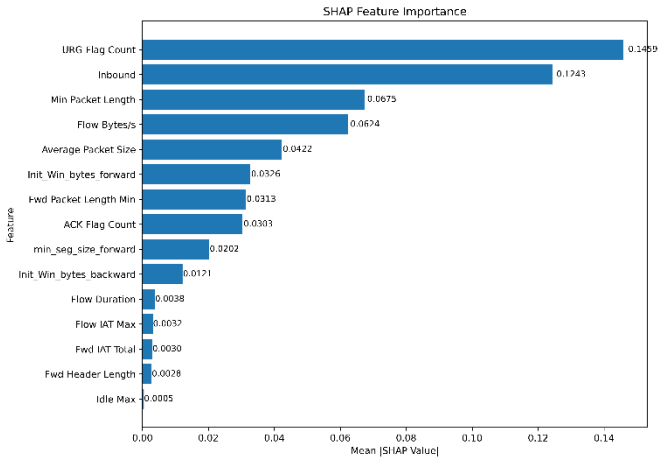

Fig. 10. Contribution of each feature for predicting the DDoS attack.

TABLE VIII. Proposed model inference time comparision with base line classifiers

Model	Model	Inference Time (ms/sample)	Average Accuracy (%)
1	ADB	4.572966	94.25
2	DT	0.439322	95.37
3	EXT	4.19616	97.80
4	KNN	1.42724	98.03
5	LDA	0.477624	89.61
6	LR	0.524046	99.31
7	NB	0.639834	91.27
8	QDA	0.626481	95.84
9	RF	3.830082	97.54
10	Ridge	0.661881	89.51
11	XGB	2.280502	98.42
12	**Proposed DENSEMLP**	40.4986	**99.69**

REFERENCES

[1] H. Chuang and L. Ye, "Applying transfer learning approaches for intrusion detection in software-defined networking", Sustainability, vol. 15, no. 12, pp. 9395, Jun. 2023.

[2] L. Chen, Z. Wang, R. Huo and T. Huang, "An adversarial DBN-LSTM method for detecting and defending against DDoS attacks in SDN environments", Algorithms, vol. 16, no. 4, pp. 197, Apr. 2023.

[3] M. Rashid, J. Kamuruzzaman, T. Iman, S. Wibow, and S. Gordan, "A tree-based stacking ensemble technique with feature selection for network intrusion detection", International Journal of Speech Technology, vol 52, no. 9, pp. 9768-9781, 2022.

[4] M. Bakro, R. Rakesh, M. Hussain, Z. Ashraf, A. Ali, S. Yaqoob, M. Ahmed,N.Parveen," Building a Cloud-IDS by Hybrid Bio-Inspired Feature Selection Algorithms Along With Random Forest Model", IEEE Access, vol 12, pp.8846-8874, 2024.

[5] A. Binbusayyis and T. Vayapuri, "Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and One-class SVM", International Journal of Speech Technology, Volume 51, No. 10, pp. 7094-7108, 2021.

[6] P. Kannari, N. Shariff, and R. Biradar, ''Network intrusion detection using sparse autoencoder with swish-PReLU activation model,'' J. Ambient Intell. Humanized Comput., Vol. 1, pp. 1-13, Mar. 2021, doi: 10.1007/s12652- 021-03077-0.

[7] H. Emiro, H. Eduardo, A. Ortiz, J. Ortega, and A. Martínez-Álvarez, "Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps", Knowledge-Based Systems, vol 71, pp. 322-338 2014, https://doi.org/10.1016/j.knosys.2014.08.013.

[8] Z. Ahmad, S. Khan, W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches", Transactions on Emerging Telecommunications Technology, vol 32, no. 1 pp. e4150, 2021, https://doi.org/10.1002/ett.4150.

[9] Ch. Khammassi and S. Krichen, "A NSGA2-LR wrapper approach for feature selection in network intrusion detection", Computer Networks, vol. 172, pp. 107183, 2020, https://doi.org/10.1016/j.comnet.2020.107183.

[10] Siddiqi, M. Ahmed, and W. Pak., "Optimizing Filter-Based Feature Selection Method Flow for Intrusion Detection System", Electronics, vol. 9, no. 12: pp. 2114, 2020, https://doi.org/10.3390/electronics9122114.

