National College of Ireland

Using Machine Learning in Intrusion Detection System to Improve Model Efficiency and Reduce Training Time Using Different Feature Selection Methods and Classifiers

MSc Research Project
MSc in Data Analytics

Naveen Rao Vittal Rao
Student ID: X22130276

School of Computing
National College of Ireland

Supervisor: Dr. Taimur Hafeez

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Naveen Rao Vittal Rao
Student ID:	X22130276
Programme:	MSc in Data Analytics **Year:** 2023-2024
Module:	Research in Computing
Supervisor:	Dr. Taimur Hafeez
Submission Due Date:	14-12-2023
Project Title:	Using Machine Learning in Intrusion Detection System to Improve Model Efficiency and Reduce Training Time Using Different Feature Selection Methods and Classifiers
Word Count:	9695 **Page Count:** 26

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Naveen Rao Vittal Rao

Date: 14-12-2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Using Machine Learning in Intrusion Detection System to Improve Model Efficiency and Reduce Training Time Using Different Feature Selection Methods and Classifiers

Naveen Rao Vittal Rao
X22130276

Abstract

This study aims to address the critical need of enhancing Intrusion Detection Systems (IDS) by the strategic application of Machine Learning (ML). The study's main objective is to shorten training times and boost model effectiveness by utilizing a range of feature selection techniques and classifiers. The comprehensive experience includes the creation of machine learning models, astute visualization, and painstaking data pre-processing. Significant findings reveal the ongoing dominance of the Random Forest model and provide insight into the subtle differences in the effects of various feature selection techniques on various classifiers. The study presents a possible avenue for bolstering cybersecurity frameworks by providing useful insights into the adaptability and robustness of machine learning in the context of intrusion detection systems.

1 Introduction

In the rapidly evolving subject of cybersecurity, machine learning (ML) stands out as a revolutionary force. Because intrusion detection systems (IDS) are crucial for safeguarding digital environments, they need to be constantly improved for optimal performance. This study addresses the crucial need to integrate machine learning into intrusion detection systems, with a focus on improving model efficiency and decreasing training time. Our goal is to fully utilize machine learning in the intrusion detection industry by utilizing a variety of Feature Selection Methods and Classifiers.

In order to ensure the integrity of the dataset, the thorough examination begins with strict data pre-treatment procedures. Visualization techniques offer valuable insights into the subtleties of the data and lay the foundation for the subsequent creation of machine learning models. One of the primary objectives is to determine which feature subsets and classifiers will maximize the overall effectiveness of IDS. The Random Forest model consistently outperforms alternative classifiers, as seen by our results. The study also clarifies the distinct benefits of a few feature selection techniques, including Select from Model, Variance Threshold, SelectKBest, and ANOVA. These discoveries paved the way for a deeper understanding of the connection between feature selection and classifier efficacy.

Despite the study's successes, we also acknowledge its limitations, providing a framework for future investigation. Our research contributes significantly to the evolving intersection of IDS and ML, which strengthens the dynamic and vital domain of cybersecurity measures.

2 Related Work

2.1 Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset.

This review of the literature looks at anomaly-based intrusion detection systems (AIDS) that use machine learning. Ziadoon Kamil Maseer's criticism highlights issues like outdated datasets and inadequate research in relevant papers. Ten popular supervised and unsupervised machine learning approaches were utilized to construct efficient ML-AIDS for networks in order to address these problems. Among the prominent methods are ANN, DT, k-NN, NB, RF, SVM, CNN, EM, k-means, and SOM. Evaluation utilizing the CICIDS2017 dataset revealed that the k-NN-AIDS, DT-AIDS, and NB-AIDS models outperformed the others in terms of identifying web attacks. However, no single system was able to precisely recognize every type of attack. The study proposes a benchmarking strategy to improve evaluation fairness and highlights the importance of benchmarking in advancing intrusion detection technologies.

2.2 Intrusion Detection Technique in Wireless Sensor Network Using Grid Search Forest with Boruta Feature Selection Algorithm.

