Scope
Volume 13 Number 3 September 2023

A Robust Feature Selection Framework for Effective Processing of Machine Learning Datasets

Samera Uga Otor

Beatrice Obianiberi Akumba, Isaac Terngu Adom and Joseph Sunday Idikwu

Department of Mathematics and Computer Science, Benue State University, Makurdi, Nigeria

Abstract

Most machine learning datasets are riddled with noise, outliers, redundant features, and blank entries. These datasets must be properly formatted for the learning models to process them and produce agood result using data preprocessing techniques such as data cleansing, feature selection, and feature engineering. Therefore, a feature selection framework was developed in this study. The framework defined a list of datasets, feature selection score functions for regressors and classifiers such as Chi-square, ANOVA, Pearson's correlation, and regressors and classifiers such as Decision Tree, Multilayer Perception Neural Network, K-nearest neighbor, and Random Forest as a pipeline. The framework was designed to choose between a regression predictive modeling anda classification predictive modeling based on the data type of the output variable. It also allows for the number of datasets, feature selection scores, regressors and classifiers to be increased or reduced as desired.

The framework was tested using the datasets CIC-DDoS2019, XIIoTID, DDoS-SDN, and DoS/DDoS-MQTT-IoT. The datasets were subjected to; several preprocessing techniques for data cleansing, which included filling the not-a-number values, infinity values, special characters, empty values, and converting negative values to positive values as needed and several feature engineering procedures, such as label imputers, encoders, and scalars. The datasets were then evaluated to get the features with the best scores for each dataset as either a classification or regression problem. Furthermore, to test for feature stability, the datasets were evaluated using recursive feature elimination (RFE). Results show that for the CIC-DDoS2019 and XIIoTID datasets, f-classif selected the best features with an accuracy of 99% to 100%. For DDoS-SDN datasets, f-regression with Random Forest regressor selected the best features with MSE of 0.0005 and R2 of 0.998%, and for DoS/DDoS-MQTT-IoT datasets, mutual info regressor with Random Forest selected the best features with MSE of 0.0128 and R2 of 94% respectively. For feature stability, the consistent features are supplied for researchers who intend to use the dataset for further research.

Keywords: Feature selection, machine learning, feature engineering, dataset, framework

1. Introduction

Raw datasets obtained by data capture tools from research test beds do not merely include features relevant to the problem at hand. The datasets contain some noise, outliers, irrelevant characteristics, and redundant characteristics, making them too huge for predictive models to work with. Working with a large dataset may require extensive processing and the consumption of resources such as time, memory space, and speed. While some features may be relevant, not all may be employed to obtain the desired results from a predictive model or algorithm. As a result, features that reduce computational complexity and resource consumption while maintaining high model accuracy should be chosen.

Feature selection is the process of reducing the number of attributes used to validate a model while maintaining its accuracy. It helps reduce computational complexity, resource consumption, and over-fitting, as well as improving the performance of the model. There are two main types of feature selection techniques: supervised and unsupervised techniques. Unsupervised feature selection methods ignore the target variable and use correlation to eliminate redundant variables, whereas supervised feature selection methods use the target variable to eliminate unimportant characteristics and choose the most crucial ones.

The supervised method is divided into filter, wrapper, and intrinsic, also known as embedded. Through the use of filters, characteristics are chosen based on statistical indicators like correlation or mutual information. Wrapper methods, on the other hand, use the model's performance as a criterion for feature selection. This approach can be computationally expensive but often leads to better results. Finally, intrinsic methods incorporate feature selection into the model itself, allowing for simultaneous optimization of both feature selection and model parameters. Regardless of the method chosen, it is important to carefully evaluate the selected features and ensure that they are relevant to the problem at hand.

As a result of the development of new computing paradigms like the internet of things (IoT), the industrial internet of things (IIoT), and the internet of everything (IoE), everything, including people, is now connected, both physically and virtually. The growing interconnection of physical and virtual components (technology), humans, and processes has resulted in the generation of massive amounts of data as well as impulsive vulnerabilities, risks, and threats to cyberspace (Behal et al., 2017).

The quality of the data plays a crucial role in the success of the learning model. Therefore, it is essential to carefully curate and preprocess them to remove redundant and missing data records, balance class distribution, and ensure high-quality annotations. This can be achieved through data cleaning, feature engineering, and feature selection, among others. By addressing these challenges, we can build more robust detection systems that are accurate, efficient, and scalable for real-world applications.

Many researchers have generated several datasets, such as the KDDCUP99, NSLKDD, TON-IoT (UNSW_IoT20) dataset, DoS/DDoS-MQTT-IoT, DDoS Dataset, XIIoTID, DDoS-SDN, and CICIDS-2019, from real-life test beds. They have also evaluated these datasets using machine learning techniques as predictive models. However, these datasets have some limitations. For example, they consist of a large number of data instances in different files, making it difficult to process them, and merging these files to include each attack label for robust detection makes the dataset larger and requires more computing and processing time. Additionally, the datasets contain some missing and redundant records and are prone to high-class imbalance, which could lead to low accuracy and a high false positive rate (FPR) for the developed system. (Ankit & Lohiya, 2020; Mahbod et al., 2009; Panigrahi & Borah, 2018).

According to Brownlee (2020), the data type of the output variable determines the type of predictive modeling problem being solved. For instance, a regression predictive modeling problem is decided by a numerical output variable, and a classification predictive modeling problem is decided by a categorical output variable. More so, knowing the data type of an input or output variable makes selecting a suitable statistical measure for a filter-based feature selection method easier. As a result, the selection of a proper feature selection and predictive algorithm is dependent on the input and output variables, as shown in Figure 1. It is important to note that there is no one-size-fits-all solution, as different statistical measures may yield different results. Therefore, this paper develops a feature selection framework that consists of a range of feature selection methods, classifiers, and regressors. This framework was further used to analyze different subsets of features chosen via different statistical measures by fitting them on the various classifiers and regressors, depending on whether it was a classification problem or a regression problem, to discover what worked best for a particular dataset. By doing so, we present a better understanding of the underlying patterns and characteristics of the dataset, which can enhance the performance of the learning model. The remaining part of this paper is organized as follows: Section two discusses the literature relevant to this study; section three is the method employed; section four discusses the results obtained; and section five is the conclusion and future work.

