Ensemble-RNN: A Robust Framework for DDoS Detection in Cloud Environment

Asha Varma Songa[1], Ganesh Redy Karri[2]
1- VIT-AP University, School of Computer Science and Engineering, Near Vijayawada, Andhra Pradesh, India.
Email: ashavarma.20phd7123@vitap.ac.in, (Corresponding Author)
2- VIT-AP University, School of Computer Science and Engineering, Near Vijayawada, Andhra Pradesh, India.
Email: guncity11@gmail.com

Received: 5 August 2023 Revised: 14 September 2023 Accepted: 18 October 2023

ABSTRACT:
The advent of cloud computing has made it simpler for users to gain access to data regardless of their physical location. It works for as long as they have access to the internet through an approach where the users pay based on how they use these resources in a model referred to as "pay-as-per-usage". Despite all these advantages, cloud computing has its shortcomings. The biggest concern today is the security risks associated with the cloud. One of the biggest problems that might arise with cloud services availability is Distributed Denial of Service attacks (DDoS). DDoS attacks work by multiple machines attacking the user by sending packets with large data overhead. Therefore, the network is overwhelmed with unwanted traffic. This paper proposes an intrusion detection framework using Ensemble feature selection with RNN (ERNN) to tackle the problem at hand. It combines an Ensemble of multiple Machine Learning (ML) algorithms with a Recurrent Neural Network (RNN). The framework aims to address the issue by selecting the most relevant features using the ensemble of six ML algorithms. These selected features are then used to classify the network traffic as either normal or attack, employing RNN. The effectiveness of the proposed model is evaluated using the CICDDoS2019 dataset, which contains new types of attacks. To assess the performance of the model, metrics like precision, accuracy, F-1 score, and recall are taken into consideration.

KEYWORDS: Cloud Computing, DDoS Attacks, Machine Learning, Deep Learning Techniques.

1. INTRODUCTION

Cloud computing has gained popularity due to its various features, such as on-demand service provision and affordability. Cloud computing is an internet-based platform that offers individuals and businesses a broad range of computing services, including networking and databases [1]. Despite having multiple advantages, the shared environment that cloud computing offers may result in threats in terms of security and the availability of its services. The Service Legal Agreement (SLA) requires a cloud service provider (CSP) to guarantee that users have access to resources and security to uphold their obligation to them. The popularity of cloud computing has continued to increase as more people and companies continue to incorporate it into their businesses. However, despite its high utilization, security remains a serious concern in cloud computing [2]. Some of the popular cloud features are virtualization and multi-tenancy [3]. The cloud environment has many customers sharing physical resources, and thus there occurs a challenge in keeping its environment secure. Risks of sharing data in the cloud include the potential for consumer data to be lost or used improperly by other parties. Several types of cyber-attacks on cloud security also occur due to vulnerabilities in the system and application, such as malicious insiders and data loss. These could harm the availability, confidentiality, and integrity of data [4][5]. There are several Communication Service Providers (CSP) that offer cloud services. The unavailability of cloud services heavily impacts CSP and cloud clients.

One of the key risks that lead to cloud unavailability is Distributed Denial of Service(DDoS). Attackers use this attack, to block users from accessing services used by real users in cloud computing. This kind of attack puts a high load on the victim server by feeding it with multiple requests. These numerous requests overwhelm the vulnerable server's bandwidth as a result, rendering it inaccessible to authorized users [6]. A sample DDoS attack scenario is presented in Fig. 1. This attack affects the network devices with malware by using a botnet. DDoS attacks are based on their target and behavior and are classified into three main categories: bandwidth, application [7], and protocol attacks. These attacks pose a severe danger to the security of numerous environments. In the first quarter of 2021, the ATLAS

Paper type: Research paper
DOI: 10.30486/mjee.2023.1986487.1137
How to cite this paper: A. V. Songa, and G. R. Karri, **"Ensemble-RNN: A Robust Framework for DDoS Detection in Cloud Environment"**, *Majlesi Journal of Electrical Engineering*, Vol. 17, No. 4, pp. 31-44, 2023.

Security Engineering & Response Team of NETSCOUT received reports of almost 2.9 million DDoS attacks, a 31% increase from the same time in 2020 [8].

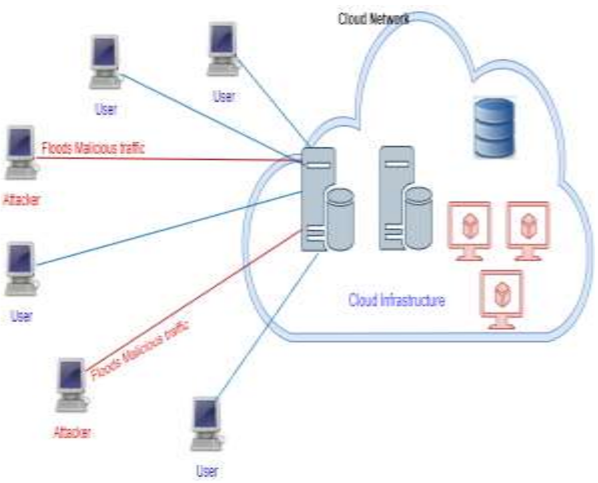

Fig. 1. Attack Scenario of DDoS.

Before any strategies are employed to mitigate DDoS attacks, these attacks must be detected first. Early detection of DDoS attacks will substantially reduce the financial loss and damage to the resources of the cloud. Intrusion Detection Systems (IDS) can detect and eliminate harmful activity in a network [9]. As DDoS attacks continue to increase with new distributed patterns, the detection systems also become vulnerable, causing more havoc in preventing such attacks in cloud space. Existing IDS have been inefficient in detecting attacks directed toward them, including reducing false alarm rates and zero-day attacks. However, with the developing and changing pattern of DDoS attacks, this strategy seems to have fallen behind. To enhance intrusion detection systems in cloud computing, an innovative approach needs to be developed that effectively monitors network traffic and identifies network anomalies.

This research is mainly focused on anomaly-based network intrusion detection systems. This IDS is developed with a combination of Machine Learning (ML) and Deep Learning (DL) techniques [10]. DL techniques heavily depend on feature engineering and are good at learning complex features automatically from raw data which is attributed to their deep structure. However, deep learning models have some considerations and challenges. They require a substantial quantity of labeled training data to achieve good performance, and the training process can be computationally intensive. Another challenge is overfitting, where the model becomes too specialized for the training data and performs poorly on unseen data. Hence, learning from raw data may increase the detection time and the possibility of increased false alarm rates. So, to increase the accuracy, and reduce the false alarm rates, and computational complexity, it is necessary to have the reduced feature set that is used for training the DL model. On the other hand ML approaches have a dependency on feature engineering to learn important information from network traffic [11]. However, with the growing network traffic, it is required to filter the relevant features for effective detection of DDoS attacks which increases the detection accuracy [12]. In recent intrusion detection studies, ensemble machine learning methods have emerged as valuable tools for achieving higher accuracy. The primary objective of ensemble methods is to leverage multiple machine learning models and combine their outputs to create a new, more robust model [13]. This approach aims to overcome the limitations of individual models and enhance the overall performance of an intrusion detection system (IDS). By combining the strengths of different algorithms, ensemble methods can effectively capture diverse patterns and behaviors associated with various types of attacks [14]. Although there are many existing models using ensemble methods, this paper uses six different ML techniques that are formed into four ensemble groups to select the top-weighted features is the novelty of our work.