As the use of computers and the internet increases, so does the need of network security. Sridevi Subbiah and Kalaiarasi Sonai Muthu Anbananthen's study investigates intrusion detection in wireless sensor networks (WSNs). Through the application of machine learning (ML) to WSN intrusion detection systems (IDS), the study addresses shortcomings of traditional ML techniques like SVM and KNN, which have demonstrated low accuracy and a high misclassification rate. They provide a unique strategy called Boruta feature selection with grid search random forest (BFSGSRF), which outperforms earlier machine learning techniques like LDA and CART. The NSL-KDD dataset experiment findings show that the BFS-GSRF model has an astounding 99% accuracy rate in detecting assaults. Additionally, the BFS-RF approach improves classifier performance significantly by achieving a 99.9% accuracy rate.

2.3 An Improved Binary Manta Ray Foraging Optimization Algorithm Based Feature Selection and Forest Classifier for Network Intrusion Detection.

Ibrahim Hayatu Hassan's research aims to create an intrusion detection model by combining a Random Forest (RF) classifier with an enhanced Binary Manta Ray Foraging (BMRF) Optimization Algorithm. The report highlights the challenges caused by the Internet's rapid growth in terms of producing new threats while emphasizing the necessity of intrusion detection systems (IDs) for network security. It emphasizes the use of metaheuristics and machine learning to boost intrusion detection rates. The proposed approach utilizes BMRF for feature extraction and RF for feature evaluation.

2.4 IDS-ML: An Open Source Code for Intrusion Detection System Development using Machine Learning.

IDS-ML is an open-source Python repository designed to streamline the process of developing intrusion detection systems (IDSs), as demonstrated by the work of Li Yang and Abdallah Shami. The employment of both traditional and state-of-the-art Machine Learning (ML) algorithms sets IDS-ML apart. IDS-ML aims to simplify the difficult task of identifying and reacting to different attacks by integrating these algorithms, providing researchers and cybersecurity specialists with an adaptable platform. Rapid creation and replication of intrusion detection system (IDS) frameworks that deal with a variety of network traffic datasets is made possible by it.

2.5 Intrusion Detection System: A Comprehensive Review.

Hung-Jen Liao and Chun-Hung Richard Lin's study provides a thorough taxonomy of intrusion detection systems (IDSs) and highlights the difficulties in handling various threats while using a substantial amount of computational power. It explores the complexities of IDS, demonstrating the advantages and disadvantages of different detection strategies. But the study focuses mostly on challenges that are still unresolved with IDSs, like managing security difficulties in wireless configurations, enhancing real-time performance, and creating VM-specific IDS solutions in cloud environments. Although the IDS methodology and issues are thoroughly described, the emphasis is placed on the ongoing need for research to meet the ever-evolving complexities of intrusion detection and network security.

2.6 Machine Learning Based Intrusion Detection System.

The focus of Anish Halimaa A and Dr. K. Sundarakantham's research is machine learning-based intrusion detection systems. In order to meet the growing demand for network security as computer connectivity develops, the study analyses network traffic data using Support Vector Machine (SVM) and Naive Bayes techniques. Using the NSL-KDD dataset, the examination demonstrates that SVM outperforms Naive Bayes in processing a large number of cases. In addition to highlighting the importance of intrusion detection and prevention in modern network systems, the conclusion emphasizes how well SVM handles huge datasets. Taking everything into account, the study highlights how vital it is to continuously advancing intrusion detection methods in order to fend off novel forms of network attacks.

2.7 A Detailed Analysis of CICIDS2017 Dataset for Designing Intrusion Detection System.

The CICIDS2017 dataset was introduced by the Canadian Institute of Cybersecurity as one of several recommendations made recently in the field to assess the effectiveness of intrusion detection systems (IDS) against the most recent threats. The CICIDS2017 dataset appeals to researchers since it exposes threats that haven't been studied previously, yet this article points out significant vulnerabilities that potentially distort the detection algorithms of traditional IDS. The authors analyse the dataset's attributes and find issues that limit the dataset's use to intrusion detection. The research proposes a combined dataset that addresses the issues discovered in response to these challenges, with the goal of enhancing the classification and detection capabilities of future intrusion detection engines.

2.8 Intrusion Detection System.

The author offers a brand-new intrusion detection method that combines sampling and Least Square Support Vector Machine (LS-SVM). The system is called OA-LS-SVM. The

approach initially splits the dataset into subgroups and then selects the most representative samples using an optimal allocation mechanism based on subgroup variability. Following that, the data are used to train the classifier using the LS-SVM. In terms of accuracy and efficiency, the results of OA-LS-SVM are encouraging. It can handle large datasets and be used in binary-class and multiclass scenarios. The authors show that OA-LS-SVM outperforms other existing methods in terms of accu0racy, efficiency, and scalability.