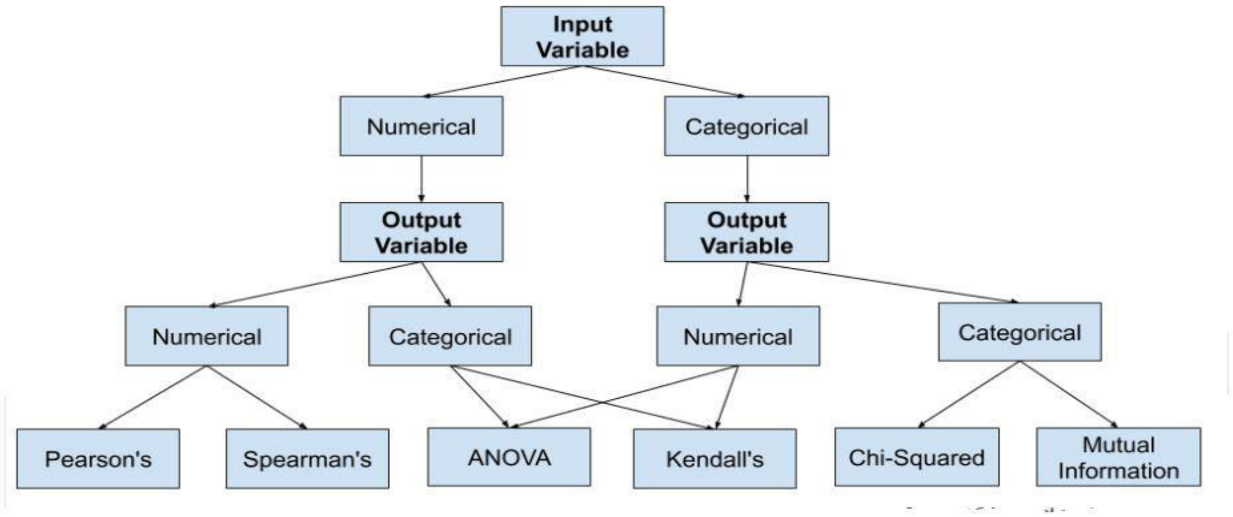

Figure 1: Input/Output Variables and the Corresponding Statistical Measures Used for Feature Selection (Source: (Brownlee, 2020))

2. Literature Review

2.1 Review of Some Machine Learning Datasets

Different researchers have proposed several datasets. Among these datasets, this work concentrates on the CIC-DDoS2019, XIIoTID, DDoS-SDN and DoS/DDoS-MQTT-IoT datasets. However, the framework was designed to accommodate as many datasets as possible.

The CICDDoS2019 was proposed by Sharafaldin et al. (2019) with the goal of correcting all flaws in earlier datasets. A new detection and classification strategy based on a set of network flow features was created using the dataset. Finally, the most essential feature sets for detecting various forms of DDoS attacks, as well as their corresponding weights, were identified.

To provide a robust dataset for DDoS evaluation, the DDoS Dataset was proposed by Devendra et al. (2019). The DDoS dataset consists of extracted DDoS flows from other public Intrusion Detection System (IDS) datasets such as CSE-CIC-IDS2018-AWS, CICIDS2017 and CIC DoS dataset2016. To introduce more variance into the dataset, DDoS data were extracted from different IDS datasets that were produced in different years and from different experimental DDoS traffic generation tools. The extracted DDOS flows were combined with Benign flows, which were extracted separately from the same base dataset and made into a single larger dataset known as the DDoS dataset.

To give researchers access to recent technologies such as software-defined networks, The DDoS SDN dataset was further proposed by Ahuja et al. (2020). This Mininet emulator-created data set for SDNs is used to classify traffic utilizing machine learning and deep learning techniques.. The project begins by configuring 10 Mininet topologies with switches connected to a single Ryu controller. Network simulation was performed for benign TCP, UDP, and ICMP traffic and malicious traffic, including TCP sync attacks, UDP Flood attacks, and ICMP assaults. The data collection has 23 features, some retrieved from switches and others calculated.

Moreso, the TON-IoT (UNSW_IoT20) dataset proposed by Moustafa et al. (2020) was introduced. This dataset was generated with the aim of meeting the demand for a large number of heterogeneous data sources used to train and validate new artificial intelligence (AI)-based security systems (Moustafa et al., 2020). It consists of merged data sources collected from datasets of IoT services, Operating systems such as Windows and Linux, and network traffic. The testbed used for the collection of the dataset employed three layers: edge, fog and cloud. The edge layer is made up of IoT and network devices; the fog layer contains virtual machines and gateways, and the cloud layer consists of cloud services, such as data analytics, linked to the other two layers. To generate the dataset, these layers were managed using the platforms of software Defined Network (SDN) and Network-Function Virtualization (NFV) using the VMware NSX and vCloud NFV platforms. The audit traces of memory,networks, processors, processes, and hard disks were used to compile the Windows datasets. The dataset was processed by selecting features using a correlation coefficient function to select the most correlated features with a cut-off value higher than or equal to 0.85% and several machine learning algorithms were applied to evaluate the dataset.

In the same vein, Al-Hawawreh et al. (2021) proposed the XIIoTID dataset. This dataset consists of new IIoT connectivity protocol behaviors, recent device actions, multiple attack kinds and scenarios, and various attack protocols. It defines attack nomenclature and includes functionality for several views, such as network traffic, host resources, logs, and warnings. Popular machine and deep learning methods were used to analyze the X-IIoTID. However, no feature selection method was applied.

Another is DoS/DDoS-MQTT-IoT. This dataset was proposed by Alaa et al.(2023) to generate a dataset based on the machine-to-machine IoT communications Message Queueing Telemetry Protocol (MQTT). A physical IoT testbed was built, and a significant amount of IoT data—including typical MQTT traffic and 10 DoS scenarios—was generated. This was further analysed using machine learning models. The researchers did not employ any feature selection method; only a small sample size was used to evaluate the machine learning models, with XG Boost and random forest algorithms performing the best.

Researchers have supplied these datasets and others for the research community to use for further research. However, they still need pre-processing to scale or perform well with the problems being solved. As a result, feature selection approaches such as Information gain, gain ratio, chi-squared, ReliefF, and symmetrical uncertainty have been in use to extract relevant features from the network traffic data for better accuracy (Suman et al.,2020; Yin et al., 2021). Others employed swarm optimization feature selection strategies to improve the performance of the generated model, such as a binary-particle swarm optimization approach (Aween& Noor, 2021) and a spider optimization algorithm (Otor et al., 2021).

A comprehensive survey of some feature selection algorithms can be found in Pradip and Chandrashekhar (2021); Rui et al. (2018); and Jie et al. (2018). They discussed the various feature selection methods applicable to machine learning problems and the core idea of how Feature selection can be applicable in various problem domains. This work leveraged the significance of feature selection to develop a framework that can be applied to several datasets to select optimal features for machine learning models.

2.2 Review of Some Feature Score Functions

The score functions used in this research are the chi-square (chi2), ANOVA (f_classif) and mutual information (mutual_info_classif, mutual_info_regression) and Pearson's correlation (f_regression), r_regression.

