[1] **Ms. K. Pramilarani***

[2] **Dr. Vasanthi Kumari P**

Performance Enhancement in Machine Learning Approaches using Recursive Feature Extraction, Maximum Variance Method and Min Max Method

Journal of Electrical Systems

Abstract: - This paper presents the well known techniques used for selecting and extracting the features in the dataset to improve the performance of the machine learning approaches in identifying DoS/DDoS attack types. Intrusion Detection is an essential part of an organization's cybersecurity defense strategy. It complements other security measures like firewalls, antivirus software, and access controls, helping to identify and mitigate potential threats that may have bypassed other security layers. By proactively identifying and addressing security breaches promptly, companies can secure their confidential information, ensure uninterrupted operations, and defend their standing against digital dangers. Machine Learning approaches are used to identify the threat in network. The input to the machine learning model should be preprocessed to enhance the overall performance of the model. Recursive Feature Elimination is one such method used to reduce the total number of features based on the importance ranking. In maximum variance method, variance for all the data samples are collected to identify the threshold value using the mean of all the variance. Data normalization is used in min max algorithm to make all the data in the similar range. These algorithms are used to preprocess the data so that the classifier overall performance will be enhanced. The data set considered for this research work is KDD+ data set which is mainly used to train and test the model for identifying different types of network attacks. The data set contains 41 different features for network connection, status, protocols, services and associated label for the attack category. The performance of any machine learning approach will be increased by reducing the number of features from the data set. The research work shows that the accuracy is improved by 10% to 20% after using the preprocessing algorithms for feature selection and noise removal.

Keywords: Intrusion Detection(ID) , cyber attacks, feature extraction, RFE, DoS attack.

I. INTRODUCTION

Intrusion Detection serves as a digital security system that detects and reacts to unauthorized and harmful actions aiming to breach the defenses of computer systems, software, and networks. The primary goal of intrusion detection is to monitor, detect and react to potential security breaches to prevent and minimize the damage caused by cyber attacks. Intrusion detection is an important part of cybersecurity. There are two categories of Intrusion Detection System. First category is called as network based IDS, which operates on network connected systems. The other one is called host based IDS, which operates on host level. Based on the research it is clearly shown that the number of features involved in any machine learning approach will affect the overall system performance. To improve the overall system performances, proper feature selection and feature extraction methods are needed. Feature selection and feature extraction are important techniques used in intrusion detection for enhancing the effectiveness and efficiency of the detection process. In this research work features are selected based on ranking method, threshold values are set based on the maximum variance method and the data will be normalized by min max algorithms to select and extract features. Section 2 discuss about the related works. Section 3 provides the introduction about dataset and types of attack given in dataset for training and testing purpose. Section 4 provides the need to preprocess the data and various methods available to preprocess the data. Section 5 explain the algorithm and example to preprocess the data using RFE, maximum variance and min max methods under methodology section. Section 6 explains Performance evaluation done on the system with/without feature reduction and comparative analysis is given under Results and Discussion and finally Section 7 gives the conclusion.

II. RELATED WORKS

Shadi Aljawarneh et. al[1] discussed about the a novel hybrid model designed to determine the threshold level of intrusion scope, utilizing the best features of network transaction data that have been provided for training purposes.

[1] Research Scholar, DSU, Senior Assistant Professor, NHCE, Department of CSE, pramiselva2020@gmail.com

[2]Professor, Department of Computer Applications, School of Engineering, DSU, vasanthi-bca@dsu.edu.in

* Corresponding Author Email: pramiselva2020@gmail.com

This will reduce the computational and time complexity. The model gives 99.8% for binary class and 98.5% accuracy gor multiclass.

Mohith hooda[2] et. al proposed the data preprocessing with data sampling and feature ranking.

Datta H. Deshmukh [3] et.al proposed classification using labels and the feature selection by fast corelation based filter technique and again final preprocessing with with discretization method to enhance the classification process.

In [4] the authors discussed about different techniques feature selection such as filter. Wrapper embedded and hybrid methods to select the feature. Then after feature selection the applied three different models such as random forest, Adaboost . neural network and naïve base classifiers.

Many researchers used the KDD data set and NSL data set for various attacks.

III. Data Set And Types Of Attack In Dataset

A. Dataset:

There are several kinds of data set available for research purpose to test and train the data. Here in this research KDD+ data set is considered to identify attack in IDS. Data set includes the network connection, connection's characteristics, connection status, protocols, services, and associated labels for the type of attack. They are represented by means data points. The KDD+ dataset is mainly used to train and test the model for identifying 4 different types of attacks. This research focuses on enhancing the efficacy of the machine learning algorithms for the detection of cyber attacks by selecting and extracting the features especially in DOS/DDOS attack category using RFE, Maximum variance and min max method.