Various academics are studying the possibility and effectiveness of implementing DL techniques in classifying DDoS attacks [15]. Training deep learning models for classification involves a forward pass, where input data propagates through the network, and a backward pass or backpropagation, where the model adjusts parameters based on computed error. This iterative process, with large-scale datasets and optimization algorithms, enables accurate predictions. One commonly used deep learning model for DDoS attack classification is the Recurrent Neural Network (RNN), specifically the Long Short-Term Memory (LSTM) variant. RNNs are suitable for analyzing sequential data, making them well-suited for capturing temporal patterns in network traffic [16].

Therefore, this proposed work is a combination of ML and DL techniques that improve the detection accuracy of DDoS attacks using ensemble feature selection and classification methods. Both techniques fall under artificial intelligence, aiming to make sense of information from big data. The development of potential Graphics Processing Units (GPUs) has increased their popularity in network security over time. They are powerful tools that can be used to learn meaningful features from the network traffic. They are also crucial for making predictions about normal and abnormal behavior based on patterns identified.

Therefore, the overall objective of this Ensemble RNN (ERNN) framework is to detect DDoS attacks in the cloud. The novelty of this paper lies in its implementation approach, which involves the selection of crucial features from an ensemble of multiple groups of machine learning techniques. These selected features are then classified using an RNN to determine if they represent normal behavior or an attack. The evaluation of this approach is conducted on the CICDDoS2019 dataset.

The contributions of this paper are

- We propose a novel ERNN IDS framework with ML-based ensemble multi-group feature selection and DL-based classification for early detection of DDoS attacks.
- Proposed an ensemble feature engineering model with six ML techniques which are formed into 4 ensemble groups for selecting top-weighted features for the detection of attacks.
- Build an RNN model to classify the network traffic with a smaller feature subset obtained from an ensemble feature selection while lowering false alarm rates.
- Evaluated this model on the CICDDoS2019 dataset and various metrics.
- We evaluate our proposed work by comparing it with state-of-the-art techniques.

The document is divided into five sections. Section 2 details related work. Section 3 deals with the proposed methodology. Section 4 presents the experiment Evaluation. Finally, section 5 concludes with a conclusion.

2. RELATED WORK

Combining several different feature selection methods into an "ensemble" can increase the reliability and consistency of the final feature set. An ensemble-based model can help surmount the drawbacks of a single feature selection model and produce more accurate and reliable results. The study in [17] integrates feature engineering and machine learning at a strategic level within a specified experimentation flow. The framework's effectiveness is validated through cross-validation and area-under-curve evaluations. The model is tested on different datasets where it achieved an accuracy of 93.5%. The author [18] proposed a lightweight deep-learning DDoS detection system. This paper uses CNN for classifying the traffic flows as normal or DDoS. The work is evaluated on 3 datasets and achieved high accuracy and a 40x reduction in processing time as compared to earlier work. In [19] The author offers a framework that involves three distinct feature selection algorithms that determine aspects that are vital for the accurate identification of malicious or suspicious DNS domains. These elements are identified in the framework as "essential features." putting out a methodology for the classification of bagging ensembles that identifies malicious internet addresses with a high degree of precision. In [20] the author proposed a Whale optimization (WO) with a DNN model for DDoS detection in cloud storage applications. The model selects the optimal features using WOA and feeds them to the DNN classifier for classifying normal or attack traffic. The model is tested using the CICIDS2017 dataset and secured an accuracy of 95.35% in classifying normal or DDoS data. In [21] the author proposed a hybrid DL (CNN-BiLSTM) model for DDoS detection in the cloud. The hybrid model uses CNN for feature selection and BiLSTM for the classification of a dataset that achieved an accuracy of 94.52%. The paper [22] proposes a new ensemble-based IDS to detect attack patterns using machine learning techniques. The effectiveness of the model is evaluated using the CICIDS 2017 dataset which resulted in good accuracy of 88.96% and 88.92% for multi and binary class classification scenarios. The author [23] proposed an ensemble voting algorithm for network intrusion detection the model selects the features from an ensemble of different feature selection techniques. These features are fed as input to different ML classifiers that achieve good accuracy in detecting DDoS attacks. The model is tested on 3 different datasets for its effectiveness. This paper [24] proposes an optimized ensemble feature selection method using super and unsupervised methods. This model is compared to 15 individual feature selection methods that achieved 76% accuracy in detecting DDoS attacks. In [25] the author proposed an LSTM model for the detection of DDoS attacks in fog computing. The model achieved good accuracy when tested on the ISCX dataset. The author in [26] proposed an innovative approach called the Adaptive Ensemble Random Fuzzy Algorithm has been proposed for anomaly detection in cloud computing. This algorithm incorporates random sample selection and a weighting strategy to enhance the accuracy of fuzzy classifiers in detecting anomalies. By randomly selecting samples, the algorithm ensures a diverse representation of the dataset, enabling a more robust anomaly detection process. Additionally, the application of a weighting strategy to the fuzzy classifiers further improves their performance, resulting in more accurate anomaly detection outcomes. Overall, this proposed algorithm offers an adaptive and effective solution for anomaly detection in cloud computing environments. This paper [27] proposes an intelligent DDoS attack detection model using an enhanced Gini index feature selection method and DT algorithm for classification. The model achieves 98% accuracy on the UNSW-NB15 dataset, outperforming advanced algorithms like Random Forest and XGBoost. In the paper [28] the proposed DDoS detection model uses a Bird swarm optimization (BSO) algorithm for feature selection and a DL classifier for the classification of normal and DDoS attacks. The BSO algorithm selects the optimal features from the traffic and these features are passed to different

DL classifiers where it achieved a detection accuracy of 98.9%. The authors [29] introduced two novel algorithms: the packet scrutinization algorithm and the hybrid classification model called Normalized K-means RNN (NKRNN). They also proposed a one-time signature for cloud user authentication to enhance security against attackers. The related work is tabulated in Table 1.

Table 1. Comparison Of Related Work.

Author	Year	Model	Technique	Datasets	Accuracy	Strength	Limitations
Aamir et al. [17]	2019	Feature engineering and ML	ML	Different datasets	93.5-average	High reliability in detection	Unable to detect novel attacks and accuracy can be improved further.
Doriguzzi et al. [18]	2020	LUCID	DL	CIC2017 CES2018	High accuracy	It boosts system storage while decreasing computational complexity.	Only one feature selection method is used
Moubayed et. al [19]	2020	Bagging ensemble classification	ML techniques	Generated	86.7%	The error in modeling was minimal.	Accuracy can be still improved. DNS typo squatting improved
Agarwal et.al [20]	2021	Whale optimization with DNN	DL	CICDDoS2017	95.35%	Its identification rate was high, while its error rate was low.	Could not detect novel attacks
Alghazzawi et al. [21]	2021	CNN-BiLSTM	DL	CICDDoS2019	94.52%	Deep feature extraction is done successfully.	Only one technique is used to select important features
Abbas et al [22]	2021	Ensemble feature selection	ML algorithms	CICIDS2017	88.96%	The model could extract the optimal subset of features	Can achieve still higher accuracy. Cannot detect novel attacks
Krishnaveni et. al. [23]	2021	Ensemble Majority voting	ML classifiers	NSL-KDD Honeypot Kyoto	High accuracy	High detection accuracy	Cannot work for novel datasets.
Saha et al [24]	2022	Optimized Ensemble feature selection	ML DL UL	UNSW-NB15	87.2%	The model's F1 score was 98%, while successfully reducing the false negatives' percentage to less than 1.8%.	Not accurate for novel datasets.
Priyadarshini et al [25]	2022	LSTM	DL	ISCX-12	98.88	High detection accuracy with minimal false alarms	Not accurate for novel datasets
Jun Jiang et al [26]	2023	adaptive ensemble random fuzzy	ML	EMOS cloud dataset	94.79%	High detection rate	cannot detect novel attacks
Bouke et al.[27]	2023	Tree based model	ML	UNSW-NB!5	98%	High precision and low error rates.	Used one FS technique and cannot detect novel DDoS attacks.
Abosuliman et.al.[28]	2023	BSO	DL	PCAP files	98.9%	High ability to detect	Unable to detect novel attacks.
Our Work(ERNN)	2023	ERNN ensembles multiple groups of ML algorithms for feature selection and uses RNN for classification.	ML DL	CICDDoS2019	As multiple ML methods are used, the most relevant features were selected which helped the RNN classifier for detecting DDoS attacks with high detection accuracy. This model can detect novel attacks with less computation time and reduced false alarms.		

