The Impact of the Feature Selection methods for prediction DDoS attacks

Hady Hammad, Lobna Elsaid, Abdallah Rashed, and Ragab Godah
School of Electrical Engineering and Computer Science (EECS)
University of Ottawa

Hammad, H.
Email: hhamm043@uottawa.ca
Elsaid, L.
Email: lelsa034@uottawa.ca
Rahsed, A.
Email: arash015@uOttawa.ca
Godah, R.
Email: rgoda036@uottawa.ca

Abstract—**Machine Learning is now an important approach used as a defensive layer against cyber security threats, Especially with DDoS attacks. To build a robust model that can efficiently detect the attacks and threats we need to focus on the preprocessing step specifically feature selection techniques. We decided to focus on the different approaches for feature selection techniques that can be implemented within a machine Learning lifecycle to gain an efficient model. First, we compared four different approaches: Univariate selection, Feature importance, game theoretic approaches for feature selection, and correlation matrix with heatmaps. Then we chose the most suitable one for our problem. And implemented it within a model that detects the Malicious traffic of DDoS attacks. This paper is a comparative approach of different used feature selection techniques implemented in different cases and presents a champion model.**

Keywords— DDoS attack, Machine Learning, Feature Selection, Univariate Selection, Feature Importance, Correlation Matrix, SHapley Additive exPlanations "SHAP".

I. Introduction

The Internet is one of the essential tools in everyday activities. The usage of the internet becomes a side brick for every corporation and startups. As usual, by increasing the usage of anything the challenges also increase. But with the internet the most important challenges are those which are related to security and user privacy. DDoS attacks are one of the security challenges that threaten the availability of any system. Distributed Denial of service attacks "DDoS" is a systematic attack that targets a corporation servers or network with a huge number of requests, that exceeds the computational power of the server, causing the server to be down and sometimes it gets the whole network down. One of the most effective proposed solutions is using machine learning to detect the malicious traffic that leads to discovering if the network is under attack.

Machine Learning is one of the trendy Data Science approaches that can be used to be learned and then classify and detect any malicious requests. And due to its ability to learn from the patterns and classify or cluster the future traffic as malicious or benign, it became an important approach to consider when dealing with such evolved threats. As seen in figure 1, machine learning is composed of multiple steps which is referred to as the machine learning life cycle. We will focus on the feature selection step. Since the data used is a high dimensional data, and to avoid the trap of the high dimensional data known as curse of dimensionality. So, we need to

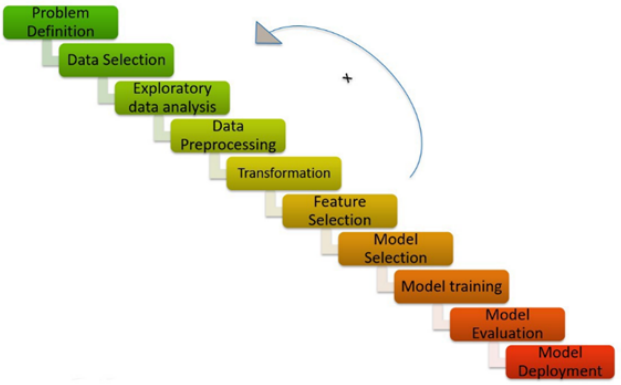

Figure 1: Machine Learning Lifecycle (Ignite, TCS)

Decrease these dimensions in a scientifically reasonable approach. As this high dimensional data may be considered as misleading for the model. Feature selection techniques is an important approach to consider, as it will help us to depend only on the most important features, which will help in reducing the time needed and the computational cost for the problem being solved. It also helps to increase the performance in the means of accuracy and efficiency.

There are many feature selection techniques. We will focus on a previously used approach: Univariate selection, Feature importance, correlation matrix with heatmaps and a new proposed solution by SHAP (Lundberg, 2018) for feature selection. Comparing all these techniques will give an intuition of how feature selection is important, and the most important features that can be used to help in defending against the DDoS attack. Finally, we will have a model based on the features selected. In Section II we will present the main purpose of the project. In section III, we will review the previous literature and related work done on feature selection for DDoS attacks.

II. Project Description

Nowadays, Machine learning contributes to every problem in our life. Machine Learning has a great approach in dealing with and detecting cyber security attacks. Specifically, machine learning is very effective in detecting DDoS attacks. Feature selection techniques used in the preprocessing phase of any machine learning project, has a great effect in enhancing the performance by increasing the rate of the detection accuracy, and decreasing the required computational power by reducing the overall execution time.

Feature selection will allow using the minimum number of the features with the most impact on the model to detect the attacks. Our goal is to create a model to select the best features to give the best performance. We will do that through a comparative approach on different feature selection techniques such as Univariate selection, Feature importance, game theoretic approaches for feature selection, and correlation matrix with heatmaps. We will test each approach and then provide the pros and cons of each. Finally, we will use only one model to make sure that we only compare the feature selection techniques.

III. Dataset Description

CICDDoS2019 [1]. Contains benign and the most up-to-date common DDoS attacks, which resembles the true real-world data (PCAPs).

It also includes the results of the network traffic analysis using CICFlowMeter-V3 with labeled flows based on the time stamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files). The dataset has been organized per day. For each day, they recorded the raw data including the network traffic (Pcaps) and event logs (windows and Ubuntu event Logs) per machine. In the features extraction process from the raw data, they used the CICFlowMeter-V3 and extracted more than 80 traffic features and saved them as a CSV file per machine.