The dataset contains 41 features and they all are grouped under some categories such as basic features, connection features, traffic features, host based features, flag features and error related features. Not all the features are important for intrusion detection. Features that are relevant can be grouped together or duplicate information available in features can be removed to improve the performance of the system. To identify DOS/DDOS attack category nearly 20 features are considered. Usually features to be considered for the DoS/DDoS attacks are duration, protocol, service, source byte, destination bytes, number of connection as count feature, service error rate, request error rate, server request error rate, rate of requests to the same service, rate of requests to different services, flag indicators, rate of connections to the same host, and destination host count, among others.

Let us have a look at the following data point which was taken from the KDD+ dataset for the more clear idea 0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,229,10,0.00,0.00,1.00,1.00,0.04,0.06,0.00,255,10,0.04,0.06,0.00,0.00,0.00,0.00,1.00,1.00,neptune,21

Here the first attribute refers to the duration of the connection, tcp indicates the protocol used for the connection, private specifies the service or application associated with the connection, REJ denotes the connection status as rejected, then subsequent values represent various numerical features related to the connection, such as total number of connections, number of connections to the same service, connection error rate, error rate for connections to a specific service. The value 1.00 indicates the label or class associated with the connection, here it is "neptune’, 21 represents the port number used for the connection. The feature that is not relevant to the attack can be removed during the preprocessing stage. The following are the few methods used in preprocessing stage to improve the accuracy of the system.

B. How to improve accuracy?

Accuracy can be improved by removing the total number of features. Overall system performance is affected by more number of irrelevant features utilized for testing and training the sample. Accuracy of the system will be improved if the count of features are reduced during testing and training process. If the feature extraction or selection are not done properly then even after reducing the features the system accuracy will go down and if the important features are removed by mistake then also system accuracy will go down. The system also oversimplified if it removes the important and relevant features used in training phase. Incorporating pertinent features while omitting non-essential ones can streamline the system and reduce complexity.

The following fig.3.1 gives the clear idea to identify the type of attack. Initially raw data set will be fed into the system which will be having several attributes for network connections. Feature extraction stage will remove the unwanted features, duplicate values and noise to reduce the test and training time. Once after extracting the feature

the next step is to identify the feature which will be having more weightage to classify the attack. Then with this data the system will be trained and tested to classify the attack based on the label for attack and non attack types.

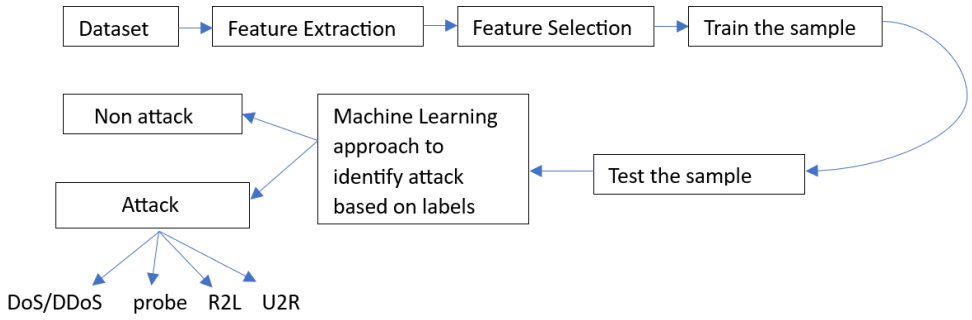

Fig 3.1. Logic to identify the attacks and non attacks

C. Attack types in the dataset:

Here four categories of attacks with their relevant attributes are available in the dataset. Each line of dataset is called data point which consists of different features used for connection status, number of packets, source destination, flags etc., In the dataset, labels are used to identify the type of attack based on the class attribute. They are normal, Neptune, DoS, smurf and back. The data set contains 4 different types of attack with 41 common attributes. For the research purpose Dos attack alone is considered. The labels with back, neptune, smurf, land, teardrop and pod etc. are considered as DoS attack. The other labels used in data set are satan, ipsweep, nmap, portsweep, mscan, imap, phf, warezclient, warezmaster, snmpgetattack, snmpguess, guess_passwd, buffer_overflow, rootkit, ftp_write, loadmodule, multihop, perl, spy etc. These labels are used to identify the normal network connections, probe, user to root attack, remote to local attacks respectively.