2.9 A Novel Statistical Technique for Intrusion Detection System.

The study looks into feature selection inside Dimensionality Reduction (DR) techniques to address the problems caused by the exponential growth of data. It examines Feature Selection (FS) methods and classifies them into three categories: Embedded, Wrapper, and Filter based on how well they function with learning models. It draws attention to the trade-offs between processing speed and accuracy as well as the advantages of dimension reduction in terms of lowering data noise. Future research objectives are highlighted, including the need for models suitable for changing data and selection based on genetic algorithms.

2.10 A Deep Learning Approach to Network Intrusion Detection.

We provide a novel deep learning method for Network Intrusion Detection Systems (NIDSs) called NDAE. It tackles the problems that NIDSs are currently facing, emphasizing data volume, the need for thorough monitoring, and a range of network protocols. In an attempt to enhance anomaly detection, deep learning is being researched due to the drawbacks of conventional signature-based techniques. The proposed model, which combines RF for classification and NDAE for feature learning, demonstrates good accuracy and low training periods using the KDD Cup '99 and NSL-KDD datasets. Outperforming Deep Belief Networks (DBNs), it offers the benefits of NDAE and suggests improvements for handling zero-day assaults in the future.

2.11 A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection.

This study looks on machine learning-driven intrusion detection methods to solve cybersecurity problems. It discusses low-frequency intrusion detection and looks into the classification of attacks. It discusses the challenges in cybersecurity and provides instances, including the DDoS attack on Estonian websites and assaults on major platforms.

The research examines intrusion detection systems that are host-based and network-based, also known as HIDS and NIDS, and classifies detection methodologies into three groups: hybrid, abuse, and anomaly. It explains machine learning methods (BP-ANN, DT C4.5, SVM) for intrusion detection. It assesses the advantages and restrictions of signature-based anomaly and abuse detection methods. The amalgamation of misuse and anomaly approaches in hybrid detection highlights the possibility of improving accuracy and decreasing false positives. Because the study recognizes the benefits of machine learning in IDS, including its ability to adapt to new attack types, it focuses on four machine learning-based IDS settings.

The paper evaluates various intrusion detection techniques, emphasizing their benefits and providing recommendations for further research, despite the lack of exact quantitative data. Its contributions include attack classification, comprehensive literature reviews, performance evaluations, and suggestions for future advancements in machine learning-based intrusion detection techniques.

2.12 Deep Learning Methods in Network Intrusion Detection: A survey and an objective comparison.

Deep learning models for network intrusion detection (IDS) have been extensively studied during the last 20 years. However, there is a lack of empirical evaluation of these models, especially with regard to the more recent standard IDS datasets. An IDS deep learning model taxonomy, survey, and benchmark implementation are provided in this study. This study evaluates the performances of four primary models: feed-forward neural network, auto encoder, deep belief network, and long short-term memory network. The evaluation makes use of both modern datasets (CIC-IDS2017, CIC-IDS2018) and historical datasets (KDD 99, NSL-KDD). The findings demonstrate that deep feed-forward neural networks are effective across all datasets in terms of accuracy, F1-score, training time, and inference speed. Surprisingly, the researchers discovered that deep belief networks, auto encoders, and unsupervised learning models did not outperform supervised feed-forward neural networks. These results suggest that deep feed-forward neural networks are the most effective deep learning models for IDS. The authors also highlight the limitations of the existing literature and make suggestions for future directions in the field of machine learning approaches to intrusion detection systems.

2.13 Deep Learning Approach for Intelligent Intrusion Detection System.

This study investigates the application of machine learning methods, specifically deep neural networks (DNNs), to develop an intrusion detection system (IDS) that can recognize and classify cyberattacks on hosts and networks. This paper proposes a highly scalable hybrid DNNs framework called Scale-hybrid-IDS-AlertNet. This system continuously monitors host-level events and network traffic with the goal of providing proactive alarms for potential cyberattacks. The paper also looks at the shortcomings of existing machine learning-based intrusion detection systems, addressing issues including high false positive rates, inadequate generalizability, and the need for scalability to keep up with the ever expanding size and dynamics of networks.