And we can see on figure[3] how data is collected daya by day. We use this data especially because of usage of CICFlowMeter-V3 and how it generates the whole data. Let us take a look at data generation: CICFlowMeter is a network traffic flow generator and analyser. It can be used to generate bidirectional flows, where the first packet determines the forward (source to destination) and backward (destination to source) direction., hence more than 80 statistical network traffic features such as Duration, Number of packets, Number of bytes, Length of packets, etc. can be calculated separately in the forward and backward directions.

Days	Attacks	Attack Time
First Day	PortMap	9:43 - 9:51
	NetBIOS	10:00 - 10:09
	LDAP	10:21 - 10:30
	MSSQL	10:33 - 10:42
	UDP	10:53 - 11:03
	UDP-Lag	11:14 - 11:24
	SYN	11:28 - 17:35
Second Day	NTP	10:35 - 10:45
	DNS	10:52 - 11:05
	LDAP	11:22 - 11:32
	MSSQL	11:36 - 11:45
	NetBIOS	11:50 - 12:00
	SNMP	12:12 - 12:23
	SSDP	12:27 - 12:37
	UDP	12:45 - 13:09
	UDP-Lag	13:11 - 13:15
	WebDDoS	13:18 - 13:29
	SYN	13:29 - 13:34
	TFTP	13:35 - 17:15

Table[1] Dataset description

IV. STATE-OF-THE-ART-REVIEW

In this section we are going to discuss our literature review by taking a summary of the most used feature selection (FS) methods in predicting Denial Of Service Attack (DDOS) and how their impact on the results of machine learning models by making comparisons before and after using these FS methods.

First Paper we will discuss is "DOS-DDOS Attacks Predicting: Performance Comparison of The Main Feature Selection Strategies" [1]. In this study they decide to take in consideration different data sets with many challenges as cyber-attacks networks traffic generate high dimensionality attributes with redundant, irrelevant information. The below table includes the used datasets.

Dataset	Creation Year	Dimensionality	Number of attacks types
KDD'99	1999	41	4
NSL_KDD	2009	41	4
UNSW_NB15	2015	48	9
CIC_IDS 2017	2017	78	6
CIC_IDS 2018	2018	79	6

Fig[4]. Datasets

Table 1: Cyber-attacks Datasets Dimensionality (Kawtar Bouzoubaa, 2022)

So, they focus on how to extract the most important security information in terms of ML model performance, execution time and generalization by using these feature selection types.

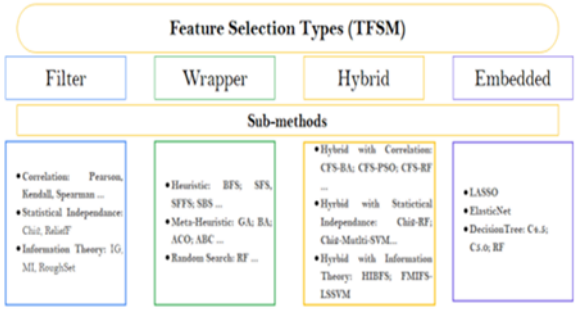

Figure 2: TFSM used in cyber security ML Models (Kawtar Bouzoubaa, 2022)

Then they record performance analysis for each sub method of each category of these methods (Filter, Wrapper, Hybrid, Embedded), and these are the performance metrics used in this study.

1- Filter sub methods include Correlation, Statistical Independent (SI), Information Theory (IT) and the Combined Filter methods (CF), the best outputs were 17 best DDOS features selected by the CFS Filter sub-method record best filter strategy precision = 99.98% as in table 3.

2- Wrapper sub methods include Heuristic (HE), Meta-Heuristic (MH) and Random Search (RS), as shown in table 4 the best outputs were by selecting 20 DOS-DDOS attacks features. The maximum precision 99.97% were recorded by the Meta-Heuristic category.

3- While in table 5 Hybrid sub-methods include Filter CFS, Statistical Filter (SF) and Information Theoretical (IT), by selecting 17 features the best precision value (99.9%) was recorded on CFS and Naïve Bayes (HFSN).

4- Embedded sub-methods include Least Absolute Shrinkage and Selection Operator (LASSO), Embedded Ensemble Optimal Feature Selection Algorithm (EEOFSA) and DT based Embedded as shown in table 6. While in table 7 they used best features to get the best ML model among these models (Naïve Bayes classifier, MI and generalized entropy) for each dataset in huge comparison to get best performance with best extracted features as below. And finally in Fig. 3 and Table. 4, we can conclude results for all methods used in this paper by comparing their best accuracy and time taken during training and testing.

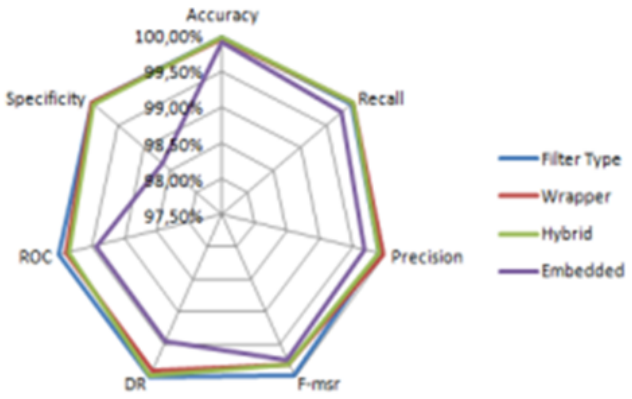

Figure 3: Best performances of the Main types of Feature Selection Methods (TFSM) (Kawtar Bouzoubaa, 2022)