The normal label represents normal network connections, indicating legitimate and non-attack traffic. The neptune label corresponds to the Denial of Service (DoS) attack category. It represents attacks that aim to exhaust the resources of a target system or network, making it unavailable to legitimate users. The smurf label is also associated with the DoS attack category which refers to a specific type of DoS attack that uses Internet Control Message Protocol (ICMP) packets to flood a target network with ICMP echo requests. The 'back' label is another representation of the DoS attack category and refers to a DoS attack known as a TCP/IP stack-based "land" attack, where a malicious actor sends TCP packets with the source and destination IP addresses set to the same value, causing the target system to become unresponsive. The satan label is associated with the Probing attack category. It represents network scanning activities where an attacker tries to gather information about a target network or system without actively attempting unauthorized access. These mappings are based on the labelling provided in the KDD+ Cup 1999 dataset and represent common attack types found in the simulated network environment. Labels are used to map the attack category such as DoS and non attack category as normal in this case.

IV. PREPROCESSING

A. Need to pre process the data

It is always a good idea to preprocess data before sending it to any machine learning approach. This approach can enhance the model's precision and efficiency, First and foremost thing is data quality. Raw data may contains noise, inconsistencies, missing values, and outliers. Preprocessing is the step which helps in cleaning and transforming the data into a usable format, It will ensure that the model receives high-quality data. Preprocessing can help to reduce the size of the data set, which in turn reduce the training time for the model. This is especially important for large data sets. If the model is a complex model then preprocessing can help to improve the performance of the model by making it more efficient and interpretable to understand how the model works and debugs.

Many machine learning algorithms perform better when features are on the same scale. This is called feature scaling. Preprocessing techniques like normalization and standardization scale features to a specific range, preventing some features from dominating others during model training. Normalizing data ensures that all features have similar ranges and distributions. This helps in cases where algorithms like gradient descent are used, as it can converge faster and more reliable. Real-world data often contains missing values, which can lead to biased or inaccurate results Preprocessing strategies provide ways to manage incomplete data, including imputation (substituting missing values with calculated estimates) or elimination of instances with missing data.

Machine learning models typically require numerical input, hence categorical variables must be transformed into a numerical representation. This is called as categorical data encoding. Techniques like one-hot encoding or label encoding are used for this purpose. High-dimensional data can be computationally expensive and may lead to overfitting. To reduce dimensionality methods like Principal Component Analysis (PCA) are effective in preserving vital information while diminishing the complexity of the data. Preprocessing involves in noise reduction to eliminate irrelevant or redundant information, making the data more informative for the model. Outliers can skew the model's training process and affect its performance. Data preprocessing can involve identifying and handling outliers appropriately, ensuring they do not unduly influence the model.

In natural language processing (NLP), preprocessing of text data is essential, involving processes such as tokenization (breaking down text into individual words or phrases), eliminating stop-words, applying stemming, and lemmatization to condense the vocabulary and enhance data quality. Time series data often requires normalization or differencing to remove trends and seasonality, making it stationary and suitable for modelling. Data preprocessing can sometimes act as a safeguard against data leakage, which occurs when test or validation set information inadvertently seeps into the training phase, resulting in deceptively favourable outcomes. Through data preprocessing, the data is formatted appropriately for the training process and helps the model learn the patterns and relationships more effectively. Proper preprocessing enhances the model's performance, generalization, and robustness.

B. Various methods to preprocess data

The efficiency of any classifier will be improved by preprocessing the data present in the dataset. Feature extraction and feature selections are having major role to decide upon the features based on the requirement. So feature selection algorithm are mandatory to refine the data for better analysis by the classifier. Some algorithms should be used to select the features before feeding the data into the classifier which in turn improve the efficiency and overall process of the classifier. There are various methods available to select the relevant features.

Feature importance is the method which will consider only the important features based on rank. The rank will be decided by certain algorithms based on the contributions given to the model performance. The features with low importance will not increase the model’s performance. Low importance features or low rank features should be removed for further processing.

The next method is Correlation Analysis which is used to identify highly correlated features and retain only one of them to avoid redundancy. Using this method duplicate values can be removed. If two or more features have strong correlation, then keep only one representative feature among the same correlated group and discard the other one.

Univariate Feature Selection method uses statistical tests or scoring method to decide upon the features. ANOVA F-value, chi-square test, mutual information, or correlation coefficient are some common techniques used in this method. These techniques use rank based classification. Using this method subset of top-ranking features will be selected based on a predefined threshold value and only those features are retained for further processing.