2.14 Deep Learning in Intrusion Detection Systems.

The paper discusses the importance of intrusion detection systems (IDSs) for network cybersecurity. IDSs are crucial for identifying and thwarting the constantly evolving cyber threats. This study focuses on the decision engine—a vital component of intrusion detection systems. There is a lot of potential for improving the decision engine with deep learning and other machine learning approaches. Deep learning can easily manage large datasets and be updated dynamically to learn new attack strategies. A number of IDS systems, including those based on anomaly and signatures, are examined in the study. Signature-based systems are good at recognizing existing attacks, but they struggle to recognize new ones. Anomaly-based systems have a higher percentage of false positives even though they are better at spotting new attacks. The report also discusses the challenges of applying machine learning to IDS. One challenge is the amount of time needed to train machine learning models. Another challenge is the demand for large training data sets. To get around these problems, deep learning makes use of parallel processing and its ability to learn from smaller datasets.

2.15 Enhanced Intrusion Detection System using Feature Selection Method and Ensemble Learning Algorithms.

The paper discusses the importance of feature selection for intrusion detection systems (IDSs). The authors divided the input dataset into subsets according to the various sorts of attacks. They then offer a feature selection method for each subset that uses an information

gain filter. The optimal feature set is then produced by combining the lists obtained for each attack. Experimental results demonstrate that this feature-light approach improves system accuracy while reducing complexity based on the NSL-KDD dataset. Throughout the study, the efficacy of several categorization methods in the context of feature selection is contrasted. Based on a voting learning method, the research adds an additional stage using Random Forest and PART to further increase performance. The results demonstrate that the product probability rule produces the best accuracy. The paper also suggests techniques for anomalous intrusion detection in Wireless Sensor Networks (WSNs). The authors discuss the challenges posed by resource constraints in wireless sensor networks (WSNs) and emphasize how important it is to defend these networks against cyberattacks. The proposed methods are evaluated on the NSL-KDD dataset, demonstrating improved accuracy and reduced complexity.

2.16 Feature Selection in Machine Learning: A New Perspective.

The study discusses the difficulties associated with processing high-dimensional data in data mining and machine learning. Feature selection is a helpful strategy for increasing the efficacy of learning by removing content that is redundant and superfluous. An detailed overview of various supervised, unsupervised, and semi-supervised feature selection methods for machine learning tasks, including classification and clustering, is provided in this study. The study distinguishes between feature extraction and feature selection, noting that the former focuses on selecting relevant features from the original dataset, while the latter involves transforming data into features with strong pattern recognition skills. The study discusses how machine learning models evaluate feature selection strategies, emphasizing the importance of both minimal processing overhead and good learning accuracy. The study concludes by highlighting the advancements made in supervised, unsupervised, and semi-supervised feature selection methods while acknowledging the need for additional research and the difficulties that still need to be tackled.

2.17 Intrusion Detection Technique in Wireless Sensor Network using Grid Search Random Forest with Boruta Feature Selection Algorithm.

The crucial issue of protecting wireless sensor networks (WSNs) from assaults that aim to impede network functionality is addressed in this work. It emphasizes how crucial intrusion detection systems (IDS) are to wireless sensor networks (WSNs) in order to detect unknown threats. Network and host security must be guaranteed as computer-related applications continue to proliferate exponentially and as the use of the Internet spreads. WSN vulnerabilities are covered in the first section of the study. These vulnerabilities can be brought on by a number of factors, such as open-air transmission, varying network topology, and inadequate physical equipment. Owing to these characteristics, WSNs are susceptible to security threats such as spoofing, hijacking, eavesdropping, and jamming. The paper then discusses the many forms of IDS, which are separated into two groups: host-based (HIDS) and network-based (NIDS). The paper then discusses the various detection techniques, including hybrid, anomaly, and signal-based techniques. Signature-based approaches are employed to detect attacks by comparing network traffic with pre-identified attack signatures. Anomaly-based methods, which search for variations in the network's usual behaviour, are used to identify attacks. Hybrid approaches combine signature-based and anomaly-based techniques. The study then discusses the use of machine learning (ML) techniques in IDS. The accuracy of intrusion detection can be raised by employing machine learning (ML) algorithms to learn from enormous amounts of data. The paper discusses a number of machine learning (ML) techniques, including support vector machines (SVMs), random

forests, and k-nearest neighbours (KNNs), that can be used with intrusion detection systems (IDS). The paper then proposes a novel IDS architecture that combines the Boruta feature selection method with a grid search random forest (BFS-GSRF) classifier. The system is tested using the NSL-KDD dataset, a well-liked dataset for anomaly detection in WSNs. According to the results, BFS-GSRF outperforms traditional ML systems with 99% accuracy in identifying attacks. The potential of IDS in WSNs are discussed in the study's conclusion. The study identifies a number of problems that need to be fixed, such as reducing the amount of time needed for training, enhancing the classifier's efficacy for real-time deployment, and developing more accurate and dependable detection techniques.