[11] Ahsan, Mostofa, G. Rahul, Md. Minhaz, and E. Kendall., "Enhancing Machine Learning Prediction in Cybersecurity Using Dynamic Feature Selector", Journal of Cybersecurity and Privacy, vol 1, no. 1, pp. 199-218, 2021, https://doi.org/10.3390/jcp1010011.

[12] Z. Halim, M. Nadeem, M. Waqas, M. Sulaiman, G. Abbas, M. Hussain, I. Ahmad, and M. Hanif, An effective genetic algorithm-based feature selection method for intrusion detection systems, Computers & Security, vol. 110, pp. 102448, 2021, https://doi.org/10.1016/j.cose.2021.102448.

[13] Y. Zhang, Sh. Cheng, Sh.Yuhui, D. Gong, and X. Zhao, "Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm", Expert Systems with Applications, vol. 137, pp. 46-58, 2019, https://doi.org/10.1016/j.eswa.2019.06.044.

[14] Alabdulwahab, Saleh, and BongKyo Moon. "Feature Selection Methods Simultaneously Improve the Detection Accuracy and Model Building Time of Machine Learning Classifiers", Symmetry, vol. 12, no. 9: pp. 1424, 2020, https://doi.org/10.3390/sym12091424.

[15] Akhiat, Y., Touchanti, K., Zinedine, A. et al. IDS-EFS: Ensemble feature selection-based method for intrusion detection system. Multimed Tools Applications, Volume 83, 2024, pp. 12917–12937, https://doi.org/10.1007/s11042-023-15977-8.

[16] I. Thaseen and C. Aswani Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM", Journal of King Saudi University, Comput. Inf. Sci., vol. 29, no. 4, pp. 462-472, Oct. 2017.

[17] S. Dwivedi, M. Vardhan, and S. Tripathi, ''Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection,'' Cluster Comput., vol. 24, no. 3, pp. 1881–1900, Sep. 2021, doi: 10.1007/s10586-020-03229-5.

[18] R. Kanna and P. Santhi, ''Hybrid intrusion detection using MapReduce based black widow optimized convolutional long short-term memory neural networks,'' Expert Syst. Appl., vol. 194, pp. 116545, May 2022, doi: 10.1016/j.eswa.2022.116545.

[19] V. Dora and V. Lakshmi, ''Optimal feature selection with CNNfeature learning for DDoS attack detection using meta-heuristic-based LSTM,'' Int. J. Intell. Robot. Appl., vol. 6, no. 2, pp. 323–349, Jun. 2022, doi: 10.1007/s41315-022-00224-4.

[20] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, ''AE-MLP: A hybrid deep learning approach for DDoS detection and classification,'' IEEE Access, vol. 9, pp. 146810–146821, 2021, doi: 10.1109/ACCESS.2021.3123791.

[21] P. R. Kanna and P. Santhi, ''Unified deep learning approach for efficient intrusion detection system using integrated spatial–temporal features,'' Knowledge. Based Systems., vol. 226, no. 107132, Aug. 2021, doi: 10.1016/j.knosys.2021.107132.

[22] I. Sharafaldin, A. H. Lashkari, S. Hakak and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," In Proc: International Carnahan Conference on Security Technology (ICCST), Chennai, India, 2019, pp. 1-8, doi: 10.1109/CCST.2019.8888419.

[23] Pontes, C. F. T., de Souza, M. M. C., Gondim, J. J. C., Bishop, M., and MA. Marotta, " A New Method for Flow-Based Network Intrusion Detection Using the Inverse Potts Model", IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp.1125–1136, 2021, doi:10.1109/tnsm.2021.3075503.

[24] S. Rajagopal, P. Kundapur, and K. Hareesha, "Towards effective network intrusion detection: from concept to creation on Azure cloud". IEEE Access, vol. 9, pp.19723-19742.

[25] J. Halladay, D. Cullen, N. Briner, J. Warren, K. Fye, and R. Basnet, "Detection and Characterization of DDoS Attacks Using Time-Based Features", IEEE Access, vol. 10, pp. 49794-49807, 2022, doi: 10.1109/ACCESS.2022.3173319.

[26] B.K. Raju, and H. Seetha, "An integrated approach explaining the detection of distributed denial of service attacks", Computer Networks, vol. 216, pp. 109269, 2022, https://doi.org/10.1016/j.comnet.2022.109269.

[27] A. Aswani, and E. Suresh, "A lightweight multi-vector DDoS detection framework for IoT-enabled mobile health informatics systems using deep learning", Information Sciences, vol 662, pp. 120209, 2024, https://doi.org/10.1016/j.ins.2024.120209.