The Chi-Square (χ^2) test is a fundamental statistical method widely used to assess the association between categorical variables. It determines whether the observed distribution of categorical data significantly deviates from the expected distribution. The Chi-Square test finds application in various domains, including feature selection, which is crucial for enhancing the performance and interpretability of machine learning models. The Chi-Square test quantifies the dissimilarity between observed and expected frequencies within a contingency table. The formula for calculating the Chi-Square statistic is presented in Equation 1.

$$\chi^2 = \sum((O - E)^2/E) \qquad 1$$

Where:
χ^2: Chi − Square
O: Observed Frequency
E: Expected Frequency
The Chi-Square test is particularly helpful when dealing with categorical features. It enables the evaluation of the statistical significance of relationships between categorical features and a categorical target variable (Han et al., 2011).
Analysis of Variance (ANOVA) is a widely used statistical method for assessing the differences in means across multiple groups or categories. In the context of feature selection, ANOVA can help identify significant features that exhibit variations between distinct categories. In machine learning, ANOVA can be applied to perform feature selection by using the f_classif method. This method calculates the F-statistic and p-values for each feature with respect to the target variable, allowing the identification of features that show significant variations across different target classes (Pedregosa et al., 2011). For one-way ANOVA, the F-statistic is computed as shown in Equation 2 (Hogg et al., 2018).

$$F = (Between\ group\ variance/(k-1))/(within\ group\ variance/(n-k)) \qquad 2$$

Where:
F: F-statistic
Between-group variance: Variance between group means
Within-group variance: Variance within individual groups
k: Number of groups
n: Total number of observations.

Mutual Information (MI) is a measure that quantifies the amount of information shared between two variables. In the context of feature selection, MI helps in understanding the strength of the relationship between a feature and the target variable. By applying Mutual Information (using mutual_info_classif or mutual_info_regression), the most informative features can be found, providing insights into the terms that strongly correlate with the target variable (Pedregosa et al., 2011). Higher MI values show stronger dependencies, making those features potentially valuable for predictive modeling.
The formula for calculating Mutual Information (MI) between feature X and target variable Y is given by Equation 3.

$$MI(X,Y) = \sum\sum \mathrm{P}(X,Y) * \log 2\left(\mathrm{P}\,(X,Y)/\left(\mathrm{P}(X) * \mathrm{P}(Y)\right)\right) \qquad 3$$

Where:
$MI(X,Y)$: Mutual Information between feature X and target variable Y
$P(x,y)$: Joint probability distribution of feature X and target variable Y
$P(x)$: Marginal probability distribution of feature X
$P(y)$: Marginal probability distribution of target variable Y

Pearson's correlation coefficient (r) measures the strength and direction of a linear relationship between two variables. It ranges from -1 to 1, where 1 shows a perfect positive linear relationship, -1 shows a perfect negative linear relationship and 0 shows no linear relationship. In regression tasks, the f_regression method is commonly used to calculate the F-statistic and p-values for each feature's correlation with the target variable. This method helps identify features that have a strong linear relationship with the target variable. The formula for calculating Pearson's correlation coefficient between feature X and target variable Y is given by Equation 4:

$$r = \Sigma\left((X - \bar{X}) * (Y - \bar{Y})\right) / \left(sqrt(\Sigma\,(X - \bar{X})^2) * sqrt(\Sigma\,(Y - \bar{Y})^2)\right) \qquad 4$$

Where:
r: Pearson's correlation coefficient between feature X and target variable Y
$\bar{X}$: Mean of feature X
$\bar{Y}$: Mean of target variable Y

3. Materials and Methods

A framework was designed to allow for the choice of several feature selection methods and predictive models based on the dataset and the problem being solved. The feature selection methods used are chi-square (chi2), ANOVA (f_classif) and mutual information (mutual_info_classif) for classification; Pearson’s correlation (f_regression), r_regression and mutual information (mutual_info_regression) for regression and recursive feature elimination. While the classifiers and regressors are KNeighbors classifier and regressor, support vector classifier and regressor, decision tree classifier and regressor, random forest classifier and regressor and multi-layerperception classifier and regressor. To make it robust, the output variables of the different datasets were changed from categorical to numerical and vice versa. The methods employed are described as follows:

3.1 Data Pre-processing

Special characters and blank cells were removed from the datasets using the data frame replace and fill not a number method in pandas. Numerical values were converted to positive values using the absolute method to enable regression feature selection algorithms to process the datasets. Furthermore, the label encoder, categorical variable encoder and standard scalar feature engineering methods were employed where necessary.

3.2 Model Building

The model was built using Python on the Google Colab platform as follows:

1. Define Score Functions: In this section, a list of score functions used for classification feature selection, such as `f_classif` and `mutual_info_classif and a list of score functions used for regression feature selection, such as `f_regression`, `mutual_info_regression`, and `r_regression` were defined. There is room to add more or remove functions when needed.

2. Define Feature Selection Methods: A list of feature selection methods for classification tasks that use the score functions defined in `classification_score_functions` to select the top 5 features and a list of feature selection methods for regression tasks that use the score functions defined in `regression_score_functions` to select the top 5 features were also defined.
3. Define Classifiers and Regressors: a list of classification models, including K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Decision Tree Classifier, Random Forest Classifier, and Multi-Layer Perceptron (MLP) Classifier and a list of regression algorithms/models, including K-Nearest Neighbors (KNN), Support Vector Regressor (SVR), Decision Tree Regressor, Random Forest Regressor, and Multi-Layer Perceptron (MLP) Regressor were defined. The classifiers and regressors can also be increased or reduced as desired.
4. Define Datasets: a list of datasets on which the models will be trained and evaluated was defined as pandas Data Frames. This can also be reduced or increased.
5. Pipeline: A pipeline was developed that iterates over the datasets using feature selection methods and classifiers or regressors defined above depending on whether it is a regression or classification problem.

The framework gives room for the list of score functions, feature selection methods, classifiers and regressors, and datasets to be increased as desired. The algorithm is as presented in Algorithm 1 and the framework is as shown in Figure 2.