Principal Component Analysis is a widely utilized technique for reducing dimensionality that maps data to a lower-dimensional space, preserving a substantial portion of the data’s variance. It accomplishes this by identifying the principal components that encapsulate the most critical information within the dataset. PCA can be used to reduce the feature dimensionality before training the Random Forest model.

The choice of feature reduction technique is contingent upon the characteristics of the data, the particular issue being addressed, and the objectives of the analytical endeavour. It is often recommended to experiment with multiple methods and evaluate the impact of feature reduction on the model's performance, such as accuracy, precision, recall, or F1-score, to determine the most effective approach for your specific dataset and classification task.

Recursive Feature Elimination (RFE) is a methodical approach to feature selection that systematically removes the least significant features through iteration. It involves training a Random Forest model on all available features and then determining the importance of each feature. The least important feature(s) are then removed, and The model undergoes retraining on the narrowed set of features. This iterative process continues until the target number of features is attained or a predetermined stopping point is achieved. This method will train the model based on the importance of the feature and eliminates the one with less importance. The best feature selection method for a specific data set and machine learning model will depend on the characteristics of the data set and the goals of the classification task. However, Recursive Feature Elimination is a general-purpose feature selection method that is

often effective for DoS attack classification. RFE was found to be the most effective feature selection method for improving the accuracy of DoS attack classification using the KDD+ dataset. In this research RFE was able to improve the accuracy of the random forest classifier by up to 10%.

V. METHODOLOGY

5.1 Recursive Feature Elimination (RFE) Method

For this research purpose random forest method is considered for identifying the DoS/DDoS attack category and to preprocess the data, Recursive Feature Elimination method along with feature importance by ranking is considered to improve the overall performance of the classifier. RFE can be used to select the features that are most important for all of the decision trees in the random forest. RFE can be used to select the most important features for classifying attacks. The Random Forest algorithm operates by constructing multiple decision trees, with each tree being trained on a randomly selected subset of the features. Benefits of RFE with Random Forest for Attack Classification:

Here are some benefits of using RFE in Random forest method to classify the DoS attack in KDD+ data set. Some of the most important benefits includes the improved accuracy, reduced training time, improves interpretability, reduce the overfitting, improves the generalization of the model and improves the robustness of the model

RFE will help to improve the accuracy of the Random forest classifier by selecting the most important features for classification. Applying this method is particularly advantageous for datasets with numerous features, as it aids in diminishing data noise and bolstering the classifier's efficacy.

RFE can help to reduce the training time of the Random forest classifier by reducing the number of features that need to be processed. This can be especially beneficial for data sets with a large number of features, as it can significantly reduce the amount of time it takes to train the classifier.

RFE is used to improve the interpretability of the Random forest classifier by identifying the most important features for classification. This can be helpful for understanding how the classifier is making predictions and for debugging the classifier if it is not performing as expected.

The accuracy of the Random forest classifier is increased up to 10% and training time is reduced upto 50 % with RFE. RFE will reduce the risk of overfitting: Overfitting frequently occurs in machine learning models, particularly when dealing with small or cluttered datasets Recursive Feature Elimination (RFE) can mitigate overfitting by isolating the most crucial features for classification, potentially enhancing the model's accuracy on new data.

Generalization ability is the ability of a machine learning model to make accurate predictions on unseen data. Recursive Feature Elimination (RFE) can enhance a model's ability to generalize by choosing a subset of features that best represent the dataset. This can help to prevent the model from overfitting to the training data and improve its accuracy on unseen data.

Robustness is the ability of a machine learning model to make accurate predictions in the presence of noise or outliers. RFE can help to improve the robustness of a model by selecting a subset of features that are less sensitive to noise and outliers. Utilizing this approach can enhance the model's accuracy with data that may not be entirely clean.

Overall, Recursive Feature Elimination is an effective technique that can be utilized to enhance the precision of a model, reduce the training time, improve the interpretability, reduce the risk of overfitting, improve the generalization ability, and improve the robustness of Random forest classifiers for DoS attack classification.

Selecting the feature in RFE helps in identifying the most informative features for attack classification. This can lead to a more interpretable model and improved generalization. By eliminating less important features, RFE can help reduce overfitting, especially when dealing with a high-dimensional feature space. When there is a large dataset training a model with fewer features is computationally more efficient which in turn improve the efficiency of the algorithm. With the small set of features it is easy to interpret the models decisions and predictions. The success of RFE mainly depends on the quality of the features, the dataset, and the chosen machine learning algorithm.

