FE-DADT: A FEATURE ENGINEERING DRIVEN DDOS ATTACK DETECTION TECHNIQUE AND EVALUATION FOR EFFECTIVE MITIGATION

SWATHI KATHI[1]* NARSIMHA GUGULOTHU[2]

[1]Dept. of Computer Science and Engineering, CVR College of Engineering, Hyderabad, India
[2]Dept. of Computer Science and Engineering, College of Engineering, Jawaharlal Nehru Technological University, Hyderabad, India
Email: [1]swathireddykathi@gmail.com, [2]narsimha06@gmail.com

ABSTRACT

Distributed denial-of-service (DDoS) attacks have been a persistent threat to the continuous operation of network services by depleting its essential resources and normal users cannot access the services. With the increasing evolution of these attacks, simple detection of differentiating malicious packets from innocent packets within traffic has become a complex task. Therefore, effective mitigation relies on accurate identification of discriminatory features of the network as opposed to similar (but non-redundant) sample-level analysis in higher dimensional space. In this paper, we present a filter-based feature selection method rooted in the statistical validity of the Student t-test that identifies the strongest predictors from traffic flow data. The task is to find a minimal but optimal set of features that improves classifier performance by noise and redundancy removal. On CICDDoS2019 benchmark dataset, the model was trained with 399,998 and tested on 112,611 samples. The results show that t-test is able to detect 58 features with statistical significace. With this optimized feature set, a eXtended Gradient Boosting (XGBoost) classifier reached a 99.82% detection rate and 97.40% classification accuracy. Our results show that a feature elimination approach based on statistical grounds, strengthens a significant increase in the performance of DDoS detection systems based on ML methods.

Keywords: *DDoS, Classification, Network traffic flow, Prediction, Feature selection, Hypothesis Testing, Machine Learning*

1. INTRODUCTION

The never-ending menace of Distributed Denial-of-Service (DDoS) attacks persistently challenges the underpinnings of stability that modern peers and goliaths establish digitally. These orchestrated attacks have mutated in rate and intricacy [1], seeking to savage administrations through inundating system assets. At the same time, the internet has seen its own evolution, that has coincided with this era of exponential growth in how we rely on web-based applications. While this mass adoption fundamentally changed commercial and social interactions, it has also tremendously expanded the attack surface for malicious actors. Thus, the need for further refined cyber security systems is not speculative but urgent, a response to what some would consider defining technological shifts such as the widespread development of insecure armies of IoT devices, the never-ending expansion of network complexity, and the deep integration of online platforms into critical infrastructure. Large networks and critical network infrastructures provide cyber threats in form of DDoS incursions, malware, and unauthorized access to system pose a specific high level risk of extreme operational disruption and extensive financial harm. The main purpose of a DDoS campaign is to bring down a service rendering it inaccessible to legitimate users for a period or permanently [2]. The impacts of these attacks vary based on the avenue through which they are carried out. As a result, they are divided into a specific taxonomy, as displayed

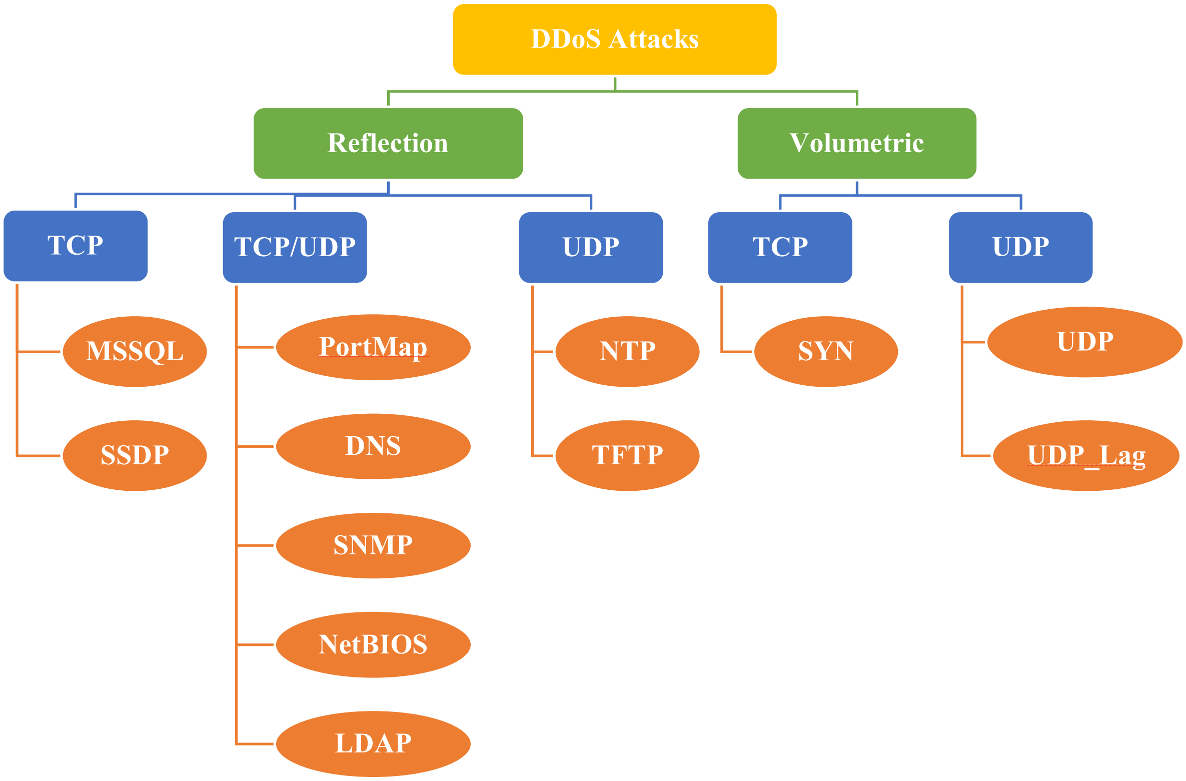


Figure. 1 Ddos Attack Taxonomy[4]

Figure 1: Most modern intrusion detection systems performance are impaired and unforeseen due to unsuitable feature extraction. One of the key TM that make the building the efficient ID system difficult, relates to curse of dimensionality; as more features you include in your system, you risk more noise spread and losing the signal of an attack. This is where feature selection plays a significant role in [10]. Through systematic identification and elimination of unrelated or duplicate features, we can effectively boost a model's accuracy while also optimizing it for speed. This is not just an optimization, but a prerequisite for building strong classifiers on the large datasets that are the backbone of NTA tasks. Although the use of benchmarks for feature selection methods has been common in foundational research [11] our foundational research cannot be validated using datasets such as NSL-KDD [12] since the NSL-KDD dataset does not represent current attack approaches. DNN-based ML using normal and attack traffic has been largely unexplored; accordingly, in this work, a dataset that can fill this gap is CICDDoS2019. This 2019-compiled list is more relevant to today's threat landscape, which has seen a range of newer DDoS attack vectors. As a result, it received widespread attention to researchers' efforts in the comparison of many types of feature selection approaches in the search for the best minimal feature subset for correct classification of the attacks [4]. One of the major disadvantages with these traditional techniques is that they need to be executed multiple times to find out an optimal feature," which significantly increases the processing time.

To address these and other related challenges, this paper proposes a feature selection approach based on inferential statistics. A taxonomy of DDoS attacks - based on techniques/methods/protocols (Fig1) DDoS attacks target flooding traffic and making it inaccessible to legitimate users [16]. Returns: The precise diagram divides the attacks into two main broad headings — Reflection, and Volumetric. In reflection attacks, the attacker still sends requests to a third-party server, but uses the spoofed IP address of the victim. The server thus reacts to the target, which increases the attack. Such attacks range from MSSQL to PortMap, NTP, SSDP, DNS and TFTP, within the SNMP, NetBIOS, and LDAP categories. In these types of attacks, different protocols are exploited to create ample amount of traffic towards the victim [5]. When you think about good old fashioned brute force, volumetric attacks are the familiar weapon of choice for disrupting a network. The only thing they want to do is block the pipes — to use up every last drop of available bandwidth so that legitimate traffic can no longer get through [13]. We can categorize these into two protocols they abuse TCP and UDP. Due to its nature, some TCP based attacks such as the SYN floodin are sarcastic because opposed to just blasting raw traffic. They leverage the very basis of communication — the TCP handshake — by blasting a plethora of connection requests that remain unfulfilled. Well, this makes the target server to wait, occupying resources, until it can no longer process real users. On the other hand, when it breaches UDP-based attacks, it is usually much easier. For example, methods such as UDP floods and UDP_Lag just saturate the target with more UDP packets than it can handle, attempting to flood the target's network capacity with sheer quantity. Understanding this breakdown is not merely academic — it is the first and most important step to being able to construct a proper defense [6]. Understanding if it is a TCP resource draining attack or a UDP deluge can help a networking team get past broad brush strokes. They can use rapid countermeasures—such as hysteresis where available to tune rate limits for a given protocol, designing intelligent filters for the game traffic, or other forms of mitigations to really halt the attack on its tracks.