Algorithm 1: Feature Selection Workflow

```
Input: List of datasets - datasets
 1. Define classification_score_functions as a list of classification feature selection score functions
 2. Define regression_score_functions as a list of regression feature selection score functions
 3. Define classification_feature_selection_methods as a list of SelectKBest feature selection methods with classification
    score functions
 4. Define regression_feature_selection_methods as a list of SelectKBest feature selection methods with regression score
    functions
 5. Define classifiers as a list of classification models (e.g., KNeighborsClassifier, DecisionTreeClassifier,
    RandomForestClassifier)
 6. Define regressors as a list of regression models (e.g., KNeighborsRegressor, DecisionTreeRegressor,
    RandomForestRegressor)
 7. Define classification_metrics as a list of classification evaluation metrics (e.g., accuracy, precision, recall,f1)
 8. Define regression_metrics as a list of regression evaluation metrics (e.g., r2)
 9. Initialize an empty dictionary classification_scores to store evaluation scores for classification models
10. Initialize an empty dictionary regression_scores to store evaluation scores for regression models
11. Iterate over each dataset in datasets with index ds_cnt
        A. Print "Dataset:", "Dataset", ds_cnt + 1
        B. Extract the feature matrix X and the target variable y from the dataset
        C. Determine the type of the task (classification or regression) based on the datatype of y
        D. IF the task is regression, set feature_selection_methods to regression_feature_selection_methods,
           models to regressors, evaluation_metric to r2_score, score_functions toregression_score_functions,andtask_type to
           "Regression"
```

```
ELSE, set feature_selection_methods to classification_feature_selection_methodsmodelsto classifiers,
evaluation_metric to accuracy_score, score_functions to classification_score_functions, and task_type
 to "Classification"
        E. Split the dataset into training and testing sets using train_test_split()
        F. Initialize an empty list selected_features_lists to store selected features for each feature selection
     method
        G. Iterate over each feature selection method and corresponding score function in
feature_selection_methods and score_functions
                    i.   Print "Feature Selection Method:", name of feature_selection_method, "Task Type:", task_type,
                "Score Function:", name of score_function
                    ii.  Apply feature selection to the training set X_train and the testing set X_test using
                            feature_selection_method
                iii. Store the selected features in selected_features_lists
                iv. Iterate over each model in models
                        a. Train the model using the training data and selected features
                        b. Make predictions on the test data using the trained model
                        c. IF the task is regression:
                                - Calculate the evaluation score using evaluation_metric and store it in
                        regression_scores with the corresponding metric (e.g., 'r2')
                               - Print "Regressor:", name of the model, "R2 Score:", the evaluation score
                  ELSE (i.e., for classification):
                               - Calculate the evaluation scores (accuracy, precision, recall, f1) using their
                  respective  metrics and store them in classification_scores
                               - Print "Classifier:", name of the model, "Accuracy Score:", accuracy, "Precision
                                 Score:",  precision, "Recall Score:", recall, "F1 Score:", f1
        H. Display the selected features by each feature selection method from selected_features_lists
        I. IF the task is regression:
                - Plot the regressors and their R2 scores using regression_scores
       ELSE (i.e., for classification):
         - Plot the classifiers and their metrics (accuracy, precision, recall, f1) using classification_scores
```

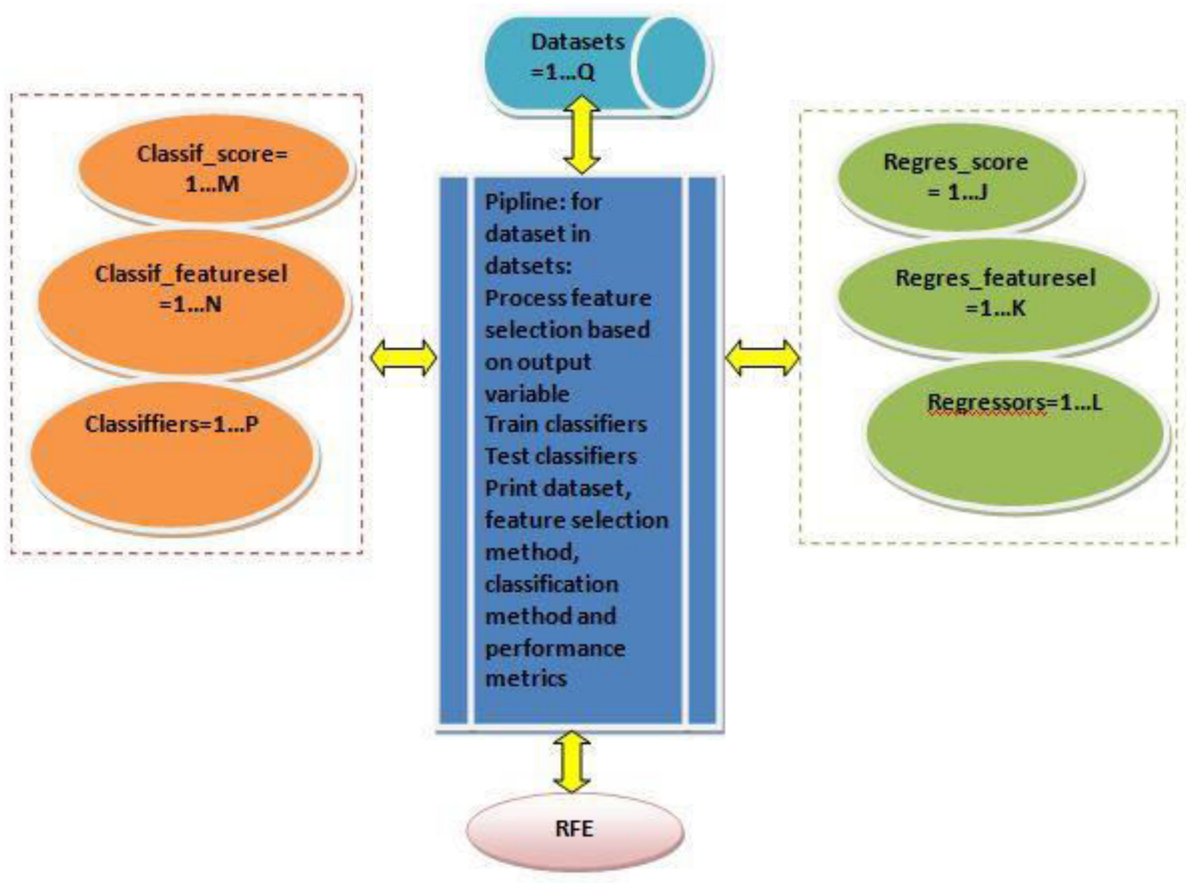
Datasets =1...Q
Classif_score= 1...M
Classif_featuresel =1...N
Classiffiers=1...P
Pipline: for dataset in datsets: Process feature selection based on output variable Train classifiers Test classifiers Print dataset, feature selection method, classification method and performance metrics
Regres_score = 1...J
Regres_featuresel =1...K
Regressors=1...L
RFE

Figure 2: The Model Architecture

4. Results and Discussions

4.1 Results

The framework was tested using CIC-DDoS2019, XIIoTID, DDoS-SDN and DoS/DDoS-MQTT-IoT datasets. The test was carried out as follows:

Scenario 1: The datasets were tested by allowing the output dataset to remain as presented, either categorical or numerical. The framework then analysed the dataset based on the output variable as either a classification problem or a regression problem. The results for these are as shown in Tables 1, 2, 3, 4 and Figures 2, 3, 4, 5

Scenario 2: The datasets were subjected to Recursive Feature elimination (RFE). This is to test for consistency in the selected optimal features. The results are presented in table 5