The significance of a feature in a decision tree model is determined by its contribution to reducing impurity, measured by the Gini impurity index. The Gini index method arranges trees from the most to the least impure. This ranking facilitates the creation of a set of key features. Mathematically, this process involves partitioning each node (A) in a decision tree to minimize its impurity (R(A)), represented by the Gini index. The Gini index is calculated

by subtracting the sum of the squared probabilities of each class from one. For a subset (A) containing samples of class (Z), the Gini impurity (Gini(A)) is defined as follows:

$R(Z)=Gini(A)=1-\sum(P_i)^2$

Probability of class (i) within the subset. Upon dividing a node (Z) into two distinct nodes (Z1) and (Z2), each with data sizes (N1) and (N2) respectively, the Gini index for the split can be expressed using the following formula:

$Ginisplit(Z)=N1/N\ Gini(Z1)+N2/N\ Gini(Z2)$

Here, (Gini(Z1)) and (Gini(Z2)) represent the Gini impurity of nodes (Z1) and (Z2), and (N) is the total number of samples before the split.

Overall, RFE is a robust method that enhances the accuracy, reduce the training time, and improve the interpretability of Random forest classifiers for DoS attack classification.

The RFE algorithm works as follows:

Step 1: The RFE algorithm first trains a random forest model on the original data set with all available features. Then the dataset need to be split for training, validation and testing based on the requirements.

Step 2: Then calculate the importance of each feature. This is calculated by measuring how much the feature contributes to the accuracy of the random forest model.

Step 3: Remove the least important feature from the data set. Here Gini index is used for ranking the important feature

Step 4: Retrains the random forest model on the data set with the feature with least importance removed.

Step 5: The above two steps are repeated until the desired number of features remains.

The features that are removed are the ones that contribute the least to the accuracy of the random forest model. This means that the remaining features are the most important for classifying attacks.

Once the RFE model has been trained, it can be used to classify new data points. To classify a new data point, pass the features of the new data point to the RFE model. The RFE model will then predict the attack type for the new data point.

The following example gives the better understanding of the RFE method. Let us consider RFE algorithm selects the 4 most important features for classifying attacks in the KDD+ data set. The features that are selected are: The duration of the connection in seconds by means of duration feature, the type of protocol used in the connection with protocol_type, the service that was used in the connection with service attribute and src_bytes attribute for the number of bytes sent from the source system. The importance of each feature is printed out. The feature with the highest importance is duration, followed by protocol_type, service, and src_bytes. These features are important because they can be used to distinguish between DOS attacks and legitimate traffic. For example, DOS attacks often have a long duration, use a common protocol like TCP, and target common services like HTTP. By focusing on these features, the RFE algorithm can improve the accuracy of DOS attack classification. First Load the dataset and preprocess the data set. Then select the features that can be used for classification all features to be included in this step. Next Create the random forest classifier by assigning the n-estimator value as 100, with the number of trees, maximum depth etc., Initialize the RFE classifier with the number of important feature. Here in this case 4 features. Fit the RFE model to data. Get the importance of each feature by using Gini index for ranking the important feature and print it.The stopping criteria can be the cross validation method to estimate performance during each iteration and stop when performance starts to degrade or once after reaching the required number of important features.

Let us consider the following values for the datapoint

duration = 100

protocol_type = tcp

service = http

src_bytes = 1000

The RFE model predicts that this data point is a normal connection because the values of the features are all within the normal range for a normal connection. The overall process of RF model with gini index is explained in the following Fig.