	TRAIN_TIME (SECOND)	TEST_TIME (SECOND)
FILTER	0.0001	0.0006
WRAPPER	0.00041	0.21
HYBRID	0.03	0.09
EMBEDDED	0.6385	98.44

Table 4: Best performance values of train and test time (Kawtar Bouzoubaa, 2022)

In this paper [4] they consider the KDD cup 99 dataset which consists of 41 features, applying many preprocessing as these dataset attributes to remove nulls and redundant, categorical attributes changed into numerical, they used a new strategy called Crow Search Algorithm (CSA) and Opposition based Learning (OBL), CSA is meta heuristic approach to search for important features same as crows hide their cheating crows (N), 1 of 2 discussions (t) will be taken, first: if crow succeeded in hide his food so no need to update his position, second: if he discovered that it's already known by the cheating crow he will update his position by this n Eq. (1) where, Rm stands for random number in the range [0,1], FLtm represents flight length of crow m at iteration.

$$S\ t+1\ m = S\ t\ m + Rm * FLt\ m * (B\ t\ n - S\ t\ m) \quad (1)$$

Then, OBL is used to optimize solutions of CSA by producing opposite solutions (track nodes in opposite way from end to start) and test if it will give the same solution or not if not, so this solution will be updated by the best position nearer to optimal solution discovered by using Eq. (2). where, P t n is the awareness probability of crow n at iteration t. Positions are updated.

S t+1 m = { if Rn ⩾ P t n update position by using Eq. (1) else update to random position} (2)

And the pseudo-code for OCSA is in **Fig. 4**

```
Start
Define population size (N), the extreme quantity iteration (K), Flight length (FL) and awareness probability P_n
Randomly initialize positions of the flock of crows (refer Fig. 4)
Calculate the opposite location of the flock of crows (refer Eq. (6))
Initialize and record the crow's memories.
Set the iteration count k = 1
While the termination criteria is not adhered to do
  For i = 1...N
    Arbitrarily select one of the crows to follow (n^th crow)
    Create R_n
    If R_n ≥ P_n^t
      Update solution using Eq. (1)
    Else
      Update random position
    End If
  End For
  Compute the latest location of the crows
  Update the memories of crow individuals
End while
Select the best position and visualize the results
End
```

Figure 4: Pseudocode for proposed OCSA based feature selection

The top features extracted by OCSA strategy are shown in table 9. Then they used these best features as an input to RNN classifier to predict DDOS attacks and compared it to other feature extraction models to evaluate the proposed algorithm (OCSA), and here is the result below.

Then after OCSA prove that it helps to extract best features, they decide to compare different algorithms rather than RNN to compare them with the proposed one (RNN) to show different performances based on intrusion detection system as Artificial Neural Network (ANN), K-nearest neighbor classifier (KNN) and naive bayes classifier by their best feature extraction model (OCSA). And here is their result by comparing all different classifiers.

Feature number	Feature name	Description	Type
1	Duration	Time period of connection	Continuous
6	Destination bytes	Bytes delivered to source from destination	Continuous
12	Logged in	1 if successfully logged in, 0 otherwise	Discrete
16	#root	Number of root access	Continuous
19	#access files	Number of operations performed on access control files	Continuous
20	#outbound commands	Number of outbound commands occur in an ftp session	Continuous
21	Is_host_login	1 if login belongs to host list, 0 otherwise	Discrete
31	srv_diff_host_ rate	% of connections to different hosts	Continuous
32	dst_host_count	Count of connections having the same destination host	Continuous
37	dst_host_srv_diff_host_rate	% of connections to the same service coming from different hosts	Continuous

Table 5: Features selection using OCSA (SaiSindhuTheja & Shyam, 2021)

Methods	Precision	Recall	F-measure	Accuracy
GA+RNN	93.02	89.40	88.45	89
FA+RNN	94.71	90.24	89.35	89.52
CSA+RNN	95.52	91.56	90.14	91.26
OCSA+RNN (Proposed)	98.18	95.13	93.56	94.12

Table 6: Percentage values of various methods represented (SaiSindhuTheja & Shyam, 2021)

After we see a lot of feature selection methods, now we will discuss the basic feature extraction models (Filter, Wrapper, Embedded) from another perspective of reviewing in this paper "Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models". Their strategy depends heavily on 2 factors which they are cost and time as it shown in Fig [6], and as long as DDoS attack datasets has many features, they decide to go on feature extraction models' approach and applying on different ML (machine learning) & DL (deep learning) models as (SVM, KNN, ANN, NB (naïve bayes).

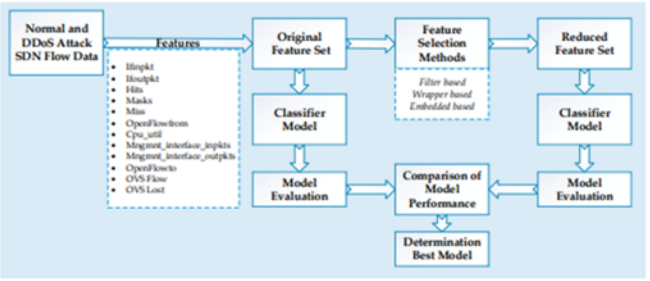

Figure 6: Process steps for applying the feature selection methods and machine learning models (Polat et al., 2020)

1- Filter method:

They use only RA (Relief Algorithm) sub method from filter model depending on the rule of thumb which says that (RA) gives best results of the set extracted from the whole features exist because it measures the distance between each feature and all classes and calculates the relation of each sample with other instances in its class.

2- Wrapper method:

They used a wrapper method to get best features depending on comparing it to all features till they reach the best accuracy but maybe that will affect their cost and time which they didn't want to reach. As They want to reach high values, they decide to use the forward selection sub method as one of wrapper algorithms as it moves each subset of features till the end of classification which will reach the highest values.