From the above table, it is clear that few techniques used single feature selection methods, and a few used either DL or ML techniques in their papers. The combination of ML and DL is used in only one article by Saha et, al.[24] but it was implemented on an obsolete dataset rather than a novel dataset CICDDos2019. The accuracy achieved was also less. So, none of the techniques have the combination of ML, DL, and

CICDDoS 2019 datasets for the detection of DDoS attacks in the cloud. In conclusion, this research has revealed that numerous studies have been conducted on the subject of identifying DDoS attacks. However, the question of the best algorithm to use to solve this problem is still open. The significance of feature selection, which can effectively decrease computational time in detecting DDoS attacks, remains largely unexplored in numerous studies. Many studies have not explored this question using the CICDDoS 2019 dataset. To address these issues, this study proposed an ERNN-based IDS framework that achieves a higher attack detection accuracy and uses less computational power than its competitors. This carries the discussion into the upcoming sections.

3. ERNN FRAMEWORK

This section describes the methodology of the proposed ERNN framework and the techniques used in it. An ERNN is a three-phase framework that includes data preprocessing, ensemble feature selection, and classification. The processing flowchart and the architecture of the ERNN framework are illustrated in Fig. 2 and Fig. 3. Each phase is explained below separately.

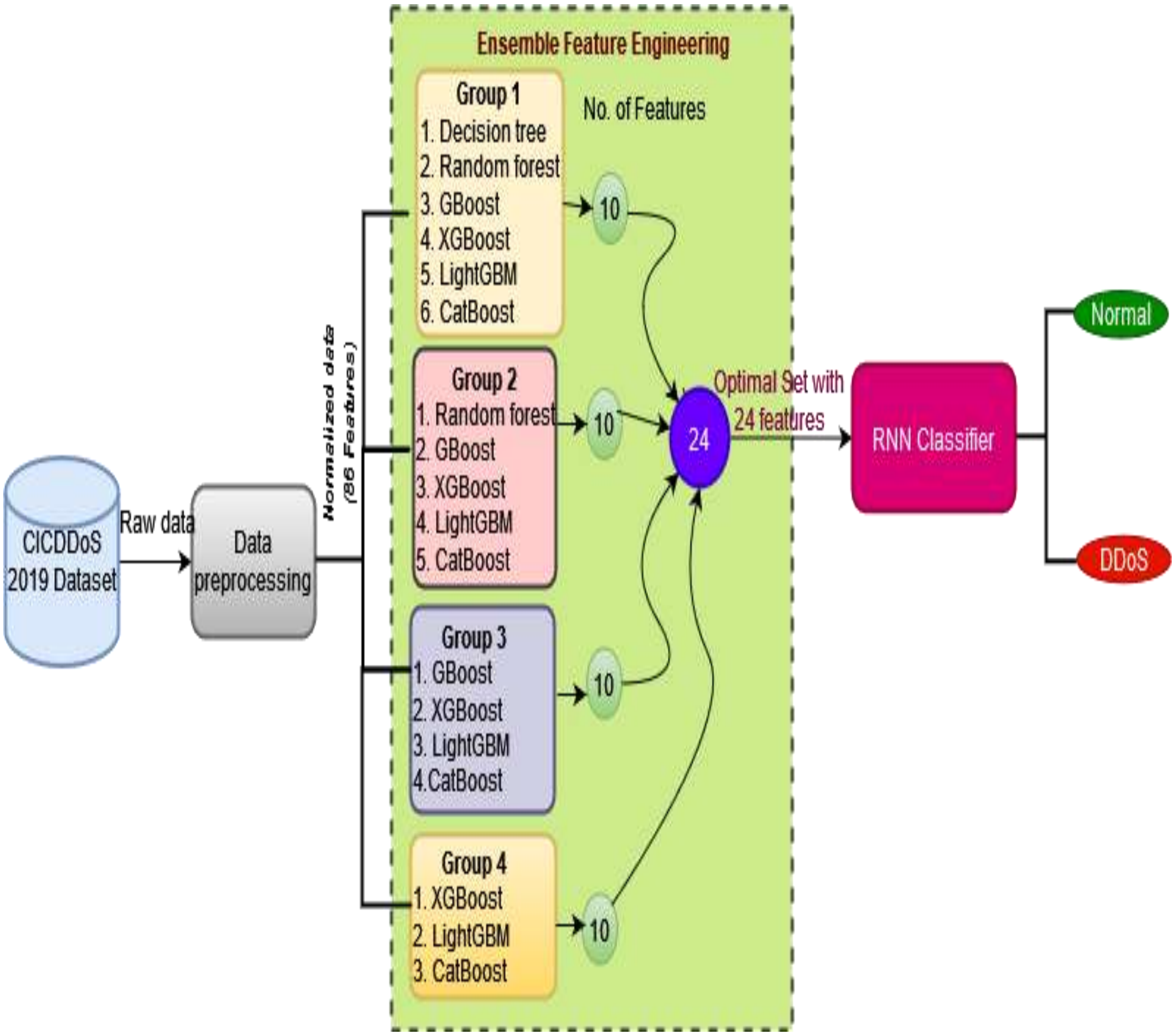

Fig. 2. ERNN Framework.

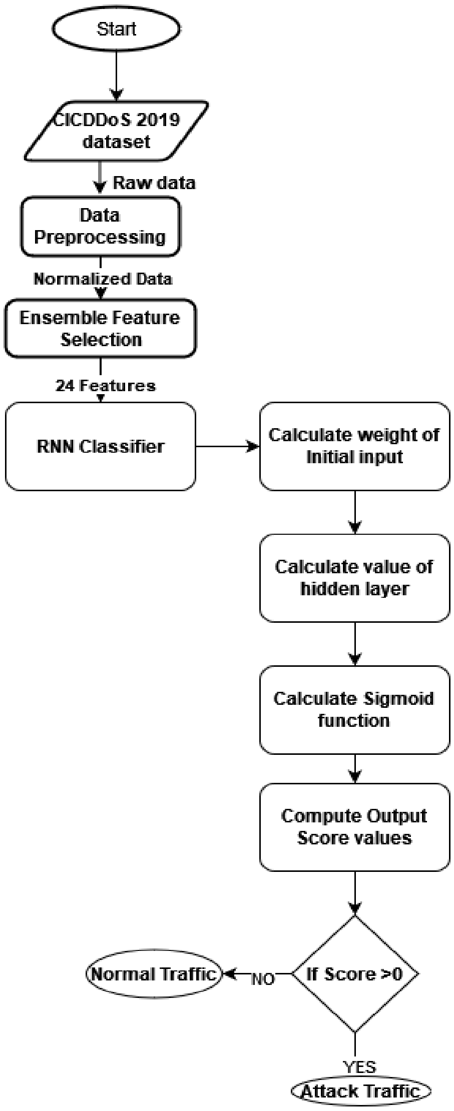
Start
CICDDoS 2019 dataset
Raw data
Data Preprocessing
Normalized Data
Ensemble Feature Selection
24 Features
RNN Classifier
Calculate weight of Initial input
Calculate value of hidden layer
Calculate Sigmoid function
Compute Output Score values
If Score >0
NO
Normal Traffic
YES
Attack Traffic

Fig. 3. Flow chart of Proposed Work.