In this paper, a feature selection approach based on the statistical T-test is presented in order to select the most important features among the feature set for DDoS detection. These features are then fed into machine learning models which are trained to improve detection rate and accuracy whilst reducing computational cost.

The remainder of this paper is structured as follows: Section 2 provides a background on related works regarding feature selection and DDoS detection. The methodology proposed, including the statistical T-test approach is described in section 3. Dataset and preprocessing steps are described in section 4. The experimental setup and the results are presented in Section 5. Even discussing the related work in Section 6 compares with existing methods. Lastly, Section 7 closes the paper and discusses the future work that may be derived from it.

Motivation and Contribution of This Work

Most of the previous DDoS detection research have focused on enhancing the accuracy of detection by utilizing complex machine learning and deep learning models or by heuristic and optimization-based feature selection techniques. Even though these approaches yield high accuracy, they employ computationally intensive processes, do not provide insight into the importance of features, or justify why features are statistically significant without directly employing a statistical method. Additionally, a number of papers use conventional datasets (e.g., NSL-KDD) and the papers who explore an advanced dataset (e.g., CICDDoS2019) present a limited explanation about the feature selection procedure.

Inspired by these shortcomings, this work approaches feature selection from a new angle where statistical validity, interpretability, and computation efficiency of feature selection are in the center of interest. Instead of computing the optimal model complexity, the introduced approach has only focused on features that show statistical significant difference between the benign and the attack traffic, measured by an independent T-test. The main contribution of this study is in showing that a simple statistically grounded filter-based procedure can exhibit comparable detection performance against a state-of-the art DDoS dataset while massively decreasing the time required for feature selection and improving interpretability of the procedure. The motivation behind this proposed work is different from the existing studies, which can relate best those studies correlated to feature-level statistical justification, whereas, in the case of the proposed work, model sophistication has been placed at a secondary position.

2. RELATED STUDIES

The ideal feature selection problem has been a popular area of research for more robust and precise Intrusion Detection Systems (IDS). Although many methods have been suggested, the relative value of these methods is usually not revealed until they are applied on standard benchmarks. Initial works in this area generally relied on classical filter methods. For instance, Singh et al. In the NSL-KDD dataset, the 41 attributes were reduced to a large number of 26 attributes using the Information Gain (IG) method [14]. It was not the filter that was innovative but rather the hybrid classifier that succeeded it, using a Support Vector Machine (SVM) combined with a Bat Algorithm (BA) to direct feature selection with a reasonable degree of success at 94.16%, illuminating a relationship between feature selection and a bio-inspired optimization model.

Outside of pure filter methods, other researchers used nature-inspired optimization methods to deal with high dimensionality. Aghdam et al. Method [15] utilized an Ant Colony Optimization (ACO) algorithm, which was highly successful in selecting only 24 features from the same NSL-KDD benchmark, and all these 24 features are sufficient to produce a competitive performance. To validate the power of this feature subset, a simple nearest neighbour classifier was trained on the features and achieved an exciting 98.9% accuracy; sometimes features are more important than the model type! Taking bio-inspired computation a step further, Sarvari et al. That proposed an entirely new algorithm known as Mutation Cuckoo Fuzzy (MCF) [13]. This approach was very aggressive, reducing the feature space to 22 attributes. Using a feature selection technique based on an Evolutionary Neural Network (ENN), the model was able to reach remarkable results (98.81% accuracy). Yet, the authors in fact themselves noted a major limitation of the study, that "our model is only validated on the outdated NSL-KDD dataset, thus it is unclear that our model can perform on modern network traffic", a note which much reflects a major limitation in the field as a whole - the need to validate with up-to-date data. Some researchers have examined more recent datasets. Sharafaldin et al. Using a radviz plot visualization technique, [4] have proposed distinct salient features for each of the attack category associated with the CICDDoS2019 dataset. On this subset, we implemented an ID3 decision tree classifier, obtaining a detection rate of 65%. In another investigation, Kshirsagar et al. A hybrid method called Information Gain + Correlation (IG+CR) for feature selection based on the service for specific attack types is presented by [18]. The authors detected some attacks from the CICDDoS2019 dataset [17], and the average accuracy of their model was 99.89%. On the other hand, their approach was only implemented on a few attack classes and they did not handle class imbalance in the dataset.

A. A. Alashhab et al [19] proposes to detect the DDoS attacks in Software Defined Networks(SDN) domain. 22 features obtained by performing statistical analysis of their own dataset. To evaluate this method utilized CICDDoS2019 dataset and achieved 98.72% Accuracy, 98.83% detection rate,18.51% FPR,99.78% precision and 98.76% fscore. However, this method didn't disclose the method to select the relevant features.

Hajimaghsoodi and Jalili [20] suggested a novel method i.e.3-pahse counter measure, RAD model to detect DDoS attacks using statistical approach. In first phase ranks allotted to users, next phase will execute at server-side and raised the alarms when threshold value exceed and the last phase can evaluate what are the users blocked based on phase alarms. To evaluate this model CICDDoS2019 dataset utilized and achieved 99% detection rate, 80% precision and 89% F1 score. However, it is not disclosed how many features were used to evaluate their model [21].

Further investigations utilizing the CICDDoS2019 dataset demonstrate a variety of effective methodologies. J. Halledy et al. [23] established that a focused set of 25 time-based features can be highly effective, achieving a 98.53% accuracy and a 0.98 F-score in DDoS detection. Expanding on feature categories, Jia et al. [22] proposed a set of 40 features, combining time-related, packet-related, and identification-related attributes. Their proposed framework, FlowGuard, employs an LSTM model for identifying DDoS attacks within IoT ecosystems. Evaluation on the CICDDoS2019 dataset yielded strong results: 98.93% accuracy, 99.32% recall, 99.71% precision, and a 0.9935 F-score.

Focusing on a specific network architecture, Almari et al. Software-Defined Networking(SDN) environments have a centralized control between controllers and switches, which are more susceptible to resource-exhaustion DDoS attacks [24]. In their two-stage approach, bandwidth regulation and an eXtended Gradient Boosting (XGB) classifier were combined and yielded remarkable performance values in terms of accuracy (99.9%) and a perfect 100% in precision, recall, and F1-score. Deep leraning architectures have also been exploited in several other studies. Cil et al. A DNN model in [25] was used that was 99.99% accurate for detection, and 94.57% accurate in classification. As the authors observe, there is often a trade-off with such models: while performance is high, their "black-box" nature can limit the interpretability of which features drive the decisions. A. A. Maiga et al. [26] designed a hybrid deep learning model combining BiLSTM with LSTM. Using 55 features from the CICDDoS2019 dataset, their model achieved an accuracy of 99.76%, a precision of 99.99%, a recall of 99.73%, an F-score of 99.86%, with a very low false positive rate (FPR) of 0.046%. In a resource-efficient approach, A. Zainudin et al. [27] developed a Hybrid-DNN model that required only 10 input features. Their work focused on detecting three specific DDoS attack types from the dataset, reporting an accuracy of 99.45%, recall of 99.46%. A summary of this literature survey is consolidated in Table 1. It highlights that a majority of these studies employ the modern, yet often imbalanced, CICDDoS2019 dataset. Despite varying feature selection methods and model architectures—including XGBoost, LSTM, and DNN—the collective results consistently show high accuracy, precision, and recall, underscoring significant advancements in DDoS detection capabilities.

Table 1. Motivation- And Outcome-Level Comparison Of Prior Studies And Proposed Work

Aspect	Prior Studies	Proposed Work
Primary Motivation	Maximize detection accuracy	Balance detection performance, efficiency, and interpretability
Feature Selection Rationale	Heuristic, optimization-based, or unspecified	Statistically grounded (independent T-test)
Dataset Relevance	Often NSL-KDD or CICDDoS2019	CICDDoS2019 (modern attack behavior)
Feature Transparency	Limited or implicit	Explicit statistical significance testing
Computational Overhead	Moderate to very high	Low
Model Dependency	Model-centric (DL-heavy)	Feature-centric, model-agnostic
Practical Deployability	Limited by complexity	Suitable for real-time IDS
Key Limitation	High cost, poor explainability	Binary classification only

Whereas Table 1 provides a more granular view in summarizing the methodological and performance-oriented insights offered in previous DDoS detection studies, Table 1 offers a more abstract comparative analysis showcasing differences in motivation, underlying philosophy on feature selection, and applicability in practice. While many current state-of-the-art approaches present exceedingly high detection efficacy, the models they employ are often complex, feature selection strategies are unstated, or training is computationally intensive. So, instead the motivation of the proposed work is to have a statistically transparent and computationally efficient feature selection mechanism that is robust to modern and large-scale data. This difference in motivation and observations make our proposed approach a more practical alternative to the complexity-based approaches available in the literature.

2.1 Literature Gaps and Problem Motivation

The high dimensionality of network traffic datasets poses a significant barrier to the efficacy of an IDS. Excessive features slow down model training and prediction, use more memory, and can even decrease accuracy if they introduce noisy or redundant data. To address this problem, conventional feature selection methods have been applied frequently in the past.