2.18 Supervised Feature Selection Techniques in Network Intrusion Detection: A Critical Review.

The paper investigates the significance of Machine Learning (ML) approaches, specifically focusing on Feature Selection (FS), for network intrusion detection. It highlights the existing issues with biases in classification, training durations, and feature variety. The paper offers a thorough examination of machine learning (ML)-based intrusion detection, evaluating available datasets and comparing several FS techniques, including state-of-the-art bio-inspired algorithms and conventional rank search. It conducts experimental evaluations of temporal complexity, feature correlation, and performance. The study's first part looks at the increasing dimensionality and diversity of variables that characterize network data traffic. Because of this, machine learning systems have a harder time identifying intrusions, which may lead to biased classifications and drawn-out training times. FS appears as a crucial pre-processing step that reduces the feature space to save only the most relevant characteristics in order to get over these problems. The article emphasizes how FS improves the functionality of ML systems and offers a comprehensive review of the application of ML methods in intrusion detection. Various FS tactics, including filter, wrapper, and embedding techniques, are discussed. Beyond the standard KDD99 dataset, other datasets are examined in the research in order to assess the effectiveness of different FS approaches. According to the authors' testing results, FS significantly improves ML systems' ability to detect intrusions. By reducing the feature space, FS improves classification accuracy and reduces training times. The research also evaluates the temporal complexity of different FS approaches to determine their applicability for real-time intrusion detection systems.

2.19 Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges.

This paper offers a thorough introduction to intrusion detection systems (IDS), going into the difficulties that different types of IDS—signal-based and anomaly-based—face and how machine learning might improve IDS performance. Citing the shortcomings of previous surveys, the research emphasizes the necessity for an updated taxonomy and a thorough analysis of recent papers in this field. It addresses the problems that modern IDS must overcome and provides a taxonomy of IDS tactics. The paper evaluates evasion tactics, points out shortcomings in current datasets (such DARPA/KDD99), and highlights the need for additional large-scale datasets that fairly depict malware activity in the contemporary era.In order to effectively handle the always changing landscape of cyber risks and intrusion detection problems, the paper's conclusion suggests utilizing surveys and datasets that are already available.

3 Research Methodology

Methodology provides an understandable way forward. It outlines the steps and processes needed to complete the project's objectives. Clarity enhances a methodology's efficiency. It allows proper agenda beginning, task organization, and resource allocation. For the Intrusion Detection System (IDS) project, preprocessing and analyzing the CICIDS2017 dataset needed an accurate methodology. The procedure includes data import, exploration, cleaning, visualization, encoding, and feature selection. The following is a list of the steps taken.

3.1 Data Importation

Importing the CICIDS2017 dataset is adding a sizable amount of structured data. I have imported the CICIDS2017 dataset into multiple data frames due to its large size and the data from multiple days were stored in different dataset file. A specific subset or attribute of the dataset is represented by each of these data frames. Next, I have concatenated all the data frame into one consolidated data frame as we don't want to keep these data frames apart after we've loaded the dataset into them. I have created a single, comprehensive data frame by combining all the multiple data frames as shown in the Fig 1, Data importation flow below.

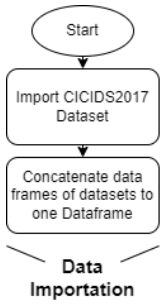

Fig 1: Data importation flow

3.2 Data Exploration and Cleaning

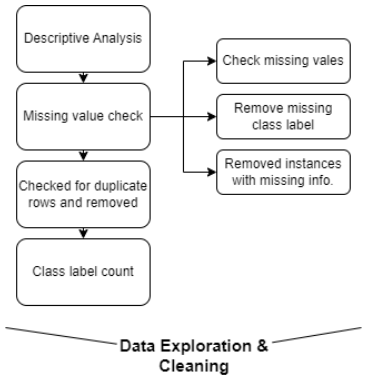

Fig 2: Data exploration and Cleaning

a. Descriptive Analysis of the dataset
   Firstly, I conducted a descriptive analysis of the dataset, which provided the statistics such as mean, standard deviation, minimum, maximum, and quartiles for each feature. This step helped to understand the central tendency, spread, and distribution of the data.
b. Handled the missing values in the dataset