3.1. Description of Dataset.

Before discussing the workflow of the ERNN let us study the CICDDoS2019 dataset used in this research. The production and availability of this dataset were made possible by the University of New Brunswick and the Canadian Institute of Cybersecurity [30]. The dataset constitutes the most recent benign and common DDoS attacks which also resemble real-world data. More than this, it contains CICFlowMeter-network V3's traffic analysis results. The flows in the Flowmeter are labeled based on the source, timestamp, source and destination ports, protocols, and attack. Multiple contemporary reflective DDoS attacks, including MSSQL, LDAP, SYN, PortMap, and UDP-Lag, are included in this dataset.

The dataset has a high level of imbalance. There is a high malicious traffic flow in comparison to normal traffic. This study uses an under-sampling technique that randomly omits data from the dominant class. This is done to guarantee the objectivity of learning and categorization. Additionally, the dataset ensures that appropriate features are selected from the input traffic patterns by the learning model. ERNN classifies incoming data as normal data or DDoS data using binary classification. For training and validation purposes, this classification algorithm splits the dataset into two halves, 80/20. However, it uses a random method to choose the validation samples. Finally, since the model cannot access the test dataset during the learning process, the model's performance is evaluated to make sure that the detection rate is accurate.

3.2. Data Preprocessing

This initial phase uses the IP traffic data as outputs and clean data as inputs. This step is essential in ML. This is due to the possibility that it will significantly increase the training process efficacy and efficiency. Additionally, it consists of essential tasks such as handling missing values, removing irrelevant data, converting labels, categorizing, and data normalization.

- **Omitting unnecessary Features:** Data preprocessing begins by getting rid of unnecessary features. Some of these unnecessary features include "inbound" and "flow id," as they don't significantly improve the issue at hand. In contrast, features attached to feature id include information about IP addresses and port numbers, which may lead the model to overfit. As a result, just the necessary features are retained, while the unnecessary ones are manually eliminated. Handling the missing values comes after determining pertinent features.
- **Handling Missing Values:** There are various methods for dealing with missing values. The suggested method substitutes the corresponding feature median for missing data. This is because it better represents the feature's predominant value. Another justification is that when an outlier is present in the data, the median provides an estimation.
- **Label Conversion:** In this study, network traffic is divided into two categories: normal traffic and DDoS attacks. Integer category indexes are created for discrete string value columns. The discrete variable is processed using category embedding. Variable embedding can be used to improve how these variables are processed by neural networks. These variables are more advantageous than one hot encoding because they require less memory and operate more quickly. When dealing with anonymous statistics, Entity

embedding facilitates the generalization of sparse data within a neural network model. The model employed in this study treats every number column with a separate level below 9,000 as a categorical variable.

- **Normalization:** Constant values in the dataset exhibit quite a wide range of variations, which increases prediction error. Dataset normalization is essential as it helps to reduce these classification errors. Additionally, it allows the model to converge at a higher rate. The standardization method, which scales the features to have a standard deviation of 1 and a mean of 0, was utilized in this study. As a result, the model can be less sensitive to outliers.

Algorithm 1: Proposed Methodology

```
Input:  Preprocessed dataset  $s^0$ : ($s_1$, $s_2$, $s_3$, .....$s_n$)
        Feature ranking algorithms F= {$f_1$,....., $f_k$}
        Ensemble groups X ---> ($X_1$, $X_2$, $X_3$, $X_4$) where each
group has a set of 6 feature ranking Algorithms from F
Step 1: For $X_i$ in 1, 2, 3, 4 do
            Assign the $s^0$ to F
            For i = 1......k do
                    Apply ranking algorithm to $F_i(s^0)$
                    Assign weights to features
                    Update feature set $s^0$ to $s^1$
            Endfor
            Select the top 10 features based on the
threshold
            Update feature set $s^1$ to $s^2$
        Endfor
Step 2:  /* training phase*/
        model =RNNclassifier($s^2$)
        Save the Model.
Step 3: /* testing phase*/
        model = load RNN classifier()
        model. predict($s^2$)
Step 4: Report the Performance Metrics
```

3.3. Feature Engineering

In this phase, we adopted an ensemble-based feature engineering approach that leveraged several powerful algorithms to enhance the effectiveness of our feature selection process [31]. Specifically, we utilized machine learning techniques namely decision trees (DT), extreme gradient boosting (XGBoost), random forest (RF), catboost, gradient boost(GB), and light gradient boosted machines (LightGBM) for ensemble feature selection [32]. By combining the strengths of these algorithms, we aimed to identify the most relevant and informative features for our analysis. Each algorithm brought its unique advantages, such as decision trees' interpretability, XGBoost's gradient boosting capabilities, random forest's ensemble nature, catboost's handling of categorical variables, gradient boost's iterative training, and LightGBM's efficiency in handling large datasets. These above-mentioned algorithms rank the importance of each feature in the dataset. This step helped us understand the relevance and contribution of each feature towards the target variable. We then created an ensemble by combining the outputs of the feature importance rankings from all the algorithms used. This allowed us to leverage the diversity and strengths of each algorithm to identify the most influential features. From the ensemble, we selected the top-ranked features based on their importance scores.

According to step 1 of Algorithm. 1, after the preprocessing phase the normalized dataset is the input to the ensemble feature selection model. To identify the influential features for our analysis, we employed six distinct techniques for feature engineering, which were organized into four groups each consisting of a different number of ranking algorithms ranging from three to six. These groups were denoted as "ensemble group x," with x representing the specific group number (1, 2, 3, or 4). The groups are shown in Table 2 respectively. Within each group, we selected the top 10 features based on their performance and relevance. Following the selection process in each group, we gathered a list of 10 features. To finalize the feature selection, we extracted the unique features from all four groups. This step ensured that we considered only the most distinctive and informative features, eliminating any duplicates that may have appeared across different groups.

By combining the top 10 features from each group and selecting the unique ones, we obtained a final set of features that captured the best attributes from our ensemble of techniques. This approach allowed us to leverage the strengths of multiple techniques while emphasizing the most valuable and distinctive features for our analysis.

Table 2. Ensemble Group List

Ensemble group	Feature ranking
1	DT
	RF
	GB
	XGB
	lightGBM
	catboost
2	RF
	GB
	XGB
	lightGBM
	Catboost

	GB
	XGB
	lightGBM
	catboost
4	XGB
	lightGBM
	catboost

The overall feature list selected from each ensemble group through the top 10 weight approach is tabulated in the following Table 3. A total of 24 features were selected which were sent as input to the RNN model for the training phase.

Table 3. Features selected per Ensemble group.