Drawbacks of traditional Feature Selection Methods It is very good for rank features but may maintain redundant features. Chi-Square — Chi-Square is fast, but mainly for categorical data it discards a lot of information from continuous features. Methods such as Forward Selection and Recursive Feature Elimination (RFE) are accurate but slow and expensive on large datasets (Wrapper method). Embedded methods like LASSO or Random Forest are computationally light but biased, removing features that are potentially useful. Lastly, the so-called nature of NSL-KDD actually makes it out-of-date because it is a good practice that the attack behaviors are updated over time, otherwise it is challenging to generalize models from constructed benchmarks onto actual networks.

Due to these limitations, the current feature selection methods are very time-consuming, ignore significant features, or fail in representing the modern attack behavior. Hence, a fast computational, statistically reasonable and easy to implement approaches are required to rapidly pick only the relevant features/ that contribute to detection accuracy. Our proposed method of Feature Selection based on T-test stands on this gap.

2.2 Problem Statement and Research Questions

The literature review and gap analysis form Section 2.1 show that existing feature selection approaches for DDoS attack detection have some limitations. Common filter-based methods tend to keep redundant or irrelevant features, while wrapper and embedded methods have a high computational cost when dealing with large-scale and high-dimension traffic datasets. Additionally, most recent works use outdated datasets like NSL-KDD or use heavy deep learning model with high accuracy, but low interpretability and unsuitable for time-sensitive intrusion detection system. Moreover, feature selection is often poorly described or based on statistical reasoning even when modern datasets like CICDDoS2019 are used.

Problem Statement: A feature selection mechanism that can efficiently select a minimal but statistically significant number of features from a high-dimensional and imbalanced network traffic dataset and represent the features with relevant contemporary DDoS attack behaviors. This mechanism would help minimize computational overhead without compromising or increasing attack detection accuracy and decreasing false positives in machine learning–based intrusion detection systems.

This issue is what this study intends to address, using the following research questions:

RQ1:

Does a statistically sound filter-based feature selection method, the independent T-test, enable us to differentiate between the benign traffic feature and the DDoS attack traffic feature?

RQ2:

Q1: Does the T-test–based feature selection approach proposed in this work reduces the feature dimensionality and the time required for feature selection, and meets high detection performance compared to conventional feature selection methods?

RQ3:

Research Question3: What is the effect of statistically selected features on various machine learning classifiers for detection rate, accuracy, balanced accuracy and false positive rate in the CICDDoS2019 dataset?

Based on the above research questions, we have designed the methodology and experimental evaluation, which will form the basis of the next sections of this paper.

3. METHODOLOGY

We applied an independent t-test to each preprocessed continuous feature to find those that differ significantly between benign and attack traffic. First, class labels were converted to binary (benign = 0, all attacks = 1) to match the detection goal (Section 4.4). For each feature we computed group means, variances and sample sizes, then calculated the Welch t-statistic and its degrees of freedom. A two-sided test with $\alpha = 0.05$ was used and features with $|t| > t\_critical$ were retained. Given the large sample sizes, the t-test is robust to mild non-normality; we also applied min-max normalization before testing. To control false discoveries from multiple testing we report both raw and FDR-adjusted p-values.

3.1 Statistical T-Test

We used an independent samples t-test to filter the most statistically significant features from our data. This implementation is ideal for what we want to do because it compares the mean of a continuous variable from two different unrelated groups, which in this case is each network traffic feature between benign traffic and attack traffic. At its heart, the t-test is a simple yet elegant concept: it produces a test statistic that measures the size of the difference of the two group means relative to the variation within each group. Now, the bigger this statistic, the less likely whatever difference you observe is purely random chance. We can compare this to a tabulated critical value (associated with the t-distribution and the alpha level will select the threshold for significance that is used) to automatically identify which features show a statistically significant difference between attack and normal traffic and are therefore good candidates for our classifier [5].

Next we describe how to usethe statistical filter step-by-step on the dataset, filtering out the noise and identifying the features that really count for a DDoS detection.

3.2 Case Study

Table 2: Example dataset for statistical T-Test In this example, we will take it as step by step case study for the statistical T-Test.

Segmentation Based on Class Label – The first step of the analysis is dividing the dataset based on its respective class labels. The separation is required for the following mean comparison. As an example of this segmentation, Tables 3 and 4 show a part of the whole data, the distribution of cases by class for the process.

Table 2. Sample Dataset Of 10X2 Size With Binary Class

Flow Duration	**Class Label**
78	Attack
82	Attack
75	Attack
253	Attack
181	Attack
25	benign
27	benign
17	benign
0	benign
35	benign

Step 1: Data Step 2: Formulating the Hypothesis:

Null Hypothesis (H₀): The feature's group means are equal ($\mu_1 = \mu_2$). Under this hypothesis, the feature is deemed irrelevant for the model, as it fails to differentiate between the classes.

Alternative Hypothesis (H₁): The feature's group means are not equal ($\mu_1 \neq \mu_2$). The observed difference is statistically significant, leading to the conclusion that the feature is a relevant predictor for the classification objective.

Step 3: Calculating mans of the two groups.

$$\text{mean} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad (1)$$

$$\overline{X}_{\text{attack}} = \frac{78 + 82 + 75 + 253 + 181}{5} = 133.8$$

$$= \overline{X}_{\text{benign}} = \frac{25+27+17+0+35}{5} = 20.8$$

Step 4: Calculating variance and standard deviation of the two groups.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\mathrm{Variance} = \frac{\sum_{i=1}^{n}(x_i - \mathrm{mean})^2}{n-1} \qquad (2)$$

$\mathrm{Variance}_{\mathrm{attack}}$

$$= \frac{(78-133.8)^2 + (82-133.8)^2 + (75-133.8)^2 + (253-133.8)^2 + (181-133.8)^2}{5-1}$$

$= 6422.7$

$\mathrm{Variance}_{\mathrm{benign}}$

$$= \frac{(25-20.8)^2 + (27-20.8)^2 + (17-20.8)^2 + (0-20.8)^2 + (35-20.8)^2}{5-1}$$

$= 176.2$

Step 5: Calculate the $T_{\mathrm{observed\_value}}$, with the following formula

$$T_{\mathrm{observed\_value}} = \frac{\bar{X}_{\mathrm{attack}} - \bar{X}_{\mathrm{benign}}}{\sqrt{\frac{(\sigma_{\mathrm{attack}})^2}{n} + \frac{(\sigma_{\mathrm{benign}})^2}{m}}} \qquad (3)$$

Here,

$\bar{X}_{\mathrm{attack}}$ = mean of attack group

$\bar{X}_{\mathrm{benign}}$= mean of benign group

$(\sigma_{\mathrm{attack}})^2$= variance of attack group

$(\sigma_{\mathrm{benign}})^2$= variance of benign group

n = number of attack samples

m = number of normal samples

$$T_{\mathrm{observed\_value}} = \frac{133.8 - 20.8}{\sqrt{\frac{6422.7}{5} + \frac{176.2}{5}}} = 3.110481854$$

Step 6: Compute degree of freedom (DOF).

If $\mathrm{Variance}_{\mathrm{attack}} == \mathrm{Variance}_{\mathrm{benign}}$ **then: DOF** $= (n + m - 2)$ (4)

else : DOF =

$$\frac{\left(\frac{\mathrm{Variance}_{\mathrm{attack}}}{n} + \frac{\mathrm{Variance}_{\mathrm{benign}}}{m}\right)^2}{\left(\frac{(\mathrm{Variance}_{\mathrm{attack}})^2}{n^2(n-1)} + \frac{(\mathrm{Variance}_{\mathrm{benign}})^2}{m^2(m-1)}\right)} \qquad (5)$$

$$\mathbf{DOF} = \frac{\left(\frac{6422.7}{5} + \frac{176.2}{5}\right)^2}{\left(\frac{(6422.7)^2}{5^2(5-1)} + \frac{(176.2)^2}{5^2(5-1)}\right)}$$

$= 1.70165E+13$

Step 7: Compute $T_{\mathrm{critical\_value}}$ using T-distribution table by 95% confidence interval.

Table 5 depicts i.e. T- distribution table. Obtained Tcritical_value w.r.t DOF of 1.70165E+13 at 95% confidence interval is 1.96. Tobserved_value and Tcritical_value w.r.t to degree of freedom information depicted in Table 6.

From table 6 we can conclude that, which feature Tobserved_value is greater than Tcritical_value that will be select as significant feature. The same procedure followed to analysing CICDDoS2019[4] dataset.

3.3 Research Protocol

This study adopts a research protocol that encompasses a structured highly reproducible workflow that remains consistent with the precept of previous DDoS detection studies but integrates statistically founded advancements in feature selection. Other approaches using the CICDDoS2019 dataset on intrusion detection [4], [19], [22], [23], and mitigation have also been taken within this protocol-driven methodology [20], [21].