3- Embedded method:

They finally used an embedded method as it combines both filter and wrapper methods to get best features by learning algorithm classification. They used Lasso Algorithm depending on the concept of eliminating the features which have least weights till they reach the reduced features wanted.

Then they try their algorithms on the reduced features, shuffle between approaches and models till reach best results and compare them to the models' performance before using feature selection methods, so before using any feature selection approaches these were the models results.

But after applying Filter feature selection these were their results precision and f1-score didn't change that much except maybe for ANN and NB.

But after applying wrapper results were better than before applying feature selection, more obvious in KNN model and even better than filter selection results.

And finally, after applying embedded it wasn't different that much from wrapper as embedded as we said is combination between filter and wrapper till reach best results so it almost same results as wrapper. Then we can conclude all performances for all models in more generic comparison**.** And we can see that best results were obtained by applying KNN model with a wrapper method with 6 features on the test dataset. These results are compared in table 11.

Classifier	Method	No. of Features	Accuracy	Sensitivity	Specificity	Precision	F1_Score
SVM	-	12	92.11%	88.71%	96.93%	91.42%	89.91%
	Filter	10	92.46%	89.13%	97.02%	92.02%	90.43%
	Wrapper	8	92.15%	90.20%	97.26%	90.23%	90.21%
	Embedded	10	92.46%	90.73%	97.41%	90.49%	90.60%
KNN	-	12	95.67%	93.87%	98.01%	97.05%	95.30%
	Filter	6	97.15%	95.88%	98.68%	98.10%	96.92%
	Wrapper	6	98.30%	97.73%	99.45%	97.72%	97.70%
	Embedded	8	96.30%	94.95%	98.85%	95.09%	94.80%
ANN	-	12	91.07%	87.27%	96.58%	89.89%	88.45%
	Filter	6	92.28%	89.02%	96.99%	91.62%	90.20%
	Wrapper	10	91.44%	87.82%	97.31%	88.11%	87.89%
	Embedded	6	92.09%	88.91%	97.42%	89.22%	89.06%
NB	-	12	94.48%	91.77%	98.29%	92.94%	91.79%
	Filter	8	95.70%	93.49%	98.65%	95.07%	93.60%
	Wrapper	10	94.87%	92.05%	98.43%	93.29%	92.01%
	Embedded	10	95.09%	93.34%	98.45%	93.44%	93.18%

Table 7: performance of the machine learning models with different techniques (Polat et al., 2020)

VI. METHODOLOGY

We implement many Feature Selection techniques to measure the influence of these methods on DDos Attack classification problems. the CICDDoS2019 dataset was chosen as it mentioned before This set includes labeled traffic samples from 12 modern DDoS attacks (NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, TFTP), in addition to benign traffic.

Binary		Multiclass	
Class	Samples	Class	Samples
ATTACK	50006249	TFTP	20082580
		DrDoS_SNMP	5159870
		DrDoS_DNS	5071011
		DrDoS_MSSQL	4522492
		DrDoS_NetBIOS	4093279
		DrDoS_UDP	3134645
		DrDoS_SSDP	2610611
		DrDoS_LDAP	2179930
		Syn	1582289
		DrDoS_NTP	1202642
		UDP-lag	366461
		WebDDoS	439
BENIGN	56863	BENIGN	56863

Table 8: Class Distribution of the Dataset CICDDOS 2019

We use an ensemble technique which is the CatBoost classifier. We are going to clarify it later after FS techniques, as a part of treating unbalancing data and F1 score in the evaluation to test the actual evaluation without any biasing to class more than the other.

a) Univariant Selection:

SelectKBest works by retaining the first k features of X with the highest scores. It takes as a parameter a score function, which must be applicable to a pair (X, y). The score function must return an array of scores, one for each feature $X[:,i]X[:,i]$ of X (additionally, it can also return p-values, but these are neither needed nor required). SelectKBest then simply retains the first k features of X with the highest scores. SelectKBest has a default behavior implemented, so you can write select = SelectKBest() and then call select.fit_transform(X, y) (in fact I saw people do this). In this case SelectKBest uses the f_classif score function.

This interpretes the values of *yy* as class labels and computes, for each feature $X[:,i]X[:,i]$ of X, an F-statistic. The formula used is exactly the one given here: one way ANOVA F-test, with K the number of distinct values of y. A large score suggests that the means of the K groups are not all equal. This is not very informative, and is true only when some rather stringent conditions are met: for example, the values $X[:,i]X[:,i]$ must come from normally distributed populations, and the population variance of the *K*K groups must be the same. I don't see why this should hold in practice, and without this assumption the F-values are meaningless.

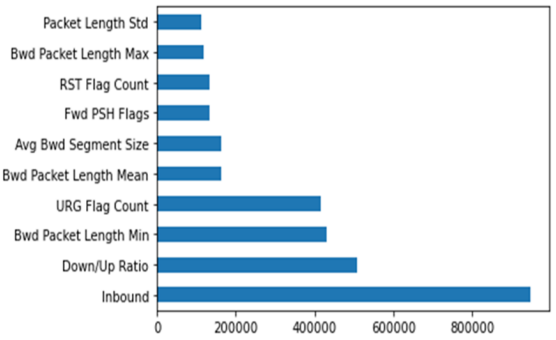

Fig [7]. SelectKbest Features

here we can find from fig[7]. the top 10 features extracted by using Selectkbest technique.

b) Feature Importance:

Feature importance is a type of feature selection. Feature importance is useful for reducing time of machine learning techniques and choice features have more effect on machine learning techniques. We can apply feature importance by using more models such as Decision tree, Extra trees and random forest. Feature importance depends on calculating the impurity weight of every feature of input [9] in any dataset.we apply feature importance by two model:

1. Extra trees:

We used Extra tree which is a type of ensemble methods algorithm. Extra tree works on aggregate the result of multiple tree models constructed randomly from the training dataset. Then calculating more mathematical criteria such as Gini importance to select features importance then sorts out the features then have been voted for. It fits every decision tree on the all dataset better than a bootstrap replica and chooses a split point at random to split the nodes [10] so we choose Extra trees. After applying Extra trees on our dataset

give us 65 feature importance from 78 features but apply our model which is CatBoost on top 10 features importance which is more efficient than using the whole 65 features and we obtain accuracy of our model 0.99998 .

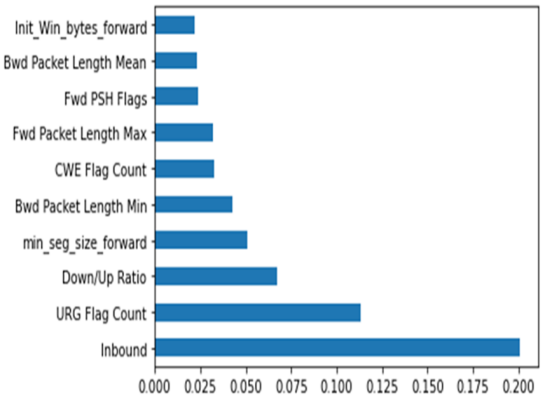

Figure 8. Top10 features

here we can find from figure 8. the top 10 features extracted by using Extra trees

2. Decision tree:
   We apply a decision tree which is a type of supervised learning of machine learning techniques and use one model in opposite to the extra tree to use it in feature importance. Decision tree in importance features based on impurity weighted of each input features or compute the importance of input features by using gini importance [11]. After applying a decision tree on our dataset, give us 26 important features from 78 features. this figure shows the all feature importance of our dataset with decision tree but apply our model which is CatBoost on top 10 features importance which more efficient than using the whole 26 features and we obtain accuracy of our model 0.99998.

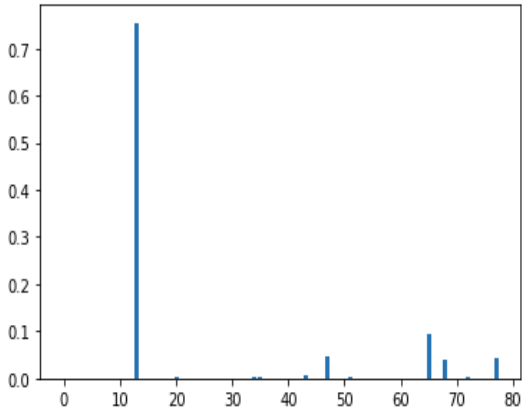

Fig [9]. all features importance

here we can find from fig[9]. the all features extracted by using decision tree

c) Recursive feature elimination with cross-validation:
The base Recursive feature elimination (RFE) model is a feature selection approach that fits a machine learning model and removes the weakest feature or set of the weakest features until the specified number of features is reached. In this approach the features are ranked by the **model's (coef_) or** (feature_importances_) attributes, and by recursively eliminating a few features per loop, RFE "attempts to eliminate dependencies and collinearity that may exist in the model".

Furthermore, RFE requires specifying a delimited number of features to keep, however it is often not known in advance as our main task is to determine how many features are valid.

To solve this problem and finding the optimal number of features, cross-validation is used with RFE to score the different feature subsets and selecting the best scoring collection of features automatically.

The RFECV visualizer plots the number of features in the model along with their cross-validated test score and variability and visualizes the selected number of features as shown in Figure 9. Moreover, the Optimal number of features after applying the RFE model with cross validation is 10 features.

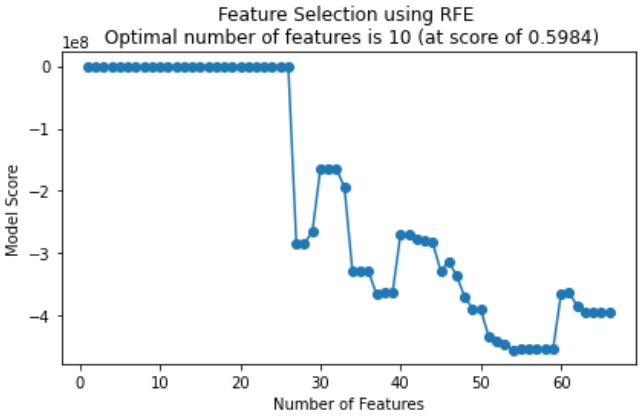

Figure 10: The Optimal number of features with RFECV

Furthermore, Those features had achieved a higher performance with our adopted model, as the model misclassified only 49 attacks as shown in the confusion matrix Figure 18.

Moreover, the classification report shows us the performance metric of our model in detail. But the prediction time was considered one of the best which was achieved with other approaches as it achieved 0.0002405 second in predicting a single attack with the Catboost trained model.

We use the feature ranking attribute to get the rank of the features and visualize the ranking of the top features as shown in Fig.4, The selected (best estimated) features are assigned rank 1.

So according to the evaluation we can find that we have 10 features with ranking 1 which the model considered as the optimal features to best in our scenario.