SNO	Feature List
1	Minimum packet length
2	Init Win bytes forward
3	ACK Flag Count
4	Forward Packet Length Min
5	Average packet size
6	Forward packet length max
7	Flow IAT Maximum
8	Flow packets/s
9	Init Win bytes backward
10	flow IAT Minimum
11	Init Win bytes forward
12	Forward packet length minimum
13	Forward packet length mean
14	Average forward segment size
15	The total backward packets
16	The total length of forward packets
17	Backward packets/s
18	Flow duration
19	Minimum segment size forward
20	flow bytes/s
21	ACK flag count
22	Flow bytes/s
23	Backward IAT total
24	Flow IAT Standard

3.4. Recurrent Neural Network Architecture

Following the feature selection comes the classification phase where an RNN technique is used to classify the data. It is a particular neural network used for categorization and detection. RNNs are a special kind of neural network that has been put to work sequentially managing data. The primary goal of RNN is to use the logical information from previous timestamps to forecast the name of the current timestamp. The architecture of RNN is depicted in above Fig. 4. The RNN model includes input, output, and two hidden layers where weight adjustments are made to generate the outputs. The weights among hidden layers are adjusted based on a comparison of the errors from the current and prior hidden layers [33] [34]. Regularized neural networks learn time series via two gradient-based methods. One is Back Propagation Through Time (BPTT) and the other one is Real-Time Recurrent Learning (RTRL). In our model, the RNN is trained with the BPTT that adjusts a neural network's weights to improve output accuracy compared to desired results for related inputs.

According to the Algorithm. 1 (Step 2), we begin by moving the chosen properties to the input layer and assigning weights to each of them in below Eq. (1) and Eq.(2)

$$\mathrm{f_i}\,(\mathrm{t}) = \sum\nolimits_{\mathrm{j}} \mathrm{p_j}(\mathrm{t})\mathrm{M_{ij}}(\mathrm{t}) \tag{1}$$

$$p_j = a_i(f_i(t)) \tag{2}$$

Where p_j, is the activation state of a neuron at time t and $M_{ij}(t)$ are the values of the weights that are tuned to provide the best results. The inputs to the activation function $\mathrm{f_i}$ rely on the network and other background layers. From Eq. (3) the prediction is made by the output layer using the sigmoid function.

$$a_{i=}\frac{1}{1+e^{-f_i}}. \tag{3}$$

Using the below-given equations Eq.(4) and Eq.(5) each neuron's output is weighed in the backward process of backpropagation.

$$p_i(t) = a_i(f_i(t), N_i(t)) \tag{4}$$

$$\mathrm{f_i}\,(\mathrm{t}) = \sum\nolimits_{\mathrm{j}\in\mathrm{H}} \mathrm{p_j}(\mathrm{t})\mathrm{M_{ij}} + \sum\nolimits_{\mathrm{j}\in\mathrm{F}} \mathrm{f_j}(\mathrm{t})\mathrm{M_{ij}} + \sum\nolimits_{\mathrm{j}\in\mathrm{N}} \mathrm{p_j}\left(\mathrm{t} - \mathrm{T_{ij}}\right)\mathrm{M_{ij}} \tag{5}$$

Where N_i is the information stored in the final step of the network's neuronal values, F is the input neurons, and H is the values of hidden layers. T is a positive integer that represents the recurrence's displacement. To calculate the difference between the actual and projected value, one uses the loss function which is given in Eq.(6). The accuracy of the model is higher with a lower loss; it is lower with a larger loss. The loss is minimized using the Bayesian regulation method. Finally, Eq.(7) is used for updating the weights.

$$L = P^{pre} - P^{act} \tag{6}$$

$$R = \delta\, L_d + \gamma L_w \tag{7}$$

During the testing process (step 3), the trained matching RNN weights are used. At last, the score value is determined which helps to classify whether the test data is normal or attacked which is shown in Eq.(8).

$$Final\ score = \{\ score \leq 0,\quad Traffic\ is\ normal;\quad score > 0,\quad Traffic\ \ is\ Attack \qquad (8)$$

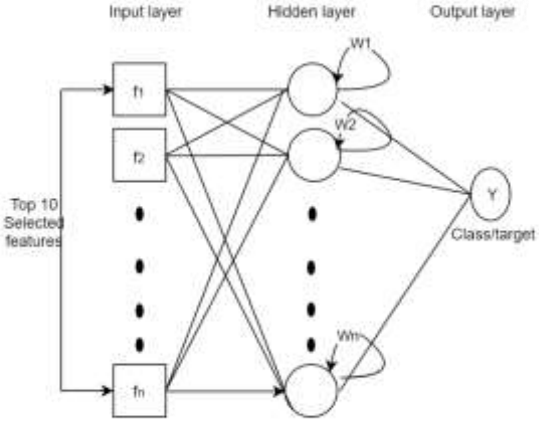

Fig. 4. Architecture of RNN.

4. EXPERIMENTAL RESULTS

This study performed tests on an Intel-based system to evaluate the suggested system. The computer ran Ubuntu 18.04-LTS and had a Core (TM) i9-9900X CPU clocked at 3.50 GHz and 128GB of RAM. Additionally, the host system has a Nvidia Titan RTX GPU with 24 GB of RAM. The experiment was conducted ten times to do away with any doubts. Additionally, PyTorch and Fastai were applied in this experiment. Facebook's AI Research Lab developed PyTorch, an open-source ML framework. A part of the framework is an optimized tensor.

Both CPUs and GPUs can use this tensor. To include cutting-edge features that are essential for creating DL models, Fastai is a framework that is created on top of PyTorch. Fastai is primarily used to offer the research community a very effective and user-friendly abstract framework. Additionally, it keeps the low-level parts to ensure flexibility while creating DL-based systems.

To initiate the training process, an RNN (Recurrent Neural Network) is constructed, utilizing the 24 features as input for training the model. To determine the proper values for different parameters, more experiments are conducted. The best results were obtained using two hidden layers with 200 and 100 neurons each, as well as 1024 batch-size samples after the model was trained for five epochs in 10 seconds. Fig. 5 below displays the distribution of traffic flows for the dataset's training, validation, and testing samples.

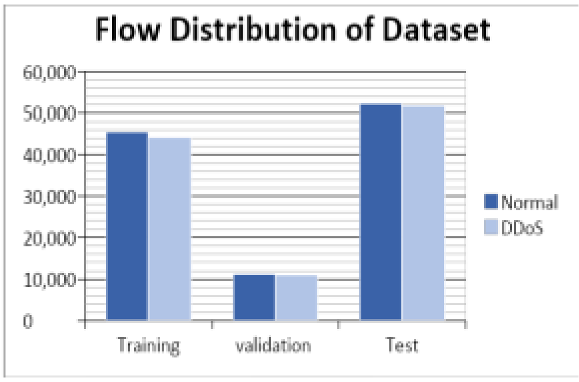

Fig. 5. Flow Distribution among Dataset.

4.1. Performance Metrics

To assess how appropriate the proposed design is in comparison to alternative methodologies, various performance indicators are used. The metric that is most frequently used to gauge an IDS's efficacy is the confusion matrix. There are also many evaluation measures derived from it. Recall, precision, accuracy, and F1 score are additional metrics that are taken into account here to determine how well our model compares to other ML based DDoS detection systems.