Table 1: Evaluation Scores for DDoS SDN dataset

Scores Regressors	F-Regression		Mutual-Info-Regression		R-Regression	
	R2	MSE	R2	MSE	R2	MSE
KNNeighbor s	0.012594	0.947075	0.003446	0.985518	0.003413	0.985659
DecissionTre e	0.000599	0.99748	0.000740	0.996890	0.000623	0.997382
RandomFore st	0.000508	0.99787	0.000652	0.997262	0.000579	0.997568
MLP	0.051955	0.78168	0.050327	0.788521	0.041216	0.826804

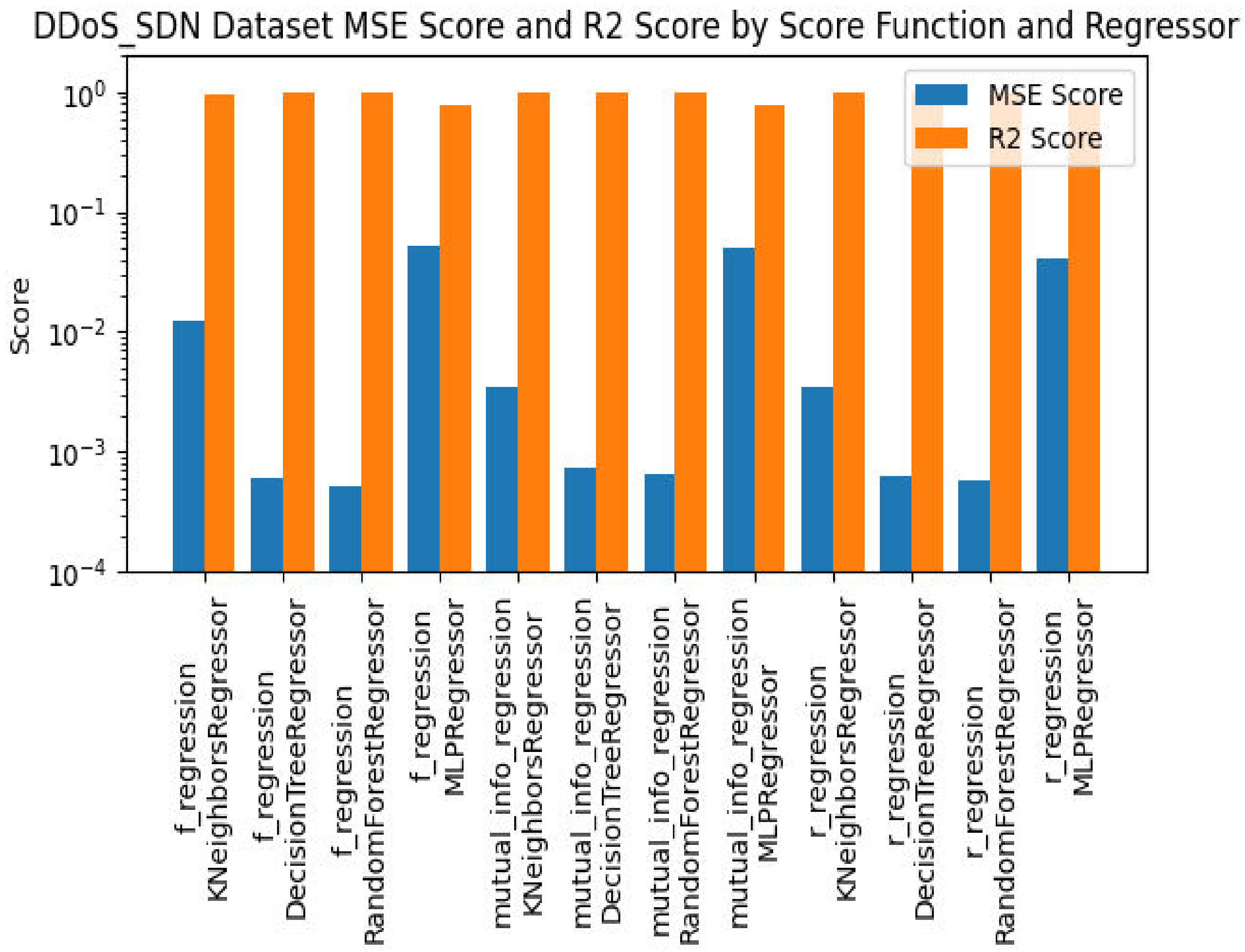

Figure 2: MSE and R2 Scores for DDoS SDN Dataset

Table 2: Evaluation Scores for X-IIoTID dataset

Scores Classifiers	F-Classif				Mutual-Info-Classif			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
KNeighbors	0.99995	1.00000	0.99991	0.99995	0.99994	0.99998	0.99991	0.99994
DecissionTree	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
RandomForest	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
MLP	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000