Next, I checked for missing vales in the dataset and found that there were 1358 missing values in the dataset. To address this missing values, I have removed the instances with the missing values as it makes sure the quality and analysis of the results after modelling.

c. Checked for Duplicate rows and handled
   Checked for the duplicate rows from the dataset and I was able to find that there were 307,376 duplicate rows in CICIDS2017 dataset. The duplicate rows play major role in skewing the analysis and results after modelling, so I removed the duplicate rows and made sure that each instances in my dataset is unique.
d. Checked for final dataset shape
   Next, I checked the dataset shape to make sure the cleaning process is successful and the final dataset shape had 2,829,358 instances and 79 features.
e. Performed class label count
   To understand the class label distribution of CICIDS2017 dataset which is important for classification problem, performed the class label count.
f. Replaced Null values with NaN
   As a standard approach I have replaced the null values with NaN values keeping consistency and python libraries works seamlessly with NaN and also it helps in sharing the data without any missing vales.
g. Value count
   To provide a summary, performed the value count of each occurrences with respect to their class.
h. Class label distribution of CICIDS2017 dataset
   For building the balanced ML model, I have performed class label distribution among the different class and showed the count of instances for each class.
i. Histogram to Visualize the numerical features
   Plotted the histogram to visualize the distribution of each numerical features to understand the pattern and data spread of each independent variable.
j. Correlation matrix calculation
   Calculated the correlation matrix of the features to identify the relation in between each features which helps us in the main task of our project which is feature selection in the future section.
k. The next section will explain about the visualization of the cleaned data.

3.3 Class Label Analysis

The class label count analysis is very important in order to understand the different classes and its distribution and it can affect the results and performance of the model, because model with the imbalanced dataset will lean towards the majority class data.

a. Performed initial class label count before pre-processing and understood different classes in the dataset which gave an idea about classifying the attack to a particular class.
b. Performed post-processing class label count after handling the missing values and removing the duplicate values and replacing null value with NaN value made sure that pre-processing will not skew the distribution of class in CICIDS2017 dataset.

c. Visualized the class label count which helped in finding the imbalances in the data which will help during the model building and evaluation step.

3.4 Data Visualization.

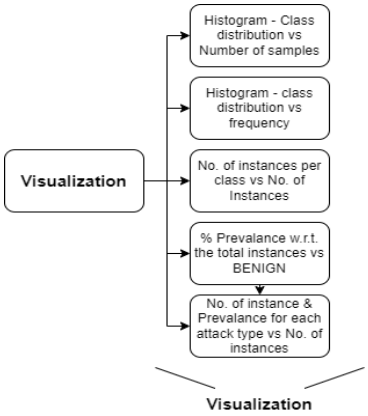

Fig 3: Visualization overview

Visualization gives us an insight of the data and helps us in processing the data furthermore which helps us take necessary decision if required before modelling and training the data and predicting the results.

a. Histogram plot for Class distribution to number of samples:

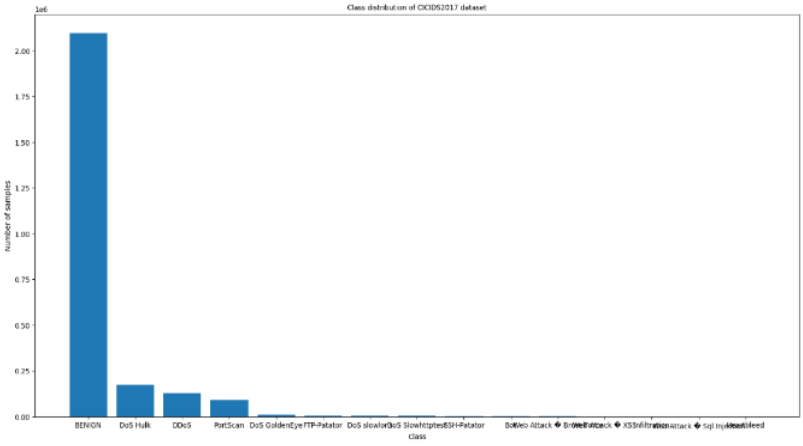

Fig 4: Number of instances per class in CICIDS2017 dataset against Number of instances

The histogram plot in Fig 4, shows, the class distribution vs. number of samples, in which each class appears as a bar against the frequency of that specific class, shows the distribution of the classes and their samples. This plot provided to the differences between the various classes. The histogram plot of the CICICDS 2017 dataset provides valuable insights into the distribution of different features. Of specific interest is the "Begin" feature, which has an extremely high incidence of 2.00 on a scale from 0 to 2.0. The starting point of a network flow is indicated by the "Begin" feature.