- Accuracy (Acy): It is computed by dividing the DDoS prediction accuracy by the average overflow rate.

$$Accuracy(Acy) = \frac{TP+TN}{TP+TN+FP+FN} \qquad (9)$$

- Precision (Pcs): This is calculated as a true positive ratio and the sum of true and false positives.

$$Precision\ (\text{Pcs}\ ) = \frac{TP}{TP+FP} \qquad (10)$$

- Recall (Rcl): This is calculated using the ratio of true positives divided by the sum of true and false positives.

$$Recall(Rcl) = \frac{TP}{TP+FN} \qquad (11)$$

- F1 Score (Fsc): This represents a recall and precision average that is balanced, where true negatives and positives signify correctly categorized traffic, and false positives and negatives signify incorrectly classified traffic.

$$F1score(Fsc) = \frac{2TP}{2TP+FP+FN} \qquad (12)$$

- Receiving Operating Characteristics

The graph displays how well the categorization method performs at each level. The false and accurate favorable rates are plotted. While TPR is assigned to abscissa and stands for specificity, TPR is assigned to ordinate and stands for sensitivity. The area under the ROC curve is a representation of the 2-dimensional area under the ROC curve represents the entire 2-dimensional area in addition to the metrics described above, which shows how accurate the proposed method is.

$$FPR = \frac{FP}{TN + FP}, \tag{13}$$

$$TPR = \frac{TP}{TP + FN}. \tag{14}$$

The effectiveness of the suggested system can be shown in the ROC curve's area under the curve. The following equation is used to calculate the Area Under the Curve (AUC).

$$AUC = \int_0^1 \frac{TP}{TP + FN} d \frac{FP}{TN + FP}. \tag{15}$$

4.2. Choose the Best Minimum List of Features

12 further trials are carried out in order to select the minimum possible relevant attributes that allow our DL model to achieve the best classification result. Table 3 provides an illustration of this. These findings show that the suggested model gives a classification superior to 99.6% on the majority of the measures taken despite removing 89% of features from the initial dataset.

The model is determined to be the best model for future investigation after being trained for 5 seconds. The balance between a classifier's sensitivity and specificity is graphically depicted by the ROC graph. Improved prediction accuracy is shown by a higher ROC score. The ROC of our model is depicted in Fig. 6 using the x and y axes, respectively, which represent the false and accurate classification rates. The graph demonstrates that the proposed model achieves outcomes close to the ideal point, where false positives are 0 and true positives, are 1. The suggested approach can correctly classify about 99.6% of DDoS and normal classes.

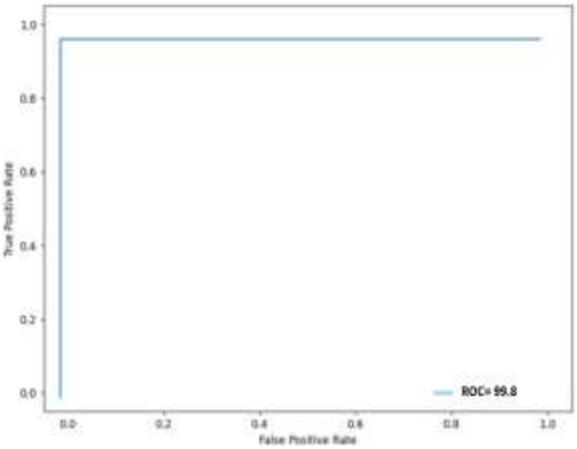

Fig. 6. ROC Curve.

4.3. Detection Performance

This was attained by calculating the quantity of samples processed by the GPU and CPU devices/second. No approach was utilized in the same testing setting to guarantee that the comparison was fair.

Prediction times for different dataset magnitudes range from 2 to 100. From Fig. 7, it was seen that for a tiny dataset, the CPU was observed to perform better than the GPU, with an average inference time of 9.6 ms/batch. Fig. 8 shows in greater detail the number of samples/second processed by ERNN using GPU and CPU with different batch sizes. As the batch size increases, fewer iterations are needed to examine samples from the dataset, but a bigger memory read is appropriate.

In 128 batch-size samples, the improvement brought on by the GPU is apparent. The findings reveal that with the CPU turned off, the suggested model can process about 519,885 samples/second with a batch size of 16,384.

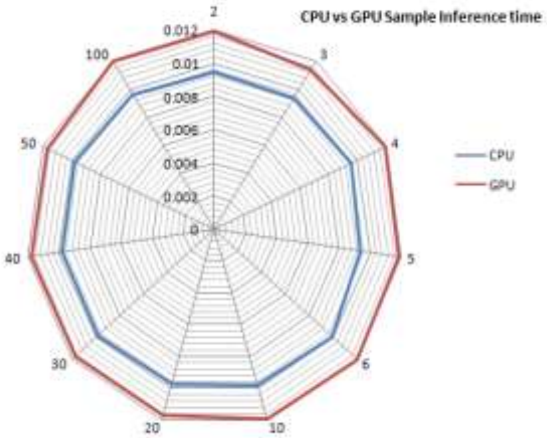

Fig. 7. Different Dataset size inference performance.

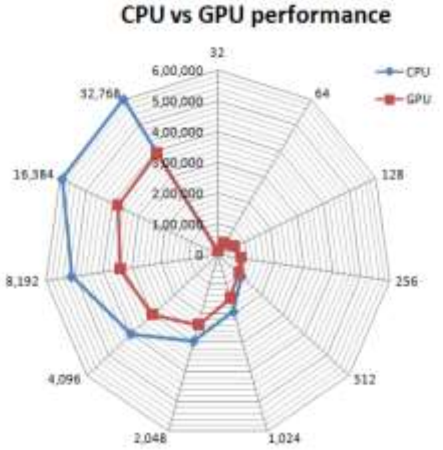

Fig. 8. Different Batch size inference performance.

4.4. Comparison with Different ML Classifiers

To evaluate the effectiveness of the proposed ensemble model, we conducted several experiments on different ML classifiers. Table 4, demonstrates the comparison of the ensemble model with ML classifiers and ensemble with RNN. The data in Fig. 9 shows that ERNN outperforms all traditional techniques. The random forest method, however, performs better than the other six because it assembles a set of decision tree algorithms from a variety of predictions from various decision tree models.

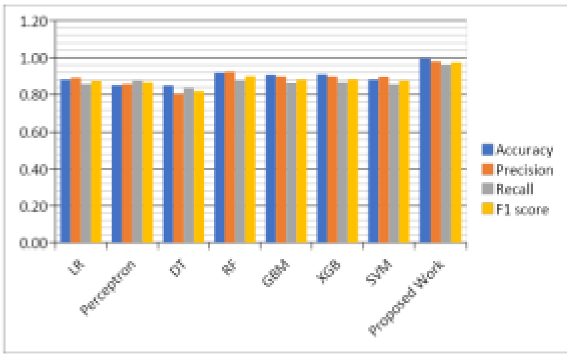

Fig. 9. ERNN versus classical ML.

Table 4. Classification results achieved between ERNN vs ML.