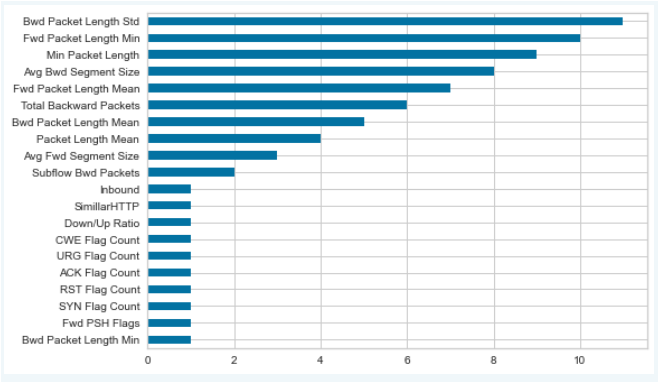

Figure 11: Feature ranking with RFECV

From the feature ranking of the Recursive Feature Elimination with cross validation implantation, we found that there are common features in most of the used approaches like Inbound attribute and others. Which is considered a good indicator for the importance of those features to identify the attacks.

d) SHAPLEY Additive Values (SHAP):

Understanding how the feature contributes to the model prediction, and interpreting the steps of the blackbox model. Is an important step to enhance the prediction stage. Reaching the point that we can understand the cause of the decision made by the model, is the point that we can interpret all the shadow work of the model. SHAP is an approach to explain the output of any machine learning model. “The goal of SHAP is to explain the prediction of an instance x by computing the contribution of each feature to the prediction” [2].

Implementation: We used SHAP Summaryplot to interpret the features and their impact over the model.

FIgure 12: SHAP Summary plot over DDoS attack CIC Dataset (Only part of the plot)

We can see from this figure that there are some features that its high values refer to the positive class and lower values refer to the negative class and vice versa. And others with a neutral impact.

Interpreting the plot and choosing the features that we believed that it will be useful we decided to use ('Fwd Packet Length Max', 'min_seg_size_forward', 'Inbound', 'URG Flag Count', 'Fwd PSH Flags', 'Flow IAT Mean').

Using those features and applying Catboost classifiers we were able to get accuracy of 0.9998 and F1_Score of 0.9999 with the minimal time per prediction of 0.0002423 seconds. And we can see that the model misclassified only two malicious network flows.

e) CatBoost Classifier:

CatBoost is based on gradient boosted decision trees. During training, a set of decision trees is built consecutively. Each successive tree is built with reduced loss compared to the previous trees. We can manage the number of trees by the starting parameters.

To prevent overfitting, use the overfitting detector. When it is triggered, trees stop being built and this is why we used this ensemble technique especially because of our imbalance dataset aiming for classification problems of DDos attack malicious or benign.

f) Deployment:

The deployment architecture is based on the ELK stack [8] to initialize the connection between all components of the project. ELK stack is composed of three main architectures, logstash, Elasticsearch and kibana.

Elastic search engine is a distributed search and analytics engine. Logstash data collection and indexing tool, collect and index data to elasticsearch. Kibana is the last step of ELK stack, responsible for data analysis and visualization through interactive dashboards, diagrams and graphs. Mainly the architecture is as follows: logstash as an indexer or data aggregator to index the data to the elasticsearch as a storage stage that can be analyzed and visualized in kibana.

By using these concepts, the model is reading the data to the kafka topic “raw_ddos_csv”, which will be cleaned and prepared for the prediction stage. Calling the pickled model “Catboost classifier” and passing the cleaned data over it obtaining both prediction and its confidence score. If the obtained label refers to the presence of an attack, the whole data related to that attack will be staged and sent to a new kafka topic “ddos_predictions”.

At this stage the Logstash is already configured as in figure 12, up, running and listening to the kafka prediction topic, once the data is reached, it will be indexed and sent to the elasticsearch and then kibana to be visualized.

```
input {
    kafka {
        codec => json
        bootstrap_servers => "localhost:9092"
        topics => ["ddos_predictions"]
    }
}

output {
    elasticsearch {
        hosts => ["localhost:9200"]
        index => "ddos-attacks-predictions"
        workers => 1
    }
}
```

Figure 13: Logstash Configuration file

And finally the dashboard created on kibana, it can be customized according to what you want to see.

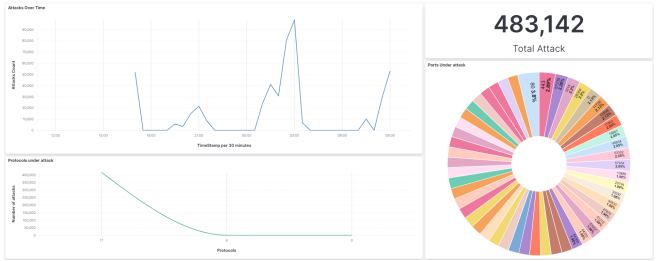

Figure 14: Kibana dashboard

In our dashboard, We visualized some important components which would help the users to identify the different characteristics of the attacks in real-time. Those components are the number of attacks through time, targeted ports, the targeted protocols in attacks, and the total number of attacks. As shown in the dashboard figure.14 the number of attacks through time would help to detect if we are under a heavy number of attacks. Furthermore.

The distribution of the targeted ports in attacks to help us to know which ports are mostly targeted by the attackers, Also, the attacked protocols, as we see in the dashboard there are three main protocols in the attacks. (0) which is Hop-by-Hop Option, (6) which is TCP Transmission Control and (17) UDP from the figure we can notice that the UDP protocol is the most targeted protocol in our case.