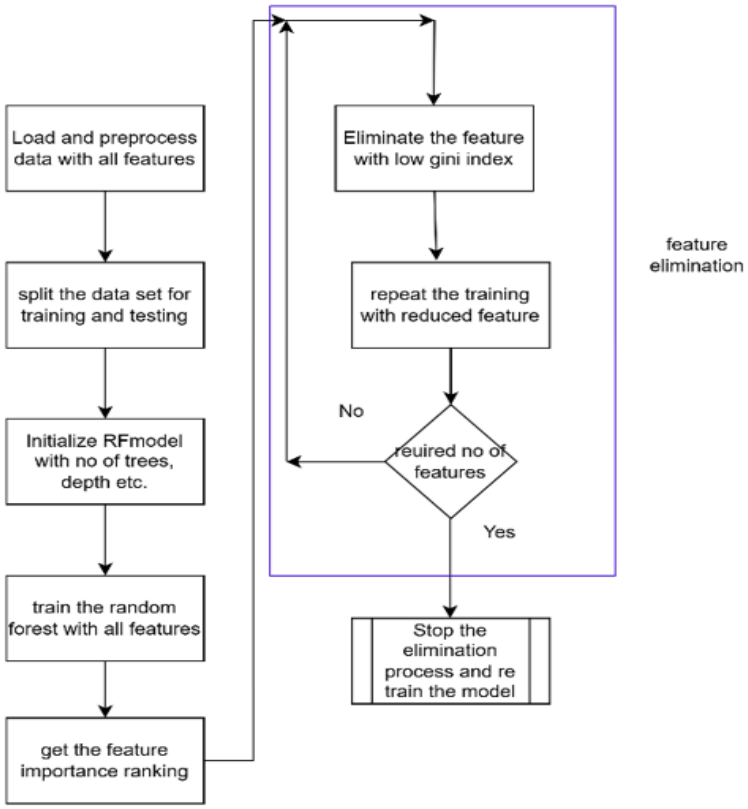

Fig. 5.1 RFE in Random forest classifier

In summary the RFE algorithm can be used with any machine learning model that can be trained on a data set with multiple features. It can be used to select the most important features for any task, not just attack classification. RFE can be used to select a fixed number of features or to select a percentage of the features. The RFE algorithm can select features based on their importance or on their ranking. In general, the Gini importance of a feature can vary depending on the data set and the specific type of attack that is being identified. The higher the Gini importance of a feature, the more important the feature is for identifying DDoS attacks. However, the src_bytes feature is often found to be the most important feature for identifying DDoS attacks, because the number of bytes sent from the source IP address can be used to identify attacks that involve sending a large number of packets.

Let's use the first sample data point to explain how random forest selects important features in the KDD+ dataset to identify DDoS attacks using the Gini index. The Gini index is a measure of how pure a node is. A pure node is one where all the data points belong to the same class.

The first step is to calculate the Gini importance of each feature in the data point. The Gini importance of a feature is calculated as follows:

Gini importance = Σ(decrease in Gini index)

where the decrease in Gini index is the amount by which the Gini index of a node is reduced when the feature is used to split the node. The higher the Gini importance of a feature, the more important the feature is for identifying DDoS attacks.

The following is the example after including the gini Index. The Gini importance of the features in the first sample data point are:

protocol_type: 0.0001

service: 0.0004

flag: 0.0012

src_bytes: 0.9983

The src_bytes feature has the highest Gini importance, which means that it is the most important feature for identifying DDoS attacks in this data point. The protocol_type and service features have very low Gini importance, which means that they are not very important for identifying DDoS attacks in this data point.

RFE can be a valuable tool for improving the performance of random forest models for attack classification. By selecting the most important features, RFE can help to reduce the size of the data set and improve the accuracy of the model. It is important to note that the importance of features can vary depending on the data set and the machine learning model being used. RFE algorithm is a powerful tool that can be used to select the most important features for a variety of tasks, including DOS attack classification.

5.2 Maximum Variance algorithm

There are three main purpose of using Max variance algorithm. First one is to reduce the dimensionality, next one is to identify the high variability across the instances and the last one is to maintain the important and informative feature for further classification. This algorithm aims to reduce dimensionality and at the same time it retains the most significant features used for classification. This algorithm is very much helpful to reduce the features in intrusion detection method. By focusing on these high-variance features, MaxVar helps capture essential information while minimizing redundancy. By selecting features with high variance, it is intentionally focus on those that exhibit substantial variability across different instances. Usually high-variance features are more likely to capture anomalous patterns, to identify network intrusions. By retaining only the most informative features, computational complexity is also reduced and in turn will enhance model performance.

Let us consider the data point from KDD+ data set which has features such as protocol, duration, REJ flag, service etc.

First high variance features across the data points are identified because those features contribute to the overall variance in the data point. If the duration feature is zero then it wont contribute much to the over all variance in the datapoint.

The Flag REJ specifies a rejected connection attempt. If in case this flag varies significantly across instances (e.g., some connections are accepted while others are rejected), it could be a high-variance feature. Similarly if the Service features specifies the Private service the it might not vary much in this specific data point, but across the entire dataset, it could exhibit variability. This algorithm focuses on relevant aspects of the data while minimizing noise. So this methods plays a important role in optimizing the feature selection in Intrusion Detection System.