Sno	Models	Acy	Pcs	Rcl	Fsc
1	LR	0.88	0.89	0.86	0.87
2	Perceptron	0.85	0.86	0.88	0.87
3	DT	0.85	0.80	0.83	0.82
4	RF	0.92	0.92	0.88	0.90
5	GBM	0.91	0.90	0.86	0.88
6	XGB	0.91	0.90	0.87	0.88
7	SVM	0.88	0.90	0.86	0.88
8	**ERNN**	**0.996**	**0.98**	**0.96**	**0.974**

4.5. Comparison with State of art DL based systems

Additionally, the proposed model was contrasted with the currently used DL-based techniques whereby precision, accuracy, recall, and F-1 score were taken into account and tabulated in Table. 5 and Fig. 10. In [35] the author proposed a hybrid deep learning model with enhanced feature selection which gained an accuracy of 94.54 using CNN and BiLSTM techniques. Another author [36] presented an IDS framework using an autoencoder for DDoS detection that achieved an accuracy of 95.4. In [37] the author proposed a hyperband-tuned DNN model for detecting DDoS attacks in the cloud. The model achieved good accuracy and reduced false alarms in identifying DDoS attacks. Another cyber threat intelligent approach using PCA with DNN was designed to detect abnormal behavior in the cloud network [38]. In [39] a cascaded feed-forward network was proposed for both the detection and prevention of cyber-attacks that achieved an accuracy of 98.6. Another approach [40] using MLP with a feature selection model detects DDoS attacks with an accuracy of 97.6 %. In [41] the author proposed an ensemble model for malware using the CNN technique. From all the above-mentioned existing works our model ERNN outperforms those in terms of performance. When evaluating the performance of different classifiers, accuracy is preferred because it shows how well the classifier performs across the remaining class distribution range. The proposed approach attains an accuracy of 99.6 in detecting DDoS attacks in the cloud.

Table 5. Performance comparison of the ERNN with different DL techniques.

Sno	Models	Year	Acy	Pcs	Rcl	Fsc
1	CNN+BiLSTM [35]	2021	94.54	94.74	92.07	93.44
2	AE IF [36]	2020	95.4	94.81	97.25	96.01
3	AE+DNN [37]	2020	98.92	97.45	98.97	98.35
4	PCA-DNN [38]	2022	98	0.99	0.97	0.978
5	PRP-FFN [39]	2022	98.6	-	-	-
6	MLP [40]	2020	97.6	99.2	94.88	-
7	Ensemble CNN [41]	2022	93.58	94.39	92.67	93.58
8	Proposed ERNN (86 features)	2023	99.3	99.2	99.1	99.4
9	**Proposed ERNN (24 features)**	2023	**0.996**	**0.997**	**0.996**	**0.997**

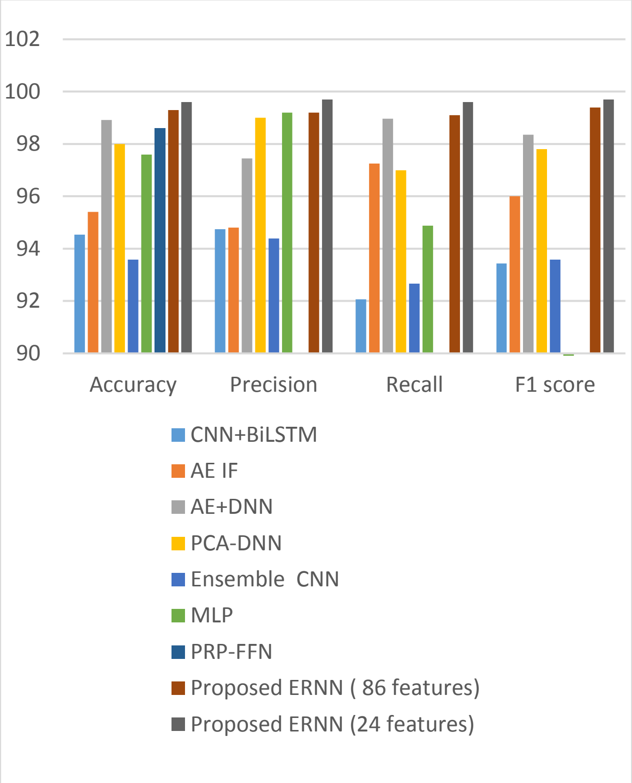

Fig. 10. ERNN versus other DL models.

5. CONCLUSION

DDoS attacks are a persistent threat to the dependability of cloud-based services and are something that service providers all over the world routinely deal with. For DDoS attacks to stop causing resource outages, a reliable detection system is required. However, only a few researchers have thought about utilizing a current dataset that is relevant to the most recent DDoS attacks and incorporates data from those attacks. Additionally, the current strategies entail creating predictions and training models using a lot of processing power. ERNN, an integrated, low-processing-overhead IDS framework using ensemble feature selection with RNN is presented in this research. Multiple ensemble groups based on different classifiers are evaluated to get the minimum feature list with the highest prediction performance. The ERNN model can be trained in 5 seconds and can remove 89% of the features from the CICDDoS2019 dataset used in this study. The model achieved an accuracy of 99.6% in classifying the traffic as attack or normal. One of the limitations of this framework is that it only detects DDoS attacks. Therefore, future work will involve a controller that will block DDoS traffic, allowing normal traffic to be forward.

REFERENCES

[1] Alduailij, Q. W. Khan, M. Tahir, M. Sardaraz, M. Alduailij, and F. Malik, **"Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method,"** Symmetry 2022, vol. 14, no. 6, pp. 1095, May 2022, doi: 10.3390/SYM14061095.

[2] N. Subramanian and A. Jeyaraj, **"Recent security challenges in cloud computing,"** Comput. Electr. Eng., vol. 71, pp. 28–42, Oct. 2018, doi: 10.1016/J.COMPELECENG.2018.06.006.

[3] U. Gurav and R. Shaikh, **"Virtualization: A key feature of cloud computing,"** ICWET 2010 - Int. Conf. Work. Emerg. Trends Technol. 2010, Conf. Proc., no. July 2020, pp. 227–229, 2010, doi: 10.1145/1741906.1741957.

[4] S. Mishra, S. K. Sharma, and M. A. Alowaidi, **"Analysis of security issues of cloud-based web applications,"** J. Ambient Intell. Humaniz. Comput., no. 0123456789, 2020, doi: 10.1007/s12652-020-02370-8.

[5] M. A. Khan, **"A survey of security issues for cloud computing,"** J. Netw. Comput. Appl., vol. 71, pp. 11–29, 2016, doi: 10.1016/j.jnca.2016.05.010.

[6] H. Karthikeyan and G. Usha, **"Real-time DDoS flooding attack detection in intelligent transportation systems,"** Comput. Electr. Eng., vol. 101, p. 107995, Jul. 2022, doi: 10.1016/J.COMPELECENG.2022.107995.

[7] B. Prabadevi and N. Jeyanthi**, "Distributed denial of service attacks and its effects on cloud environment- A survey,"** 2014 Int. Symp. Networks, Comput. Commun. ISNCC 2014, 2014, doi: 10.1109/SNCC.2014.6866508.

[8] M. Mittal, K. Kumar, and S. Behal, **"deep learning approaches for detecting DDoS attacks: a systematic review,"** Soft Comput., pp. 1–37, Jan. 2022, doi: 10.1007/S00500-021-06608

[9] Mallampati, S. B., & Seetha, H, **"A Review on Recent Approaches of Machine Learning, Deep Learning, and Explainable Artificial Intelligence in Intrusion Detection Systems,"** Majlesi Journal of Electrical Engineering, vol. 17, no. 1, pp. 29-54, 2023.

[10] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad**, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches,"** Trans Emerg. Tel Tech, vol. 32, 2021, doi: 10.1002/ett.4150.

[11] D. Alghazzawi, O. Bamasaq, H. Ullah, and M. Z. Asghar, **"Efficient Detection of DDoS Attacks Using a Hybrid deep learning Model with Improved Feature Selection,"** Appl. Sci. 2021, Vol. 11, Page 11634, vol. 11, no. 24, p. 11634, Dec. 2021, doi: 10.3390/APP112411634, 2021.