VII. PERFORMANCE EVALUATION

We evaluate every model by calculate the accuracy of model, F1-score, confusion matrix and Time which accuracy and f1-score are:
confusion matrix: is summary of results of classification which is the number of correct predictions and the number of misclassified predictions.
Accuracy $= TP + TN/\ TP + FP + TN + FN$
F**1-score = 2** $*$ ***Pression*** $*$ ***Recall/ Pression +Recall***
Time consumed per a single prediction.

Accuracy	Accuracy = (TP +TN) / (TN+FP+FN+TP)	Shows how many of the predictions are correct ([25], [16]).
F-measure	F-measure= 2.(Precision*Recall) / (Precision + Recall)	Computes the score between the precision and recall accuracies of the model for a given threshold [16].

VIII. RESULTS

We use these metrics specifically to be fair enough when it comes to comparing results without any bias to any class. It will give actual accuracy in comparison to each criteria not like generic accuracy which may be misleading.

Techniques	Accuracy	F1-score	Time
Univariant	0.99889	0.99944	0.00178
Importance features by Extra Trees	0.99998	0,99999	0.00094
Importance features by decision trees	0.99985	0.99993	0.00040
Recursive feature elimination with cross-validation	0.99899	0.99949	0.000244
SHAP for Feature Selection	0.99980	0.99990	0.000242

The comparison shows that the SHAP model is implemented for feature selection. Gives the best prediction in the best time. and time is an important factor in solving this problem.

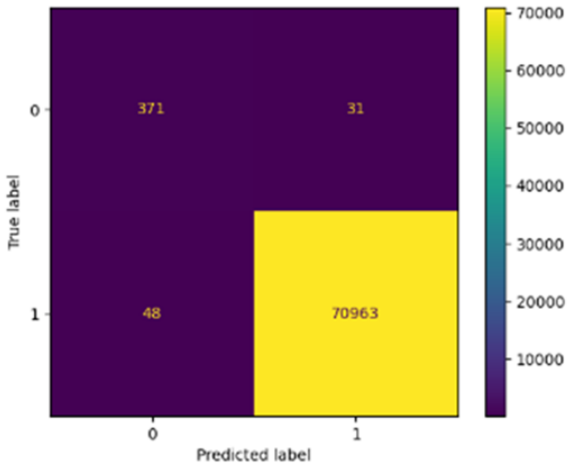

Figure 15: CatBoost with selectkbest confusion matrix

In figure 15, we can see from confusion matrix of selectkbest that it misclassified 49 samples for class1 and 23 samples for class 0 and may be that because we used only 10 features instead of using features from 20 to 30 as it was applied in Decision tree and Recursive feature elimination and this is why we will need more tuning hyper parameters.

And just to mention why we use Selectkbest ? Selectkbest retains first k features of input with high scores by applying simple statistics techniques unlike other FS which depend on an iterative approach, so this statistics approach will reduce time and give more efficient results. This is why we use this approach.

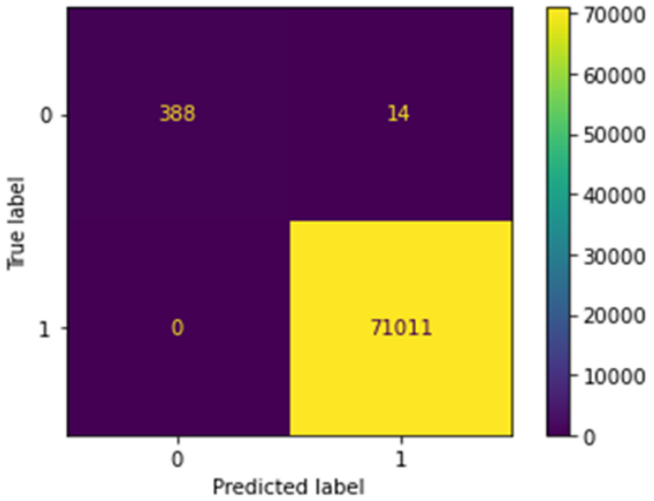

FIgure 16: CatBoost with Extra trees confusion matrix

In figure 16, We can see from the confusion matrix of Extra trees that it misclassified 14 samples for class0 so it is not very important because class 0 is BENIGN.

In figure 17, We can see from the confusion matrix of Decision Tree that it misclassified 7 samples for class0 and misclassified 2 samples for class1 so we tried to use Backward Feature Elimination because it depend on deleting small F-statistic and make our model more complex by using tuning hyper parameters to be able to know more details.