Algorithm

1. Consider all the features initially
2. Identifies features with high variability across instances.
3. Retains features that capture the most significant variations.
4. Remove the feature which is not contributed much to the overall variance
5. Repeat the steps 2 to 5 in all the instances

1. Feature selection

a. Compute the variance of each feature across all instances.

b. Select the top-k features with the highest variance (where k is determined based on domain knowledge or experimentation).

c. These high-variance features capture the most significant variations in the data.

2. Dimensionality Reduction

a. Apply techniques like PCA (Principal Component Analysis) to further reduce dimensionality or min max algorithm to normalize the data within some range

b. PCA identifies linear combinations of features that explain the maximum variance or min max to remove the outliers

c. The resulting reduced feature set retains essential information while minimizing redundancy.

3. Anomaly Detection

a. Train an anomaly detection model (e.g., Isolation Forest, One-Class SVM, Random Forest Classifier) using the selected features.

b. Anomalies (intrusions) are instances that deviate significantly from the expected patterns captured by high-variance features.

c. The model can identify network anomalies based on these features.

5.3Min max Method

The goal of Min max algorithm is used to transform the features (variables) to specific range (usually between 0 and 1 or -1 to 1). This scaling ensures that all features have a consistent scale, which is essential for various machine learning algorithms.

This algorithm works in the following way.

1. Apply the min max scaling individually to all the features

2. Compute the scaled value using the following formula for each feature (x),

[x_{text{scaled}} ={{x - x_{text{min}}}}/{{x_{text{max}}- x_{text{min}}}}]

Where (x) represents the original value of the feature.

(x_{text{min}}) is the minimum value of that feature in the dataset.

(x_{text{max}}) is the maximum value of that feature in the dataset.

The numerator ((x - x_{text{min}})) computes how far the original value (x) is from the minimum value.

The denominator ((x_{text{max}} - x_{text{min}})) represents the range of possible values for that feature.

By dividing the difference between (x) and (x_{text{min}}) by the total range, feature is normalized to [0, 1] interval.

This ensures that all features lie within the range [0, 1].

3. Train the intrusion detection system such as the random forest classifier or neural network or deep learning model using scaled features.

Let's see with example

When (x = x_{text{min}}), the scaled value becomes 0.

When (x = x_{text{max}}), the scaled value becomes 1.

Intermediate values of (x) are linearly transformed to lie within this range.

Suppose if the minimum duration observed in the dataset is 10 seconds and the maximum duration is 100 seconds

x_{text{min}} = 10

x_{text{max}} = 100

compute the scaled duration for any specific connection time (x):

x_{text{scaled}} = (x – 10)/(100 – 10)

For instance:

If a connection lasted 30 seconds, its scaled duration would be:

x_{text{scaled}} = (30 – 10)/(100 – 10)

= 20 / 90

= 0.22

If a connection lasted 80 seconds, its scaled duration would be:

x_{text{scaled}} = (80 – 10)/(100 – 10)

= 70 / 90 = 0.78

These scaled values can now be used in machine learning models without causing issues due to varying scales.

It scales features proportionally to their original range and ensures consistent scales for all features. This method is useful for various machine learning algorithms. MinMax scaling ensures that all features have a consistent scale, and prevents any single feature from dominating the model to Normalize the values. Outliers are easily handled by rescaling features to a bounded range. Many Machine Learning Models will perform better when features are within a similar range.Data Imbalance will be easily handled by MinMax scaling by ensuring both normal and attack instances contribute equally. Usually MinMax scaling is often applied after other preprocessing steps like data denoising and feature selection.

VI. Results And Discussion

The dataset consists of nearly 1 lakh and 25 thousand datapoints with 41 features and each data point represents status of the connection with features such as duration, service, protocol type, flag, src bytes, dst bytes, server etc. All the features are not useful to build the model and they may reduce the overall accuracy of the system. For the research purpose three different algorithms are considered to check the accuracy of the model by retaining the needed feature and eliminating the un wanted features. The results shows nearly 10 to 20 % increase in the accuracy and very good improvement in the overall

Accuracy is calculated to evaluate the performance of the system. The following chart gives the comparison between the accuracy calculation with all 41 features and accuracy calculation after applying the RFE algorithm to select only the important feature.