[12] Dehkordi, A. B., Soltanaghaei, M., & Boroujeni, F. Z, **"A Hybrid Mechanism to Detect DDoS Attacks in Software Defined Networks,"** Majlesi Journal of Electrical Engineering, vol. 15, no. 1, 2021.

[13] Jaw, E., & Wang, X, **"Feature selection and ensemble-based intrusion detection system: an efficient and comprehensive approach,"** *Symmetry*, vol. 13, no. 10, pp. 1764, 2021.

[14] Mahdi, M. M., Mohammed, M. A., Al-Chalibi, H., Bashar, B. S., Sadeq, H. A., & Abbas, T. M. J, **"An Ensemble Learning Approach for Glaucoma Detection in Retinal Image,"** Majlesi Journal of

Electrical Engineering, vol. 16, no. 4, pp. 117-122, 2022.

[15] Bahmani, A., & Monajemi, A, **“Introducing a Two-step Strategy Based on Deep Learning to Enhance the Accuracy of Intrusion Detection Systems in the Network,”** Majlesi Journal of Telecommunication Devices, vol. 8, no. 1, pp. 21-25, 2019.

[16] Filonov, P., Kitashov, F., & Lavrentyev, A,”**Rnn-based early cyber-attack detection for the tennessee eastman process,”** arXiv preprint arXiv:1709.02232, 2017.

[17] M. Aamir, S.M.A. Zaidi, **“DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation,”** Int. J. Inf. Secur, vol. 18, pp. 761–785, https://doi.org/10.1007/s10207-019-00434-1, 2019

[18] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martínez-del-Rincón and D. Siracusa, **"Lucid: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection,"** in *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 876-889, June 2020, doi: 10.1109/TNSM.2020.2971776.

[19] Moubayed, A., Aqeeli, E., & Shami, A, **“Ensemble-based feature selection and classification model for DNS typo-squatting detection,”** In 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-6). IEEE.

[20] Agarwal, A., Khari, M., & Singh, R, **“Detection of DDOS attack using deep learning model in cloud storage application,”** Wireless Personal Communications, pp. 1-21, 2021.

[21] Alghazzawi, D., Bamasag, O., Ullah, H., & Asghar, M. Z, **“Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection,”** Applied Sciences, vo. 11, no. 24, pp. 11634, 2021

[22] Abbas, A., Khan, M. A., Latif, S., Ajaz, M., Shah, A. A., & Ahmad, J, **“A new ensemble-based intrusion detection system for internet of things,”** Arabian Journal for Science and Engineering, pp. 1-15, 2021.

[23] Krishnaveni, S., Sivamohan, S., Sridhar, S. S., & Prabakaran, S, **“Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing,”** Cluster Computing, vol. 24, no. 3, pp. 1761-1779, 2021.

[24] Saha, S., Priyoti, A. T., Sharma, A., & Haque, A, **“Towards an Optimized Ensemble Feature Selection for DDoS Detection Using Both Supervised and Unsupervised Method,”** Sensors, 22(23), 9144, 2022.

[25] Priyadarshini, R., & Barik, R. K, **“A deep learning based intelligent framework to mitigate DDoS attack in fog environment,”** Journal of King Saud University-Computer and Information Sciences, 34(3), 825-831, 2022.

[26] Jiang, J., Liu, F., Ng, W. W., Tang, Q., Zhong, G., Tang, X., & Wang, B, **“AERF: Adaptive ensemble random fuzzy algorithm for anomaly detection in cloud computing,”** Computer Communications, 2023.

[27] Bouke, M. A., Abdullah, A., ALshatebi, S. H., Abdullah, M. T., & El Atigh, H, **“An intelligent DDoS attack detection tree-based model using Gini index feature selection method,”** Microprocessors and Microsystems, vol. 98, pp. 104823, 2023

[28] Abosuliman, S. S, **“Deep learning techniques for securing cyber-physical systems in supply chain 4.0,”** Computers and Electrical Engineering, vol. 107, pp. 108637, 2023.

[29] Balamurugan, V., & Saravanan, R, **“Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation,”** Cluster Computing, vol. 22, no. 6, pp. 13027-13039, 2019.

[30] I. Ortet Lopes, D. Zou, F. A. Ruambo, S. Akbar, and B. Yuan, **“Towards Effective Detection of Recent DDoS Attacks: A deep learning Approach,”** Secur. Commun. Networks, vol. 2021, 2021, doi: 10.1155/2021/5710028

[31] Haider, S., Akhunzada, A., Mustafa, I., Patel, T. B., Fernandez, A., Choo, K. K. R., & Iqbal, J. (2020), **“ A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks,”** Ieee Access, vol. 8, pp. 53972-53983.

[32] Naveen Bindra and Manu Sood, **“Detecting DDoS Attacks Using Machine Learning Techniques and Contemporary Intrusion Detection Dataset,”** Autom. Control Comput. Sci., vol. 53, no. 5, pp. 419–428, Sep. 2019, doi: 10.3103/S0146411619050043/TABLES/3.

[33] O. Yousuf and R. N. Mir, **“DDoS attack detection in Internet of Things using recurrent neural network,”** Comput. Electr. Eng., vol. 101, no. May, p. 108034, 2022, doi: 10.1016/j.compeleceng.2022.108034.

[34] Z. Chiba, N. Abghour, K. Moussaid, A. El, and M. Rida, **“Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms Network intrusion detection system Deep Neural Network Genetic algorithm Simulated Annealing Algorithm,”** Comput. Secur., vol. 86, pp. 291–317, 2019, doi: 10.1016/j.cose.2019.06.013.

[35] V Alghazzawi, D., Bamasag, O., Ullah, H., & Asghar, M. Z, **“Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection,”** Applied Sciences, vol. 11, no. 24, pp. 11634, 2021

[36] K. Sadaf and J. Sultana**, "Intrusion Detection Based on Autoencoder and Isolation Forest in Fog Computing,"** in IEEE Access, vol. 8, pp. 167059-167068, 2020, doi: 10.1109/ACCESS.2020.3022855.

[37] Bhardwaj, A., Mangat, V., & Vig, R, **“Hyperband tuned deep neural network with well posed stacked sparse autoencoder for detection of DDoS attacks in cloud,”** IEEE Access, vol. 8, pp. 181916-181929, 2020.

[38] Al-Fawa'reh, M., Al-Fayoumi, M., Nashwan, S., & Fraihat, S, **“Cyber threat intelligence using PCA-DNN model to detect abnormal network behavior,”** Egyptian Informatics Journal, vol. 23, no. 2, pp. 173-185, 2022.

[39] Kadhim, Q. K., Al-Sudani, A. S., Almani, I. A.,

Alghazali, T., Dabis, H. K., Mohammed, A. T., ... & Mezaal, Y, "**IOT-MDEDTL: IoT Malware Detection based on Ensemble Deep Transfer Learning,"** Majlesi Journal of Electrical Engineering, vol. 16, no. 3, pp. 47-54, 2022.

[40] Wang, M., Lu, Y., & Qin, J, **"A dynamic MLP-based DDoS attack detection method using feature selection and feedback,"** Computers & Security, vol. 88, pp. 101645.

[41] Jain, R., & Bhatt, C, **"PRP-Based Cascaded Feed-Forward Network for Detection and Prevention of DDoS Cyber Attacks,"** International Journal of Innovative Research in Technology and Management, Vol. 6, no. 2, pp. 131-140, 2022.