To begin with, in accordance with previous studies [4], [6], a current and publicly available benchmark dataset characterized by realistic traffic and a broad set of modern DDoS attack types, CICDDoS2019, was chosen. Both training and testing datasets are applied in respect to the pre-defined structure of the dataset to prevent information leakage, similar to the experimental practices followed in [22], [23].

Second, we preprocess the data by removing non-informative attributes and identifiers, and also features with zero variance, which is a common way of filtering out the noise and reducing the dimensionality of the dataset considered in previous intrusion detection researches [18],[24]. To provide for uniform range so that all features participate equally during learning, min–max normalization was applied to scale feature values within a constant range [25, 26].

Third, independent T-test was performed for feature selection to find statistically significant features to separate benign traffic and DDoS attack traffic. This statistically driven filter-based method (inspired similarly to [19] and [20] in that it relies on statistical characterizations of the features rather than using a heuristic or wrapper based approach as in earlier works [14], [15]) supports fast feature selection with high interpretability due to an explicit significance testing framework.

Fourth and finally, all attack types are grouped under a single attack label in the dataset, therefore transforming this multi-class labels into a binary classification problem. This transformation is compatible with past DDoS detection works that treat DDoS detection as an attack detection problem instead of attack classification problem [4], [23], making the comparison of results straightforward.

Fifth, the selected feature subset was used to train several machine learning classifiers, comprising of Decision Tree, Logistic Regression, Naïve Bayes, Random Forest, Linear Discriminant Analysis, Quadratic Discriminant Analysis, and XGBoost. Inspired by recent research findings which showed that ensemble models (like XGBoost) are robust and provide superior detection performance of DDoS detection tasks [22], [24], [27], we also utilize them.

Finally, we performed stratified 10-fold cross-validation on our training dataset, and we validated on an independent test set. We evaluated the performance by standard intrusion detection metrics, such as detection rate, accuracy, balanced accuracy, precision, recall, F1-score and false positive rate, which is in line with the evaluation protocols in previous studies [19], [23] and [26]. Such a protocol provides an effective and reliable comparison to the state of the art while ensuring reproducibility.

4. DATASET

For this experimentation, a publicly accessible CICDDoS2019 dataset was employed [4]. According to an evaluation study by Raghupathi et al. [6], the CICDDoS2019 dataset meets all eleven criteria outlined in Gharib's evaluation framework [7] for a high-quality intrusion detection dataset. This makes it particularly suitable for training machine learning algorithms to recognize contemporary distributed denial-of-service (DDoS) attacks, unlike other publicly available alternatives [8,9].

Table 3: T- Distribution Table

DOF	0.05	0.01	0.001	DOF	0.05	0.01	0.001
1	12.706	63.657	636.619	30	2.042	2.75	3.646
2	4.303	9.925	31.599	31	2.04	2.744	3.633
3	3.182	5.841	12.924	32	2.037	2.738	3.622
4	2.776	4.604	8.61	33	2.035	2.733	3.611
5	2.571	4.032	6.869	34	2.032	2.728	3.601
6	2.447	3.707	5.959	35	2.03	2.724	3.591
7	2.365	3.499	5.408	36	2.028	2.719	3.582
8	2.306	3.355	5.041	37	2.026	2.715	3.574
9	2.262	3.25	4.781	38	2.024	2.712	3.566
10	2.228	3.169	4.587	39	2.023	2.708	3.558
11	2.201	3.106	4.437	40	2.021	2.704	3.551
12	2.179	3.055	4.318	42	2.018	2.698	3.538
13	2.16	3.012	4.221	44	2.015	2.692	3.526
14	2.145	2.977	4.14	46	2.013	2.687	3.515
15	2.131	2.947	4.073	48	2.011	2.682	3.505
16	2.12	2.921	4.015	50	2.009	2.678	3.496
17	2.11	2.898	3.965	60	2	2.66	3.46
18	2.101	2.878	3.922	70	1.994	2.648	3.435

19	2.093	2.861	3.883	80	1.99	2.639	3.416
20	2.086	2.845	3.85	90	1.987	2.632	3.402
21	2.08	2.831	3.819	100	1.984	2.626	3.391
22	2.074	2.819	3.792	120	1.98	2.617	3.373
23	2.069	2.807	3.768	150	1.976	2.609	3.357
24	2.064	2.797	3.745	200	1.972	2.601	3.34
25	2.06	2.787	3.725	300	1.968	2.592	3.323
26	2.056	2.779	3.707	500	1.965	2.586	3.31
27	2.052	2.771	3.69	1000	1.962	2.581	3.3
28	2.048	2.763	3.674	2000	1.96	2.576	3.291
29	2.045	2.756	3.659				

Table 4: Decision To Consider A Feature Is Significant If The Null Hypothesis (H_0) Is Rejected

Feature Name	**Absolute value of $T_{observed\_value}$**	**Degree of Freedom**	**$T_{critical\_value}$**	**Null Hypothesis (H0)**	**Alternate Hypothesis (H1)**	**Significant Feature**
Flow Duration	3.110481854	1.70165E+13	1.96	**Rejected**	Accepted	Yes

We specifically use the CSV version, which is the only version of the dataset that is partitioned to distinct training and test sets. Over 50 mill instances, 13 classes in total in the entire training archive. It consists of one class for normal, genuine traffic, and 12 different classes of DDoS attack traffic. The testing archive includes 8 classes, which contains one benign class with 20 million examples and 7 attack types. For statistical reasons (since one needs some data to experiment on), while ensuring computational efficiency, a sub-sample of few records were selected. This subset includes 399,998 training instances and 112,611 testing instances. To prepare these samples, a contiguous block of instances was chosen from the front of the entries for each class in the original CSV files.

Table 7: Training Dataset Used For Experimentation

Class Label	**No. of Instances**	**Category**
BENIGN	56,425	Legitimate Traffic
DNS	31,194	Protocol-Specific
LDAP	31,194	Protocol-Specific
MSSQL	31,194	Protocol-Specific
NetBIOS	31,194	Protocol-Specific
NTP	31,194	Protocol-Specific
SNMP	31,194	Protocol-Specific
SSDP	31,194	Protocol-Specific

SYN	31,194	Transport Layer
TFTP	31,194	Protocol-Specific
UDP	31,194	Transport Layer
UDP_Lag	31,194	Transport Layer
WebDDoS	439	Application Layer
Total	**3,99,998**	

The detailed distribution of instances across class labels for both the training and testing subsets is provided in Tables 7 and 8.

The CICDDoS2019 dataset is highly imbalanced because some attack categories have thousands of samples, while others (like WebDDoS) have very few. Training directly on such data can bias classifiers toward the majority classes. To address this, we used two strategies. First, during dataset construction we selected a balanced subset for testing, so evaluation was not skewed toward majority classes. Second, at the training stage, we applied stratified sampling so that both benign and attack classes were proportionally represented in each fold of cross-validation. These steps helped reduce bias and gave a more realistic measure of model performance.

Table 8: Distribution Of Instances In The Testing Dataset

Class Label	Instance Count	Classification
BENIGN	56,306	Legitimate
PortMap	9,072	DDoS Attack
LDAP	9,072	DDoS Attack
MSSQL	9,072	DDoS Attack
NetBIOS	9,072	DDoS Attack
SYN	9,072	DDoS Attack
UDP	9,072	DDoS Attack
UDP_Lag	1,873	DDoS Attack
Total	**1,12,611**	

Table 9: Cicddos2019 Dataset 60 Features Obtained After Initial Preprocessing

Serial Number of 60 Selected Features in 84 Features									
8	9	10	11	12	13	14	15	16	17
18	19	20	21	22	23	24	25	26	27
27	29	30	31	32	33	34	35	36	37
41	42	43	44	45	46	47	48	49	51
52	54	55	56	58	59	60	61	73	74
75	76	77	78	79	80	81	82	83	84

4.1 Pre-processing

Attributes that have fixed identifiers, offer limited value for recognizing the dynamic patterns inherent in DDoS attacks were removed. Subsequent analysis identified an additional 19 features that were either redundant or exhibited zero variance, leading to their removal. This process resulted in a refined dataset of 60 relevant features.

Table 10: 58 Features Obtained From T Test. Each Feature Mentioned With The Feature Number W.R.T. Order In The Dataset.

Serial No fo Features in the original dataset with 84 Features									
8	9	10	11	12	13	14	15	16	17
18	19	20	21	23	24	25	26	27	28
29	30	32	33	34	35	36	37	41	42
43	44	45	46	47	48	49	51	52	54
55	56	58	59	60	61	73	74	75	76
77	78	79	80	81	82	83	84		

Removed instances that contained these values since they cannot be used and trigger errors with many machine learning algorithms. Then the record of records class labels whose the invalid's number of entry was ratio less than 0.05% from the whole class content would be deleted based on class balance. Table 9 depicts the 60 features used for model training.

Tables 10 and 11 directly answer RQ1. In the former section, the independent T-test was able to identify 58 out of the 60 preprocessed features, to have a statistically significant difference between the two groups (benign and DDoS attack traffic); therefore, the independent T-test is instrumental in distinguishing one group from another. The smaller set of features preserved the most discriminative information although redundant or weakly informative attributes were eliminated, proving that statistically founded filter based method can be effectively used to support DDoS detection.