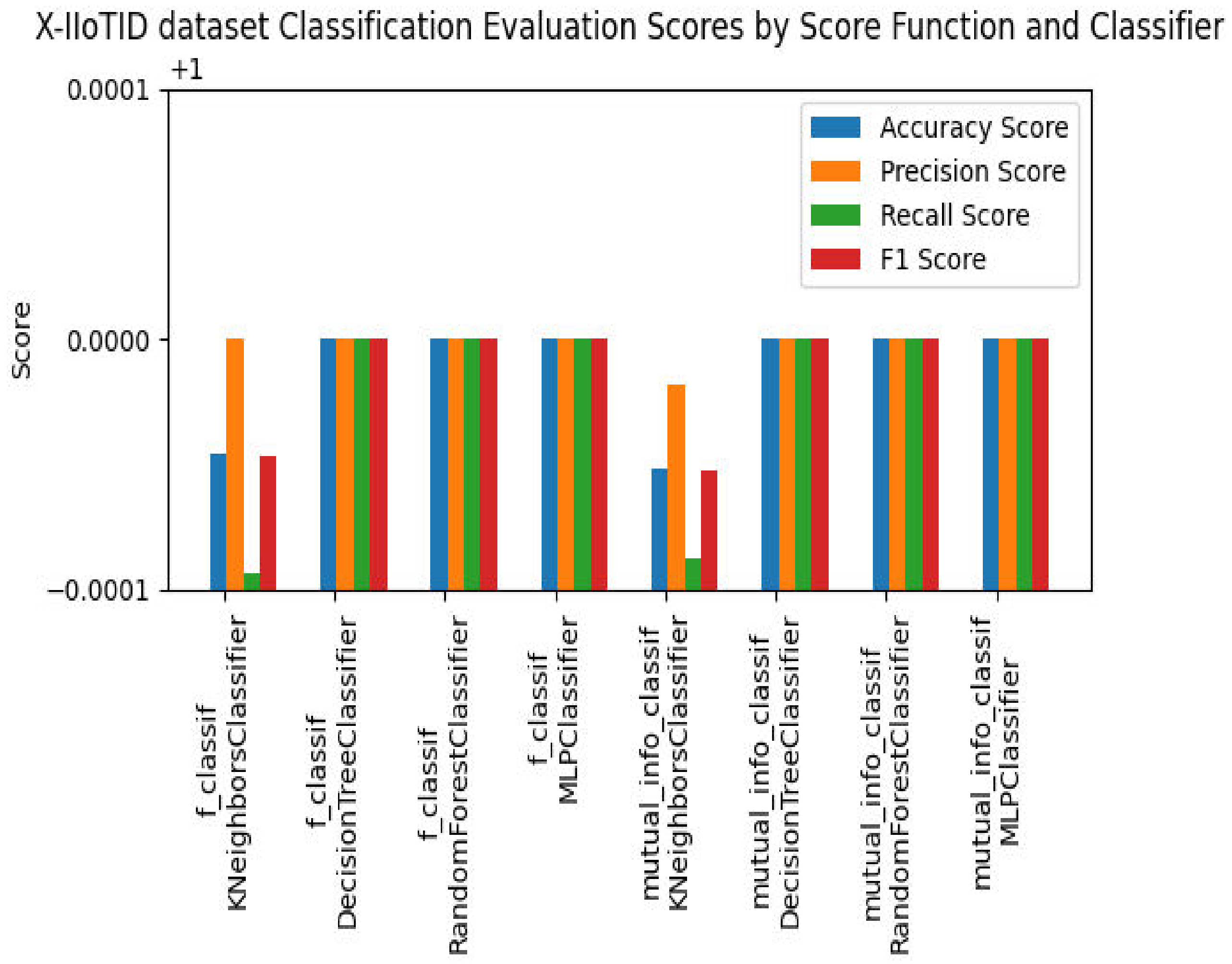

Figure 3: Accuracy, Precision, Recall and F1 Scores for X-IIoTID Dataset

Table 3: Evaluation Scores for CICCDOS2019 dataset

Scores Classifiers	F-Classif				Mutual-Info-Classif			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
KNeighbors	0.99999	0.99997	0.99992	0.99995	0.99934	0.99993	0.99939	0.99966
DecissionTree	0.99991	0.99993	0.99997	0.99995	0.99937	0.99985	0.99951	0.99968
RandomForest	0.99995	0.99997	0.99997	0.99997	0.99950	0.99989	0.99960	0.99975
MLP	0.99987	0.99989	0.99997	0.99993	0.99722	0.99975	0.99741	0.99857

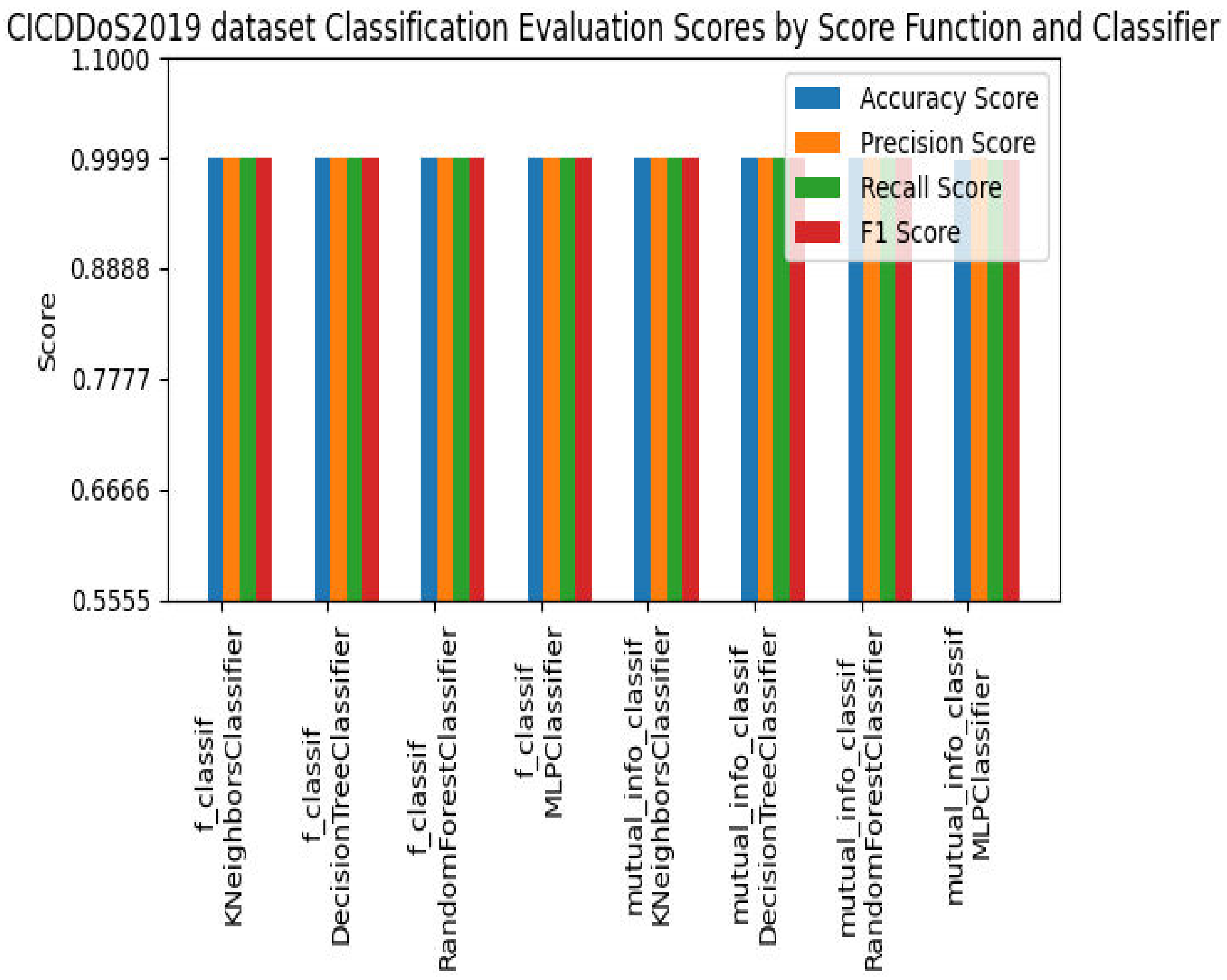

Figure 4: Accuracy, Precision, Recall and F1 Scores for CICCDOS2019 Dataset

Table 4: Evaluation Scores for DoS_DDoS_MQTT_IoT dataset

Scores Regressors	F-Regression		Mutual-Info-Regression		R-Regression	
	R2	MSE	R2	MSE	R2	MSE
KNeighbors	0.84843	0.03218	0.88888	0.02359	0.80260	0.04191
DecissionTree	0.85334	0.03114	0.91787	0.01744	0.84250	0.03344
RandomForest	0.86856	0.02791	0.93969	0.01280	0.87940	0.02560
MLP	0.76243	0.05044	0.83070	0.03594	0.71437	0.06064