[28] Raza, M. Saibtain, M. Nowsin, I. Hwang, and MS. Rahman, "Feature-Selection-Based DDoS Attack Detection Using AI Algorithms", Telecom vol. 5, no. 2, pp. 333-346. 2024, https://doi.org/10.3390/telecom5020017.

[29] Thi-Thu-Huong Le, S. Heo, J. Cho, H. Kim, "DDoSBERT: Fine-tuning variant text classification bidirectional encoder representations from transformers for DDoS detection", Computer Networks, vol. 262, pp. 111150, 2025, https://doi.org/10.1016/j.comnet.2025.111150.

[30] S. Mahdavifar and A. A. Ghorbani, "CapsRule: Explainable Deep Learning for Classifying Network Attacks," in IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 9, pp. 12434-12448, Sept. 2024, doi: 10.1109/TNNLS.2023.3262981.

[31] A. Devrim, S. Hizal, and U. Cavusoglu, "A new DDoS attacks intrusion detection model based on deep learning for cybersecurity", Computers and Security, vol. 118, pp.102748, 2022, https://doi.org/10.1016/j.cose.2022.102748.

[32] A. A. Najar, and S. M. Naik, "A Robust DDoS Intrusion Detection System Using Convolutional Neural Network", Computers and Electrical Engineering, vol. 117, no. 109277, pp. 1-19, 2024, https://doi.org/10.1016/j.compeleceng.2024.109277.

[33] M. Raghupathi and V. Radhakrishna, "Integrating Machine Learning and T-tests to Optimize Distributed Denial of Service Attacks Detection", International Journal of Intelligent and Engineering Systems, vol.17, no. 6, pp.1023-1043, 2024, https://doi.org/10.22266/ijies2024.1231.76.

[34] H. Zouhri, A. Idri, and Ratnani, "A. Evaluating the impact of filter-based feature selection in intrusion detection systems". International. Journal of. Information Security, vol 23, pp. 759–785, 2024. https://doi.org/10.1007/s10207-023-00767-y.

[35] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: modeling the costs of mislabeling," IEEE Access, vol. 8, pp. 4806–4813, 2020.

[36] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", In Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl., pp. 53-58, Jul. 2009.

[37] I. Sharafaldin, AH. Lashkari, and AA. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization". In Proc: International Conference on Information Systems Security and Privacy ISSN 2184-4356, pages 108-116. 2018, DOI: 10.5220/0006639801080116.

[38] A. Gharib, I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "An Evaluation Framework for Intrusion Detection Dataset," In Proc: International Conference on Information Science and Security (ICISS), Pattaya, Thailand, pp. 1-6,2016, doi: 10.1109/ICISSEC.2016.7885840.

[39] Koroniotis, Nickolaos, N. Moustafa, E. Sitnikova, and B. Turnbull. "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset.", Future Generation Computer Systems, vol. 100, pp. 779-796. 2019.

[40] AH. Lashkari, G. Draper-Gil, MS. Mamun and AA. Ghorbani, "Characterization of Tor Traffic Using Time Based Features", In Proc: International Conference on Information System Security and Privacy, SCITEPRESS, Porto, Portugal, 2017.

[41] Gerard Drapper Gil, Arash Habibi Lashkari, Mohammad Mamun, Ali A. Ghorbani, Characterization of Encrypted and VPN Traffic Using Time-Related Features", in Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016),Italy, pp.407-414, 2016.

[42] Himanshu and . H. . Singh Arri, "A Proposed Method for Real-Time Automatic Cloud Storage and Analysis of Detected Suspicious Activities to Ensure Data Integrity and Security", JASTT, vol. 6, no. 2, pp. 262–276, Sep. 2025.

[43] A. Dave, "Intelligent Resource Management and Secure Live Migration in Cloud Environments: A Unified Approach using Particle Swarm Optimization, Machine Learning, and Blockchain on XenServer", JASTT, vol. 6, no. 2, pp. 393–407, Nov. 2025.

ABBREVIATIONS

ADB	AdaBoost
DT	Decision Tree
EXT	Extra Tree
FS	Feature Selection
KNN	K Nearest Neighbour
LDA	Linear Discriminant Analysis
LR	Logistic Regression
NB	Naïve Bayes
QDA	Quadratic Discriminant Analysis
RF	Random Forest
ReLU	Rectified Linear Unit
XGB	Extended Gradient Boosting