4.2 Feature Selection:

In ML, feature selection is paramount to the efficiency of every model. The commonly used Filter based methods such as IG and Chi-2, score features with respect to class labels, while Wrapper based methods such as Forward and Recursive Feature Selection, Iteratively Train a classifier and return the best feature subset. Here, we will be covering a T-Test based filter method. From the initial round of pre-processing, a total of 60 features were retained, which were then assessed by way of a T-Test to determine their significance in separating the classes. As a consequence, 58 features were chosen as the most relevant ones. Table 10 contains a comprehensive list of all selected features, indexed accordingly.

4.3 Normalization

After preprocessing, the first statistical summary of the dataset reveals the different scales of features – a tenfold difference between minimum and maximum of features. In fact, this imbalanced scale is a recognized overfitting risk factor: algorithms may start focusing on the size of the values instead of their correlate relationship. Next, since all feature values need to be in the same scale, we used min-max normalization on the features which put all feature values into a box between 0 and 1. This method (described in Equation 1) guarantees that all features are taken into account at an equal output proportionality at a time during the analysis.

$$\mathbb{X}' = \frac{\mathbb{X} - \mathbb{X}.\min()}{\mathbb{X}.\max() - \mathbb{X}.\min()} \qquad (6)$$

4.4 Data Transformation

A critical preprocessing step involved converting the multi-class problem into a binary classification task. This was achieved by re-encoding the class labels. In the training set, which initially contained 13 labels, the single 'benign' category was labeled as 0. Every other class, representing different attacks, was grouped under a single label of 1. The testing dataset underwent an identical conversion process, ensuring both datasets were aligned for building and evaluating a binary classifier.

4.5 Evaluation Metrics

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure2: Confusion Matrix: it is used to represent how the model predicted the test instances. To represent binary class classification predicted information used 2x2 square matrix, rows are representing actual class label and columns for predicted label.

		Predicted Class	
		1	0
Actual Class	1	**TP**	**FN**
	0	**FP**	**TN**

Figure2 Confusion Matrix

5. EXPERIMENTATION

Experimentation was done in 2 phases: first phase finding significant features and the second phase discriminate the attack and benign traffic. To validate the machine learning models applied cross fold validation method with 10 folds while train the models. To ensure reliable evaluation, we applied 10-fold stratified cross-validation on the training dataset. The data was divided into 10 equal subsets while maintaining the original class distribution in each fold. In each iteration, nine folds were used for training and the remaining fold was used for validation. This process was repeated 10 times, and the final performance was reported as the average across all folds. Stratification was chosen to handle class imbalance and ensure that both benign and attack samples were represented in every fold. Workflow of the feature selection and classification of network traffic depicted in figure 3.

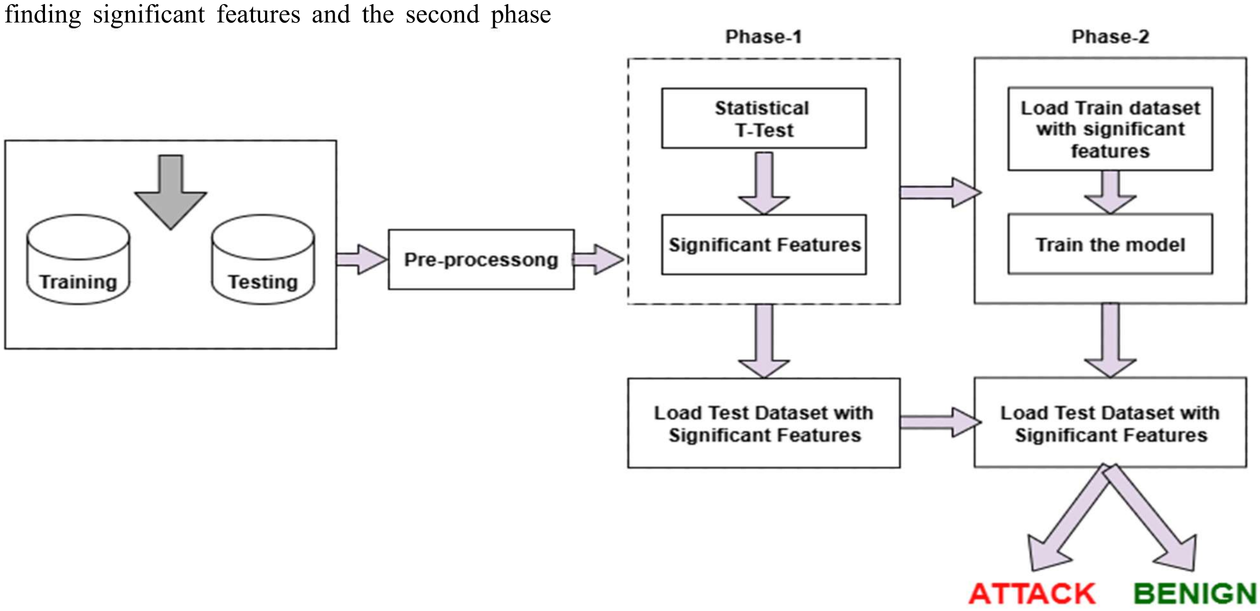

Phase-1
Phase-2
Training
Testing
Pre-processong
Statistical T-Test
Significant Features
Load Train dataset with significant features
Train the model
Load Test Dataset with Significant Features
Load Test Dataset with Significant Features
ATTACK
BENIGN

Figure 3: NIDS Frame Work with Feature Selection

Algorithm: To select significant features

```
Input: Dataset with N features
Output: N′ Significant features

NC ← Number of columns in dataset
CL ← Number of class labels in datase
1. Begin
2. N′ ← [ ]
3. if D(label) ==′ BENIGN′ do
        B ← D(label)
        n ← number of instances in B
```

```
  else:
        A ← D(label)
4. for i = 1 to CL do:
        CF ← [ ]
                        m ← number of instances in A of CL[i]
        for j = 1 to NC do:
              $\overline{X}_A$ = mean(A[j])
              $\overline{X}_B$ = mean(B[j])
              $(\sigma_A)$ = Variance(A[j])
              $(\sigma_B)$ = Variance(B[j])
              $T_{\text{Observed\_value}} = \frac{\overline{X}_A - \overline{X}_B}{\sqrt{\frac{(\sigma_A)^2}{n} + \frac{(\sigma_B)^2}{m}}}$
              if $T_{\text{Observed\_value}} > T_{\text{Table\_value}}$ do
                    CF ← A[j]
              end if
        if not in N′ do
              N′ ← CF
  end for
```

After performing feature selection method in phase 1 obtained 58 features, those are depicted in Table 10.

PHASE 2:

Step 1: load the training and testing datasets with significant features.

Step 2: separate the Input and out from the dataset

Step 3: normalize the input dataset

Step 4: train the machine learning classifier

Step 5: predict the test input traffic data

Step 6: classify the traffic data either attack or benign

Step 1: Load the Training and Testing Datasets with Significant Features

To build an effective DDoS attack detection model, we first need to gather network traffic data. This dataset typically contains network flow parameters such as packet count, source and destination IP addresses, protocol type, packet size, and timestamps. The dataset should include both normal and attack traffic labeled appropriately.

Step 2: Separate Input and Output from the Dataset

Once the dataset is loaded, we divide it into two parts:

Input Features (X): These are the network traffic attributes selected for training the model.

Output Labels (Y): This is the target variable that indicates whether the traffic is benign (0) or attack (1).

By structuring the dataset this way, the classifier can learn the relationship between input features and attack patterns.

Step 3: Normalize the Input Dataset

Since network traffic features often have different scales (e.g., packet size vs. number of connections), normalization is essential.

Step 4: Train the Machine Learning Classifier

With the prepared dataset, we train a machine learning model to recognize patterns in DDoS attacks. The model is trained using the training dataset, where it learns to distinguish between benign and attack traffic.

Step 5: Predict the Test Input Traffic Data

After training, the model is tested on unseen test data to evaluate its performance. The classifier processes network traffic and predicts whether each instance is a DDoS attack or normal traffic.

Step 6: Classify the Traffic Data as Either Attack or Benign

Finally, the model assigns a classification to incoming network traffic. If an instance is identified as an attack, appropriate mitigation measures (e.g., firewall rules, rate limiting, or IP blocking) can be deployed to prevent service disruption.

6. RESULTS AND DISCUSSION

In this section, we present the experimental results with respect to the research questions posed in Section 2.2. We perform an analysis of the performance of the proposed T-test– based feature selection method with respect to identifying statistically significant features (RQ1), enabling feature dimensionality reduction and lowering the computational cost of the detection process, while still achieving the same detection performance of sustained simulated attacks, (RQ2) and improving classification effectiveness, including an increased detection rate with an associated false positive rate, across a greater number of machine learning models per baseline classifiers(RQ3).