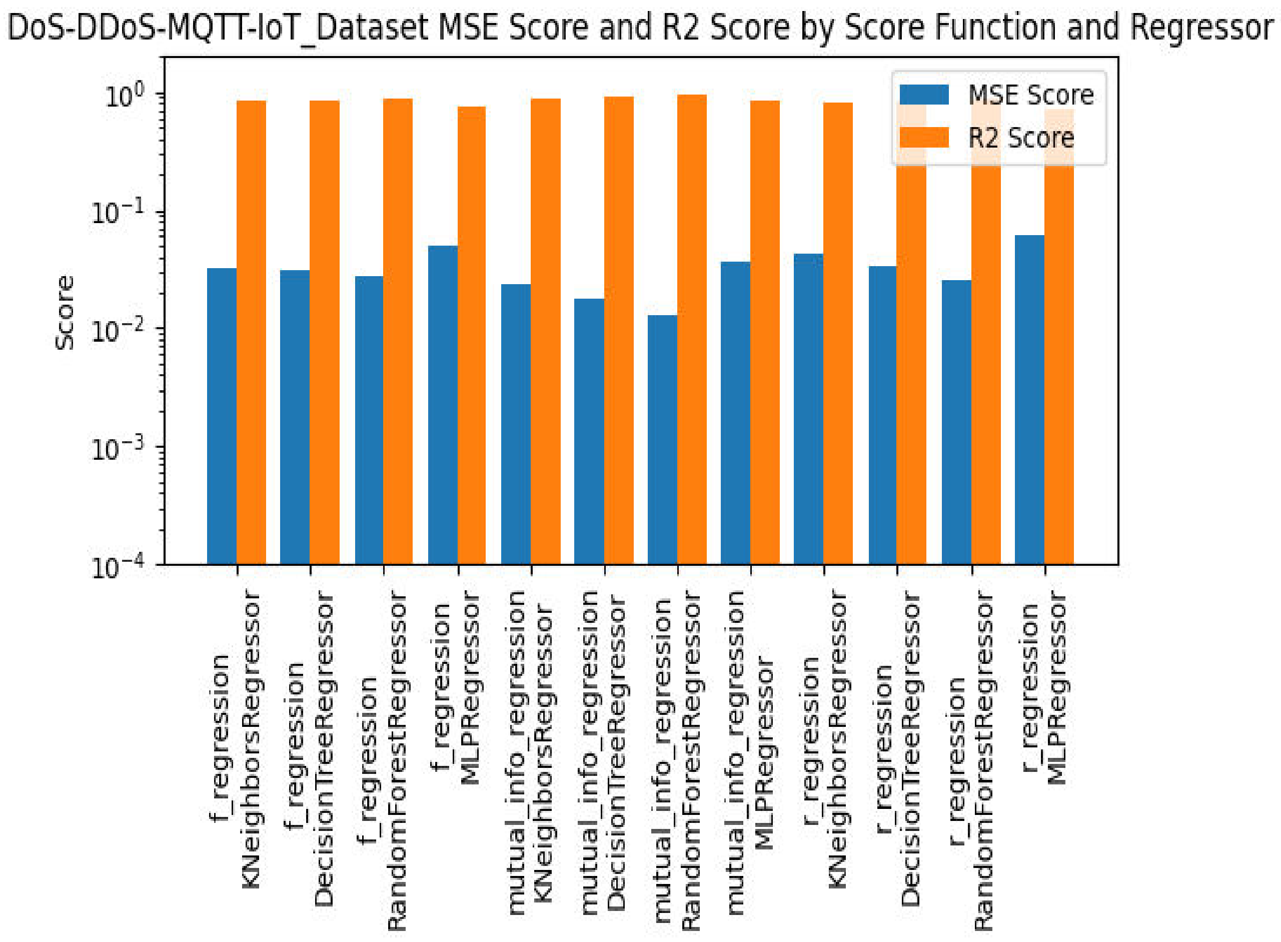

Figure 5: MSE and R2 Scores for DoS_DDoS_MQTT_IoTDataset

Table 5: RFE Results

SCORES / REGRESSION DATASET	Features selected by f_regression	Features selected by mutual_info_ regression	Features selected by mutual_info_ regression	RFE features
DDOS SDN	dt, pktcount, bytecount, flows, Protocol	pktcount, bytecount, pktperflow, byteperflow, pktrate	pktcount, bytecount, pktperflow, pktrate, Protocol	dt, switch, pktcount, bytecount, dur, dur_nsec, tot_dur, flows, packetins, pktperflow, byteperflow, pktrate, Pairflow, Protocol, port_no, tx_bytes, rx_bytes, tx_kbps, rx_kbps, tot_kbps
DOS_DDOS_MQTT IOT	Epoch Time, Protocol, Time delta from previous displayed frame, Syn, Retain	Frame length on the wire, Time delta from previous displayed frame, Time since reference or first frame, Frame length on the wire.1, Stream index	Epoch Time, Protocol, Time delta from previous displayed frame, Time since reference or first frame, Stream index	Message Type, QoS Level, Epoch Time, Protocol, Frame length on the wire, Time delta from previous displayed frame, Time since reference or first frame, Frame length on the wire.1, Stream index, iRTT, Time since first frame in this TCP stream, TCP Segment Len, Calculated window size, Syn, Reset, Acknowledgment, Keep Alive, User Name Length, Password Length, Clean Session Flag.1, Will Retain, Will Flag, Topic Length, Msg Len
SCORES / CLASSIFICATION DATASETS	Features selected by f_classif:	Features selected by mutual_info_ classif:		RFE
X-IIOTID	Des_port, is_with_payload , Std_system_tim e, class1, class2	Timestamp, Scr_ip_bytes , total_bytes, class1, class2		Selected features: Timestamp, Avg_nice_timeAvg_system_time, Avg_iowait_time, Avg_ideal_timeAvg_ldavg_1Avg_num_Proc/s, Avg_num_cswch/s class1,class2
CICCDOS2019	Source Port,Protocol,F wd Packet Length Min, Min Packet	Source Port, Total Length of FwdPackets, Min Packet		Fwd Packet Length Max, Fwd Packet Length Min, Bwd Packet Length Std, Flow Bytes/s, Flow Packets/s,

	Length, Inbound	Length, Average Packet Size,Subflow Fwd Bytes		Max Packet Length, Packet Length Mean, Packet Length Std, Avg Bwd Segment Size, Active Min