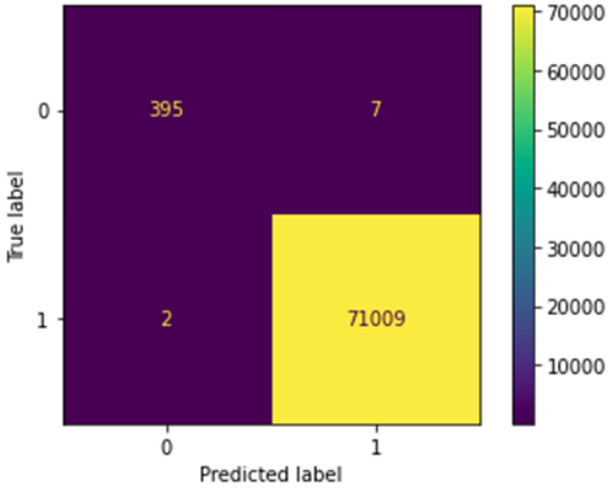

FIgure 17: CatBoost with Decision tree confusion matrix

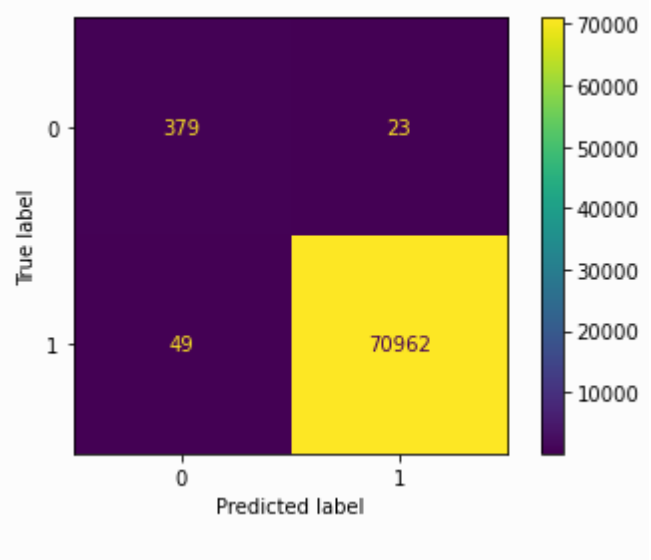

Figure 18: Catboost with RFECV Confusion Matrix

From the figure.18 which contains the confusion matrix of Catboost with Recursive feature elimination with cross-validation. We can notice that it misclassified 23 samples for class 0 and misclassified 49 samples for class 1. so we tried to tune the hyperparameters of RFECV , but it didn't make any improvements.

IX. CONCLUSION

In this experiment, we used the DNS attacks data sheet from the CICDDoS2019 dataset. Which contains different types of attack data generated from a simulation server. The dataset was used as a basis for identifying and verifying the impact of applying different approaches of feature selection on the performance of the proposed DDoS attack detection model. After applying those different feature selection approaches and tested their performance with the CatBoost trained model. We reach a conclusion that the SHAPLEY

Additive Values (SHAP). Interpreting models is an important concept to gain a good intuition of how the features affect the prediction process. SHAP is an applicable technique that can give a brief summary of how each of the values can affect the prediction and assign values to each feature to what extent it affects the prediction. The performance metric is measured with many aspects. As we compared the archived results of the different approaches with three main indicators. Those indicators are Accuracy, F1 Score and the tome of predicting a single attack. Furthermore, we used ELK stack which is composed of three main architectures, logstash, Elasticsearch and kibana to build a dynamic dashboard. And we visualized some influential components which would help the users to identify the different characteristics of the attacks in real-time. To be able to take some quick actions to prevent those attacks based on the extracted insights.

X. FUTURE WORK

After applying all current state, we want to enhance our application by merging other DDOS attacks such as DDoS LDAP, DDoS MYSQL, DDoS NetBIOS, DDoS NTP, DDoS SNMP, DDoS SSDP and DDoS UDP. applying more machine learning techniques such as SVM, Random forest, XGBoost if distribution of label in our dataset is convergent or not convergent because the ensemble methods able to deal with not convergent in distribution dataset and SVM if distribution of label in our dataset is convergent. applying more different techniques for FS as forward feature selection and Backward Feature Elimination [12] to have more generic comparison. applying more complex models to detect many numbers of attacks and Tune hyperparameters to see how that will affect other FS and final results.

References

[1] DDoS 2019 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. Unb.ca. (2022). Retrieved 12 April 2022, from https://www.unb.ca/cic/datasets/ddos-2019.html.

[2] Molnar, C. (2022). Interpretable Machine Learning (2nd ed.).

[3] Bhattacharyya, D. (2016). DDoS Attacks. CRC Press. https://doi.org/10.1201/b20614

[4] Lundberg, S. (2018). SHapley Additive exPlanations. Retrieved at: 23 Feb 2022. https://shap.readthedocs.io/en/stable/

[5] Bouzoubaa, K., Taher, Y., & Nsiri, B. (2022). DOS-DDOS Attacks Predicting : Performance Comparison of The Main Feature Selection Strategies. International Journal Of Engineering Trends And Technology, 70(1), 299-312. https://doi.org/10.14445/22315381/ijett-v70i1p235

[6] SaiSindhuTheja, R., & Shyam, G. (2021). An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment. Applied Soft Computing. https://doi.org/10.1016/j.asoc.2020.106997

[7] Polat, H., Polat, O., & Cetin, A. (2020). Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models. Sustainability, 12(3), 1035. https://doi.org/10.3390/su12031035

[8] The ELK Stack: From the Creators of Elasticsearch. Elastic.co. (2022). Retrieved 12 April 2022, from https://www.elastic.co/what-is/elk-stack

[9] Lee, C. (2017). Feature Importance Measures for Tree Models — Part I. Medium. Retrieved 13 April 2022, from https://medium.com/the-artificial-impostor/feature-importance-measures-for-tree-models-part-i-47f187c1a2c3.

[10] Gupta, A. (2020). ML | Extra Tree Classifier for Feature Selection - GeeksforGeeks. GeeksforGeeks. Retrieved 13 April 2022, from https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/.

[11] Brownlee, J. (2020). How to Calculate Feature Importance With Python. Machine Learning Mastery. Retrieved 13 April 2022, from https://machinelearningmastery.com/calculate-feature-importance-with-python/.

[12] Gupta, A. (2020). Feature Selection Techniques in Machine Learning. Analytics Vidhya. Retrieved 13 April 2022, from https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/.

[13] Araujo, P., Silva, A., Junior, N., Cabrini, F., Santos, A., Guelfi, A., & Kofuji, S. (2021). Impact of Feature Selection Methods on the Classification of DDoS Attacks using XGBoost. Journal Of Communication And Information Systems, 36(1), 200-214. https://doi.org/10.14209/jcis.2021.22