Table 12 results show performance of the model before applying the statistical T-Test and Table 13 results w.r.t to 58 features obtained from the T-Test. According to the Table 14 with 58 features among all the machine learning classifiers XGB classifier performed well with 99.82% detection rate and 97.40% accuracy. Proposed model performance is equal and higher compared to 60 features model performance. There is an improvement in detection rate and accuracy of XGB model using 58 features. It is stated that instead of 60 features, can use 58 features to discriminate the attack and benign traffic. In figure (5, 6) receiver operating system curves showing the performance of 58 and 60 features.

State-of-art we conducted experimentation with CICDDoS2019 dataset on traditional feature selection methods. In table 11 depicted best feature subset obtained from traditional feature selection methods and performance evaluation comparison with our proposed feature selection method. Our proposed model achieved 99.82% detection rate(sensitivity) and it is on par with the traditional feature selection methods. Merit of our model is selected significant features in very less time compare to traditional methods.

The comparative analysis in Table 12 confirms the advantage of the proposed feature selection method over contemporary models. It excels in reducing the False Positive Rate to 5% (a significant drop from the 18% in [19]) and surpasses the works of Sarafaldin et al. [4] and Hajimaghsoodi and Jalili [20] in detection rate, precision, and F-score. This enhanced performance is achieved with only a minimal decrease (2.08%) in accuracy relative to [19].

Table 11: Comparative Results Of Existing Feature Selection Methods On Cicddos2019 Dataset

Method	**Time taken for feature selection**	**Number of Features**	**Model**	**SN(%)**	**SP(%)**	**Pre(%)**	**Acc(%)**	**F_Score**
IG	1997.59s	40	Logistic Regression	99.46	96.88	96.95	98.17	98.19
Chi_2	166.04s	15	Random Forest	99.92	94.96	95.20	97.44	0.9750
FFS	18356s	47	Logistic Regression	99.48	96.95	97.02	98.21	0.9824
RFE	42928	35	XGB	99.82	97.43	97.49	98.62	0.9864

RFI	1271.58s	18	XGB	99.75	93.35	93.75	96.55	0.9666
LASSO	29252s	56	XGB	98.88	94.81	95.01	96.84	0.9691
Base Line Features	-	60	XGB	99.17	94.16	94.43	96.66	0.9674
Proposed	**3.87s**	**58**	**XGB**	99.82	94.98	95.21	97.40	0.9746

RQ2: The time for feature selection and the results of dimensionality reduction presented in Table 11. The traditional Information Gain, Chi-square, RFE and LASSO could take upwards of days for certain datasets, the proposed T-test–based method can achieve a significant reduction in feature selection time and can select similar or a lower number of features compared to traditional methods. It implies that our approach is able to achieves a trade-o between run time performance and detection ability, which is the basis that our approach is both scalable and practical for large-scale and real-time application of ID.

Table 12: This Table Shown The Performance Of Base Classifiers Against Cicddos2019 Dataset With 60 Features

Machine Learning Classifier	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F1_Score	Balanced Accuracy
DT	91.93	84.44	85.53	88.19	0.8861	88.19
LR	99.50	93.17	93.58	96.33	0.9644	96.33
NB	89.03	95.14	94.82	92.08	0.9183	92.08
RF	94.30	93.42	93.48	93.86	0.9388	93.86
LDA	99.26	88.10	89.29	93.68	0.9401	93.68
QDA	100.00	0.11	50.03	50.06	0.6669	50.06
XGB	99.17	94.16	94.43	96.66	0.9674	96.66

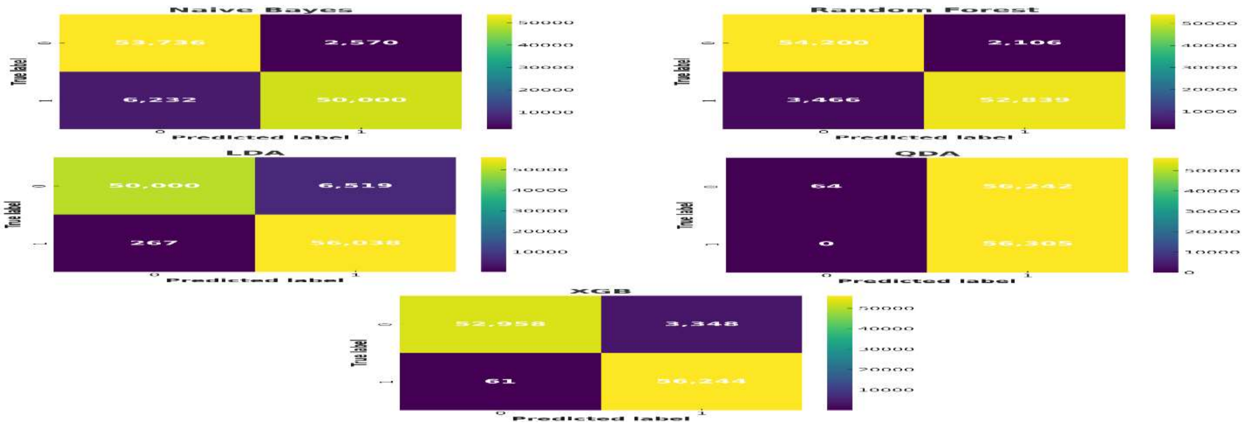
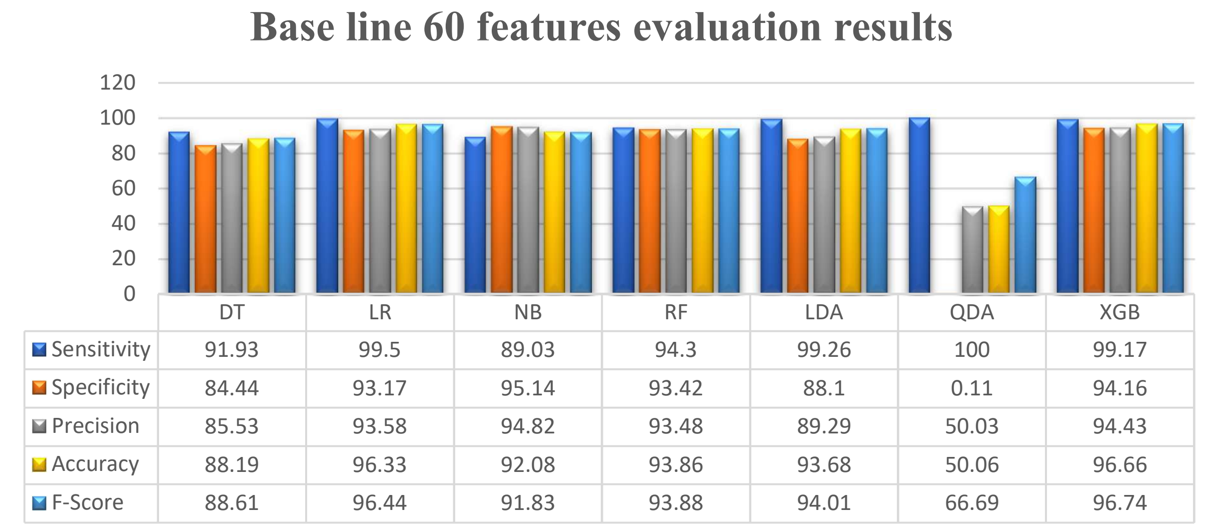


Figure 4 : Confusion Matrices For Base Classifiers Against Proposed Model Selected 58 Features Of Cicddos2019 Dataset

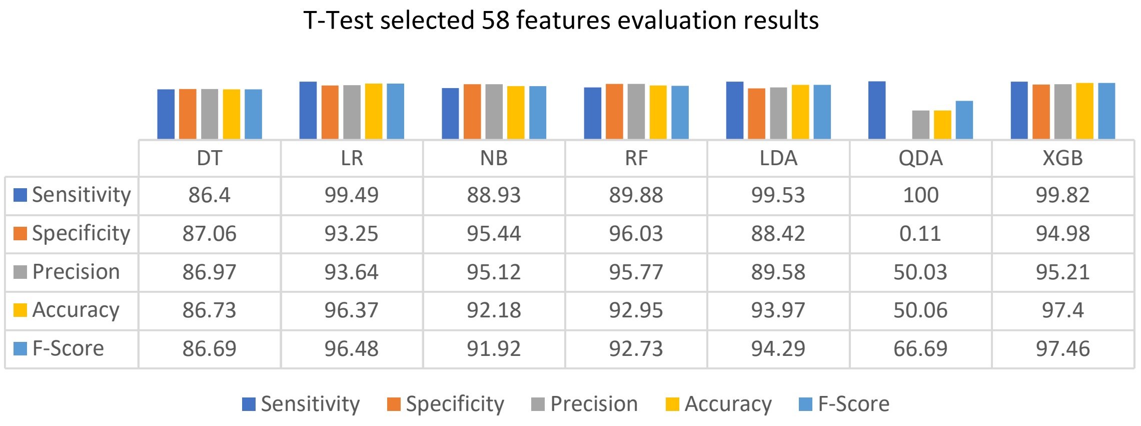


	DT	LR	NB	RF	LDA	QDA	XGB
Sensitivity	91.93	99.5	89.03	94.3	99.26	100	99.17
Specificity	84.44	93.17	95.14	93.42	88.1	0.11	94.16
Precision	85.53	93.58	94.82	93.48	89.29	50.03	94.43
Accuracy	88.19	96.33	92.08	93.86	93.68	50.06	96.66
F-Score	88.61	96.44	91.83	93.88	94.01	66.69	96.74

Figure 5: Base line 60 features evaluation results

Table 13: This Table Shown The Performance Of Base Classifiers Against Cicddos2019 Dataset With 58 Features Selected Through Proposed Model.