4.2 Discussion of Results

Scenario 1: For the CIC-DDoS2019 and XIIoTIDdatasets f-classif selected the best features with accuracy of 99% to 100%, for the DDoS-SDN dataset, f-regression with Random Forest regressor selected the best features with MSE of 0.0005 and R2 of 0.998%; and for the DoS/DDoS-MQTT-IoT dataset, mutual info regressor with random forest selected the best features with MSE of 0.0128 and R2 of 94% respectively. The f_classif feature selection method successfully identified the best features for the CIC-DDoS2019 and XIIoTID datasets. The accuracy achieved is exceptional, ranging from 99% to 100%. This implies that the selected features have strong discriminatory power and contribute significantly to the classification task, resulting in highly accurate predictions. The f_regression feature selection method, combined with the Random Forest Regressor, yielded impressive results for the DDoS-SDN dataset. The low MSE indicates that the selected features contribute to accurate regression predictions with minimal error. The high R2 score of 0.998 suggests that the selected features explain a sizable part of the variance in the target variable, indicating strong predictive power.The obtained MSE of 0.0128 for the DoS/DDoS-MQTT-IoT dataset signifies relatively accurate regression predictions. The R2 score of 94% indicates that the selected features capture a substantialportion of the target variable's variance, highlighting the effectiveness of the feature selection process.
Scenario 2: The RFE was used to test the stability of the features. Those features consistent with at least two of the feature selection methods were considered stable. Therefore, the consistent features for each of the datasets are presented in Table 6.

Table 6: Stable Features Selected

Dataset	Features
DDOS_SDN	dt, Protocol,pktcount, bytecount, pktperflow, byteperflow, pktrate
DOS_DDOS MQTT_IOT	Epoch Time, Protocol,syn,Frame length on the wire, Time delta from previous displayed frame, Time since reference or first frame, Frame length on the wire.1, Stream index
X-IIOTID	Timestamp, class1, class2
CICCDOS2019	Source Port, Fwd Packet Length Min, Min Packet Length,

5. Conclusion

This study developed a feature selection framework for machine learning datasets. The framework defined a list of datasets, score functions for regression and classification and regressors and classifiers as a single pipeline. The datasets, score functions, regressors and classifiers can be increased or reduced as desired. The framework analysed the datasets based on the data type of the output variable as either a regression or classification problem. It also analysed the datasets for feature stability using the RFE feature selection method. Results obtained show that for the CIC-DDoS2019 and XIIoTID datasets, f-classif selected the best features with accuracy of 99% to 100%, for the DDoS-SDN dataset, f-regression with Random Forest regressor selected the best features with MSE of 0.0005 and R2 of 0.998%; and for the DoS/DDoS-MQTT-IoT dataset, mutual info regressor with random forest selected the best features with MSE of 0.0128 and R2 of 94%, respectively.
In summary, the results suggest that different feature selection methods are suitable for different datasets and tasks. The high accuracy, low MSE, and high R2 scores obtained from these methods indicate that the selected features are valuable and contribute significantly to the performance of the predictive models. These findings reinforce the importance of appropriate feature selection in enhancing model accuracy, interpretability, and overall effectiveness for various types of datasets and machine learning tasks.
Therefore, the results can help guide researchers who want to use this dataset as to what features are relevant to save time, cost and model complexity.

References

1. Alaa, A., Leslie, F, S., Mike, J., Patryk, S., and James, J. K. (2023). DoS/DDoS-MQTT-IoT: A dataset for evaluating intrusions in IoT networks using the MQTT protocol. *Computer Networks 231* (2023) 109809, 1-8.
2. Al-Hawawreh, M., Sitnikova, E., andAboutorab, N. (2021). X-IIoTID: A Connectivity- and Device-Diagnostic Intrusion Dataset for Industrial Internet of Things. *IEEE Internet of Things Journal*, 2327-4662.
3. Ankit, T., andLohiya, R. (2020). A Review of the Advancement in Intrusion Detection Datasets. *Procedia Computer Science167* (2020), 636–645
4. Aween, A. S., and Noor, G. M. (2021). Intelligent feature selection using particle swarm optimization algorithm with a decision tree for DDoS attack detection. *International Journal of Advances in Intelligent Informatics 7*(1), 37-48.
5. Behal, S., Saluja, K. K., and Sachdeva, M. (2017). Discriminating flash events from DDoS attacks: A comprehensive review. *International Journal of Network Security 19*(5),734-741.
6. Brownlee, J. (2020, August 20). How to Choose a Feature Selection Method for Machine Learning.*Machine Learning Mastery.*
7. Devendra, P. M., Prasanta, B. V., and Amarnath, C. (2019). Machine Learning DDoS Detection Using Stochastic Gradient Boosting. *International Journal of Computer Sciences and Engineering 7*(4), 157-166

8. Han, J., Pei, J., and Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann
9. Hogg, R. V., McKean, J., & Craig, A. T. (2018). *Introduction to Mathematical Statistics*. Pearson.
10. Jie, C., Jiawei, L., Shulin, W., and Sheng, Y. (2018). Feature selection in machine learning: a new perspective. *Neurocomputing* (2018)
11. Mahbod, T., Ebrahim, B., Wei, L., & Ali, A. G. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. *In proceedings of the 2009 IEEE symposium on computational intelligence in security and defence applications*.
12. Moustafa, N., Keshk, M., Debie, E., andJanicke, H. (2020). Federated TON IoT Windows Datasets for Evaluating AI-based Security Applications. arXiv:2010.08522v1 [cs.CR]
13. Otor, S. U., Akinyemi, O. B., Aladesanmi, A. T., Aderounmu, A. G., andKamagaté, B. H. (2021). An improved bio-inspired based intrusion detection model for a cyberspace. *Cogent Engineering*.*8*(1), 1-23.
14. Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. *International Journal of Engineering & Technology 7(*3.24), 479–482.
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weis, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., andDuchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal *of Machine Learning Research. 12*
16. Pradip, D., & Chandrashekhar, A. (2021). A comprehensive survey on feature selection in the various fields of machine learning. *Applied Intelligence* 1-39
17. Rui, Z., Feiping, N., Xuelong, L., & Xian, W. (2018). Feature Selection with Multi-view Data: A Survey. *Information Fusion*
18. Sharafaldin, I., Habibi, A. L., Hakak, S., &Ghorbani, A. A. (2019). Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy *IEEE 53rd International Carnahan Conference on Security Technology*. Chennai, India.
19. Suman, N., Santanu, P., & Koushik, M. (2020). Detection of DDoS Attack and Classification Using a Hybrid Approach. *The Third ISEA Conference on Security and Privacy ISEA-ISAP* (pp. 41-47).
20. Yin, M. S., Pye, P. A., & Aye, S. H. (2021). A Slow DDoS Attack Detection Mechanism using Feature Weighing and Ranking. *In Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management* (pp. 4500-4509).