Table 1. Accuracy and F1 score before and after applying the algorithms

Method	Accuracy	F1 score
All features	85.8	83.227
After using RFE algorithm	95.6	98.8
After using Maximum variance method	98.34	98.02
After using Min max method	99.96	99.7

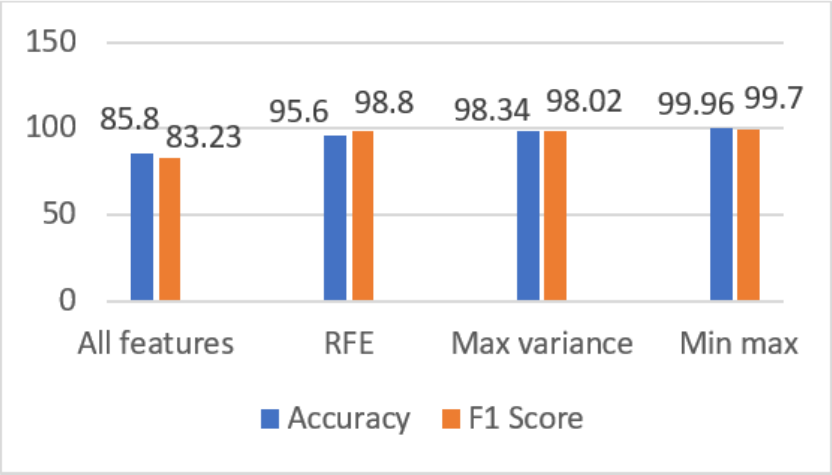

Fig. 6.1 Accuracy and F1 score using 3 algorithms

VII. CONCLUSION

Choosing the data set is the first step in any ML approach. Then Preprocessing takes the next step to make the data ready for any processing. There are several different classifiers are available to classify the data into the proper group using supervised or unsupervised learning models. In this research paper the reason for preprocessing, various methods to preprocess the data are discussed. Even though there are several techniques available only few techniques are very good in selecting the features. The result section clearly gives the accuracy and F1 score before and after applying the feature selection algorithm. When the model uses the feature selection methods then the performance of the system will be increased and complexity will be decreased. If the dataset contains imbalanced features then more than one feature selection algorithms can be used select the features and normalize the data.

REFERENCES

[1] Shadi Aljawarneha, Monther Aldwairia,b, Muneer Bani Yasseina, Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model,2016,Elsevier, Journal of computational science

[2] Mohit Hooda, Jameer Babu, P. Surya Vamsi, Gopakumar G, An Improved Intrusion Detection System Based on KDD Dataset Using Feature Ranking and Data Sampling, 2020, International Conference on Communication and Signal Processing.

[3] Datta H.Deshmukh et. al, Improving Classification Using Preprocessing and Machine Learning Algorithms on NSL-KDD Dataset, International Conference on Communication, Information & Computing Technology.

[4] Mikołaj Komisarek et. al, How to Effectively Collect and Process Network Data for Intrusion Detection?, MDPI, Entropy.

[5] Alsirhani, A., Sampalli, S. and Bodorik, P., 2019. DDoS detection system: Using a set of classification algorithms controlled by fuzzy logic system in apache spark. IEEE Transactions on Network and Service Management, 16(3), pp.936-949.

[6] Gutierrez, J.P. and Lee, K., 2021. High Rate Denial-of-Service Attack Detection System for Cloud Environment Using Flume and Spark. Journal of Information Processing Systems, 17(4), pp.675-689.

[7] Liu, C., Gu, Z. and Wang, J., 2021. A hybrid intrusion detection system based on scalable K-Means+ random forest and deep learning. IEEE Access, 9, pp.75729-75740.

[8] Rajesh Thomas et. al, A Survey of Intrusion Detection Models based on NSL-KDD Data Set, The Fifth HCT INFORMATION TECHNOLOGY TRENDS (ITT 2018), Dubai, UAE

[9] Zhenpeng Liu, A Deep Random Forest Model on Spark for Network Intrusion Detection, Hindawi Mobile Information Systems Volume 2020, Article ID 6633252, 16 pages.

[10] K. Pramilarani, et. al "Using MapReduce and Time Series Analysis to Defend Against DDoS Attacks", -, "Advances in Transdisciplinary Engineering series ,Volume 32,Recent Developments in Electronics and Communication Systems, January 2023.

[11] K.Pramilarani," Social Media and Bigdata", International Journal For Research & Development In Technology ,2017

[12] K.Pramilarani, Vasanthi kumari P ,"Analyzing Various Techniques to Safeguard user Sensitive Data" , in IJERT, Volume 9, Issue 6 , June – 2020.

[13] Ms. Pramilarani K et. al , Empowering Law Enforcement and Emergency:An In-Depth Exploration of Criminal Face Detection Technology Integrated with Emergency SOS Sending Protocols, IEEE Explorer, 2024 4th International Conference on Data Engineering and Communication Systems (ICDECS).