ML Classifier	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F1_Score	Balanced Accuracy
DT	86.40	87.06	86.97	86.73	0.8669	86.73
LR	99.49	93.25	93.64	96.37	0.9648	96.37
NB	88.93	95.44	95.12	92.18	0.9192	92.18
RF	89.88	96.03	95.77	92.95	0.9273	92.95
LDA	99.53	88.42	89.58	93.97	0.9429	93.97
QDA	100.00	0.11	50.03	50.06	0.6669	50.06
XGB	99.82	94.98	95.21	97.40	0.9746	97.40



	DT	LR	NB	RF	LDA	QDA	XGB
Sensitivity	86.4	99.49	88.93	89.88	99.53	100	99.82
Specificity	87.06	93.25	95.44	96.03	88.42	0.11	94.98
Precision	86.97	93.64	95.12	95.77	89.58	50.03	95.21
Accuracy	86.73	96.37	92.18	92.95	93.97	50.06	97.4
F-Score	86.69	96.48	91.92	92.73	94.29	66.69	97.46

Figure 6: T-Test Selected 58 Features Evaluation Results

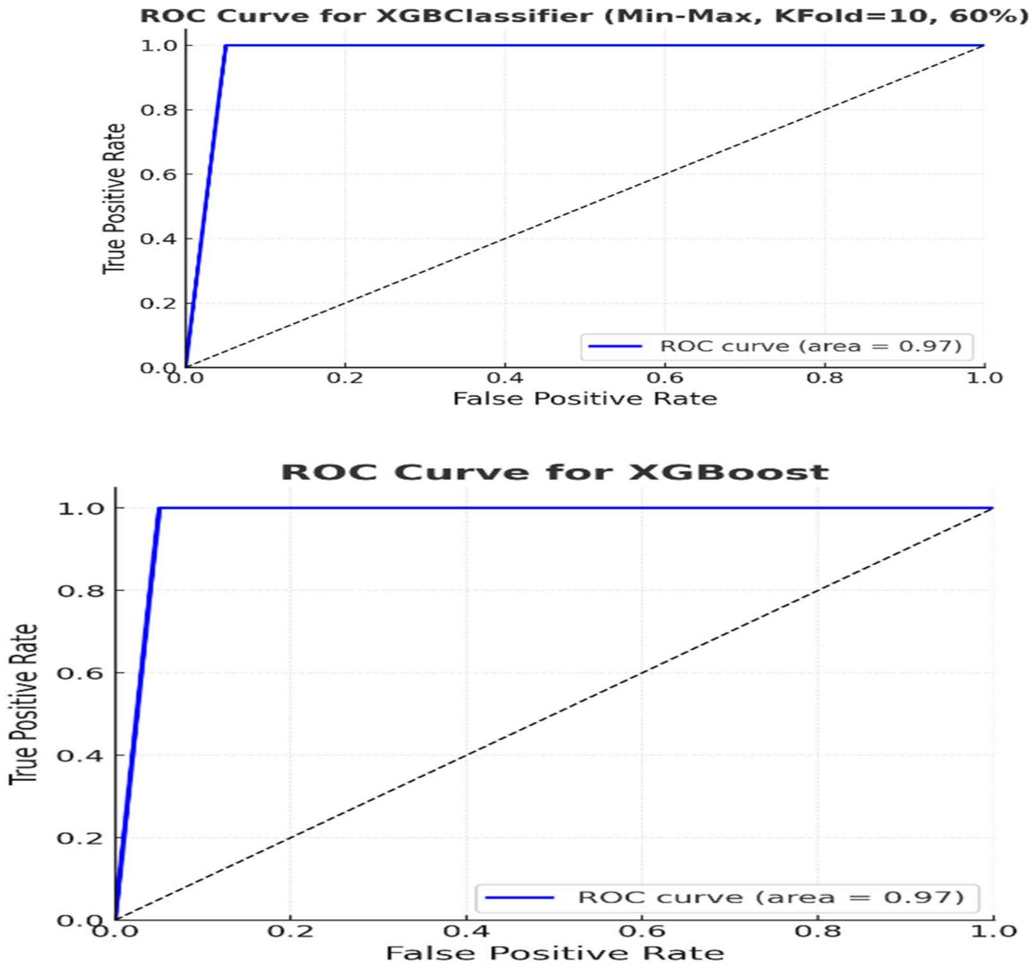


Figure 7: ROC curve for XGB model with 58 features

Table 14 compares the detection rate, accuracy, balanced accuracy, and false positive rate (FPR) of our proposed method with several existing approaches that also used the CICDDoS2019 dataset.

The results show that our model achieved a detection rate of 99.82% and an accuracy of 97.40%, which are competitive with or higher than most recent studies.

Importantly, our model reduced the FPR to 5.55%, which is significantly lower than the 18.5% reported by Alashhab et al. [19]. While some deep learning methods (e.g., Cil et al. [25], Maiga et al. [26]) reached very high detection or accuracy values, they generally require more complex models and longer training times.

In contrast, our T-test based feature selection combined with XGBoost gives a simpler and faster solution, while still maintaining strong detection performance.

Overall, Table 14 highlights that our method achieves a good balance between accuracy, detection rate, and efficiency, making it suitable for practical deployment.

Table 14: Comparison With Existing Works

Author	**Detection rate(%)**	**Accuracy (%)**	**Balanced Accuracy (%)**	**FPR (%)**
Sharafaldin, I[4]	65	NA	NA	NA
A. A. Alashhab et al[19]	98.81	**98.70**	NA	18.5
Hajimaghsoodi and Jalili [20]	99	**NA**	NA	NA
Jia et. al [22]	99.31	**98.90**	NA	NA
J Halledy et. al [23]	NA	**98.58**	100	NA
Almari et. al [24]	100	**99.9**	NA	NA
Cil et. al [25]	99.98	**99.97**	NA	NA
A.A Maiga et. al[26]	99.73	**99.76**	NA	0.046
Base line Feature subset	99.17	96.66	96.66	5.84
Proposed work	**99.82**	97.40	**97.40**	**5.55**

RQ3: The classification results achieved with the selected 58 features gives an unambiguous answer to RQ3. Tables 12–14 and Figures 6–8 show that the XGBoost classifier provided the highest detection rate (99.82%), balanced accuracy (97.40%) and a low false positive rate (5.55%) in relation to baseline feature sets and many other approaches. Statistical features selection enhances classifier performance As revealed by these results, the statistical features selected positively affect the performance of the classifier, especially by decreasing false alarms while maintaining high detection capability for many kinds of attacks.

The experimental results collectively indicate that this feature selection method fulfills the requirements of the research objectives outlined in this study, whereby detection efficiency is improved, computational cost is lowered, and classification reliability is boosted.

6.4 Critical Evaluation with Respect to Research Goals and State-of-the-Art

The Table 15 is the conclusive comparison of the proposed method in alignment to the original objectives and the contemporary state-of-the art solutions. The main focus of this work is to create a statistically based yet computationally efficient feature selection method for DDoS detection. Although some deep thinking-based methods reach slightly better performance, it comes with a high degree of computational expense and lack of interpretability. On the other hand, the T-test-based method shows competitive detection performance compared with the state-of-the art, while at the same time providing a clearer and with considerably reduced feature selection time. With this analysis, we illustrate that our proposed method, which focuses on efficiency, reproducibility and practical deployability, presents a reasonable trade-off compared to complex state-of-the-art models.

7. PROBLEMS AND OPEN RESEARCH ISSUES

Although the proposed computationally efficient T-test–based feature selection method shows an effective detection performance but still several research issues are open. The present work formulates the problem of DDoS detection as a binary classification problem and it fails to differentiate between classes of attacks which hinders fine-grained analysis. Furthermore, the independent T-test considers each feature individually and does not reflect possible interactions between the features. It is also tested in an offline scenario and does not directly incorporate CHANGED types of attack or concept drift in real network traffic. In addition, validation on a single dataset only, and the generalizability and clinical implementation should be validated in larger cross-dataset and real-time environments. These limitations point to important avenues for future research.

Table 15. Critical Comparison Of Proposed Method With Research Objectives And State-Of-The-Art Solutions

Study / Method	**Feature Selection Strategy**	**Model**	**Dataset**	**Detection Performance**	**Computational Complexity**	**Interpretability**	**Limitations / Critique**
Sharafaldin et al. [4]	Visualization-based (Radviz)	ID3	CICDDoS2019	Low detection rate (65%)	Low	High	Poor detection accuracy; not scalable for real-world deployment
Alashhab et al. [19]	Statistical (unspecified)	Online ML Ensemble	CICDDoS2019	High DR (98.83%)	Moderate–High	Medium	Feature selection process not clearly defined; high FPR (18.5%)
Hajimaghsoodi & Jalili [20]	Statistical thresholding	RAD	CICDDoS2019	High DR (99%)	Moderate	Medium	Number of features not reported; lower precision
Jia et al. [22]	Manual feature grouping	LSTM	CICDDoS2019	High accuracy (98.9%)	High	Low	Deep learning complexity; limited interpretability
Cil et al. [25]	No explicit FS	DNN	CICDDoS2019	Very high accuracy (99.97%)	Very High	Low	Black-box model; high training cost
Maiga et al. [26]	Not specified	BiLSTM+LSTM	CICDDoS2019	Very high accuracy (99.76%)	Very High	Low	Complex architecture; difficult real-time deployment
Proposed Method	**Statistical T-Test (explicit)**	**XGBoost**	**CICDDoS2019**	**DR 99.82%, Acc 97.40%**	**Low**	**High**	Binary classification only; feature interactions not modeled

8. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this paper, we proposed a feature selection approach for DDoS attack detection based on the independent T-test all features compared by the T-test shown high performance against to state of art methods, our proposed approach is designed statistically to address the challenges of high dimensional and imbalanced nature of network traffic dataset. The traditional filter, wrapper, and embedded feature selection techniques suffer from their ineffectiveness for optimization due to their inherent limitations, therefore, the proposed method proposes to identify the smallest subset of statistically significant features, which perfectly separates benign from attack traffic in new network environments.

Experimental results showed that the proposed method successfully satisfies the research objectives described in this work. In particular, the T-test–based feature selection method obtained 58 discriminative features from CICDDoS2019 dataset that resolved the problem of feature redundancy and noise. Using numerous machine learning classifiers, the prediction rate was 99.82% while the accuracy score was 97.40% and the balanced accuracy score was 97.40% along with a low false positive rate of 5.55% using XGBoost model. The results show that statistically selected features can successfully preserve or improve on detection performance whilst reducing both feature dimensionality and computational burden.

Our method achieves a better balance of effectiveness and computational efficiency in comparison to baseline state-of-the-art approaches reported in the literature. The resulting approach matches or outperforms many state-of-the-art models, while at the same time maintaining interpretability and drastically reducing feature selection time. This idealizes the method for practical intrusion detection systems demanding fast, reliable, and clear decision mechanisms.

There are some ways for extending the work as the future research directions. Our contributions are twofold: First, the developed statistical feature selection framework can be used for multi-class DDoS attack classification to identify different types of attack. Second, this approach could work with online or streaming traffic to make it realtime DDoS detection capable in dynamic networks. Finally, we can consider the incorporation of statistical feature selection in adaptive or hybrid learning models to provide further robustness against changing attack patterns and concept drift in large scale networks.

Conflicts of interest

The authors declare no conflict of interest

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing original draft preparation, writing review and editing, visualization, have been done by 1st author. The supervision and project administration have been done by 2nd author.

REFERENCES

[1] https://www.clickz.com/internetgrowthusage-stats-2019-time-onlinedevices-users/235102/

[2] Diro, A. A., and Chilamkurti, N, “Distributed attack detection scheme using deep learning approach for Internet of Things”, *Future Generation Computer Systems*, Vol 82, pp. 761-768, 2018.

[3] Sadique, K. M., Rahmani, R., & Johannesson, P, “Towards security on internet of things: applications and challenges in technology”, *Procedia Computer Science*, Vol. 141, pp. 199-206, 2018.

[4] Sharafaldin, I., Lashkari, A. H., Hakak, S., and Ghorbani, A. A, “Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy”, *In 2019 international carnahan conference on security technology (ICCST),* pp. 1-8, IEEE, 2019.

[5] Johnson, Richard A., Irwin Miller, and John E. Freund, *Probability and statistics for engineers*, vol. 2000. 2000.

[6] Raghupathi Manthena, Radhakrishna Vangipuram, “A Comprehensive Research Study on Evaluating Intrusion Detection Datasets for DDoS Attack Detection”, *In Information Security, Privacy and Digital Forensics. ICISPD 2023*, London, Pearson Education,. *Lecture Notes in Electrical Engineering, Springer, Singapore,* 2023.

[7] A. Gharib, I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "An Evaluation Framework for Intrusion Detection Dataset," *2016 International Conference on Information Science and Security (ICISS)*, *Pattaya, Thailand,* pp. 1-6, 2016.

[8] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada,* pp. 1-6, 2009.

[9] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. "Toward generating a new intrusion detection dataset and intrusion traffic characterization", *In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018),* pp. 108-116, 2018

[10] Tan Z, Jamdagni A, He X, Nanda P, Liu RP, "A system for denial-of-service attack detection based on multivariate correlation analysis", *IEEE Transactions on Parallel and Distributed Systems,* Vol. 25, No. 2, pp. 447–456, 2014.

[11] Hoque, Nazrul, Dhruba K. Bhattacharyya, and Jugal K. Kalita. "A novel measure for low-rate and high-rate DDoS attack detection using multivariate data analysis", *2016 8th International Conference on Communication Systems and Networks (COMSNETS). IEEE,* 2016.

[12] Hoque, Nazrul, Dhruba K. Bhattacharyya, and Jugal K. Kalita. "FFSc: a novel measure for low-rate and high-rate DDoS attack detection using multivariate data analysis", *Security and Communication Networks,* Vol. 9, No. 13, pp. 2032-2041, 2016.

[13] S. Sarvari, N. F. Mohd Sani, Z. Mohd Hanapi and M. T. Abdullah, "An Efficient Anomaly Intrusion Detection Method With Feature Selection and Evolutionary Neural Network," *IEEE Access,* Vol. 8, pp. 70651-70663, 2020.

[14] N. Singh and A. Kaur, "Feature selection for artificial neural network based intrusion detection system", *Int. J. Technol. Res. Eng.,* Vol. 2, No. 11, pp. 2681-2683, 2015.

[15] M. H. Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization", Int. J. Netw. Secur., vol. 18, no. 3, pp. 420-432, May 2016.

[16] Gaurav Tripathi, Vishal Krishna Singh, Varun Sharma, Majithia Vivek Vinodbhai, "Weighted Feature Selection for Machine Learning Based Accurate Intrusion Detection in Communication Networks", *IEEE Access,* Vol. 12, pp. 20973-20982, 2024.

[17] NETSCOUT DDoS Threat Intelligence Report / January 2025 to June 2025, URL: https://www.netscout.com/threatreport/ddos-attack-vectors/

[18] Kshirsagar, D., Kumar, S, "A feature reduction based reflected and exploited DDoS attacks detection system", *J Ambient Intell Human Comput,* Vol. 13, pp. 393–405, 2022.

[19] Alashhab, Abdussalam A., et al. "Enhancing DDoS Attack Detection and Mitigation in SDN Using an Ensemble Online Machine Learning Model", *IEEE Access,* Vol. 12, pp. 51630 – 51649, 2024.

[20] M. Hajimaghsoodi and R. Jalili, "RAD: A Statistical Mechanism Based on Behavioral Analysis for DDoS Attack Countermeasure", *IEEE Transactions on Information Forensics and Security,* Vol. 17, pp. 2732-2745, 2022.

[21] Johnson, Richard A., Irwin Miller, and John E. Freund, "Probability and statistics for engineers", London: Pearson Education, Vol. 2000. 2000.

[22] Y. Jia, F. Zhong, A. Alrawais, B. Gong and X. Cheng, "FlowGuard: An Intelligent Edge Defense Mechanism Against IoT DDoS Attacks,", *IEEE Internet of Things Journal,* Vol. 7, No. 10, pp. 9552-9562, 2020.

[23] J. Halladay et al., "Detection and Characterization of DDoS Attacks Using Time-Based Features," *IEEE Access*, Vol. 10, pp. 49794 - 49807, 2022.

[24] H. A. Alamri and V. Thayananthan, "Bandwidth Control Mechanism and Extreme Gradient Boosting Algorithm for Protecting Software-Defined Networks Against DDoS Attacks", *IEEE Access,* Vol. 8, pp. 194269-194288, 2020.

[25] Cil, Abdullah Emir, Kazim Yildiz, and Ali Buldu. "Detection of DDoS attacks with feed forward based deep neural network model." Expert Systems with Applications 169 (2021): 114520.

[26] A. -A. Maiga, E. Ataro and S. Githinji, "Intrusion Detection With Deep Learning Classifiers: A Synergistic Approach of Probabilistic Clustering and Human Expertise to Reduce False Alarms," in IEEE Access, vol. 12, pp. 17836-17858, 2024.

[27] A. Zainudin, L. A. C. Ahakonye, R. Akter, D. -S. Kim and J. -M. Lee, "An Efficient Hybrid-DNN for DDoS Detection and Classification in Software-Defined IIoT Networks", *IEEE Internet of Things Journal,* Vol. 10, No. 10, pp. 8491-8504, 2023.