The peak at 2.00 suggests that a considerable proportion of the samples have a "BENIGN" value that is in this range, indicating a normal pattern or behavior during the early stages of network flows.

The representation shows the varying frequency of various attack types. "DoS Hulk" has a rather high occurrence of 0.23, indicating that a considerable portion of the dataset contains

this specific type of Denial-of-Service attack. Parallel to this, "PortScan" exhibits notable port scanning activity with a frequency of 0.13. However, "DoS GoldenEye" appears less frequently at 0.02, suggesting the persistence of another type of DoS attack.

The attack categories "FTO-Patator," "DoS Slowloris," and "Slowhttp" all have attributes that show a frequency of 0.10, indicating that these types of attacks occur in the sample rather frequently. These results provide important insights on the distribution of various attack types and the frequency related to them, as well as the nature of the dataset.

The importance of the dataset lies in its wide range of attributes related to various aspects of network traffic, like flow time, packet lengths, flags, and statistical metrics. This provides a detailed summary to aid in understanding the characteristics of the dataset. Plotting's distinctive spikes and patterns highlight specific behaviors and attack types, opening the door for more research and the creation of a useful intrusion detection model.

b. Bar plot explaining class distribution vs frequency:

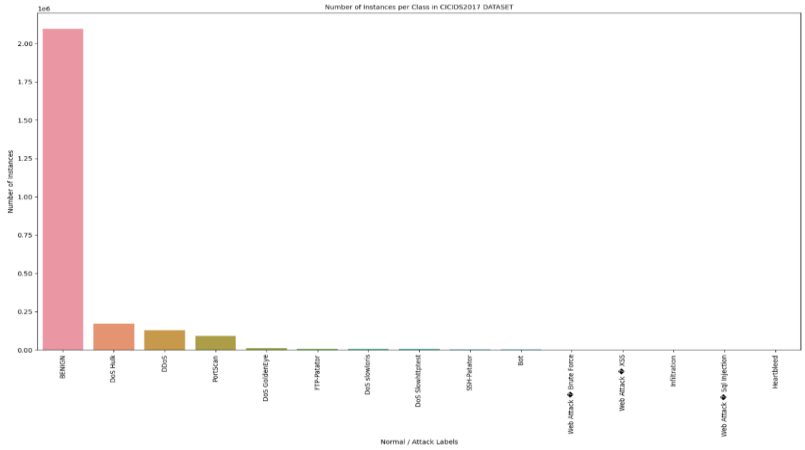

Fig 5: Attack types vs Number of Instances

The histogram in Fig 5 illustrates the number of instances by class in the CICIDS2017 dataset provides a clear and insightful overview of the distribution of the different attack categories. In the range of 0 to 2.0, the "BEGIN" class stands out strongly due to its considerable occurrence of 2.15. This means a significant number of the dataset's instances belong to the "BEGIN" class, showing how frequently this particular type of network activity or communication start occurs.

The "DoS Hulk" class follows closely with a frequency of 0.235, showing a significant prevalence of Denial-of-Service attacks defined by high traffic volume. In addition, the class "DDoS" appears 0.125 times per second, highlighting instances of Distributed Denial-of-Service attacks involving several compromised machines.

A significant number of instances involving searching of network ports is shown by the significant frequency of 0.12 shown by the "PortScan" class, which represents port scanning operations. Next, with a frequency of 0.03, the "DoS GoldenEye" class shows instances of a particular kind of Denial-of-Service attack.

Also, based on their respective frequencies of 0.05, which is the attack types "FTO-Patator," ”DoS Slowloris," and "DoS Slowhttptest" show a moderate presence. Further, the "SSH-

Patator" and "Bot" categories have a lower occurrence rate of 0.01; this indicates a relatively lower frequency of occurrences connected to SSH-based attacks and botnet-related activities. This histogram provides useful information on the overall structure of the CICIDS2017 dataset by clearly visualizing the distribution of cases across different types of attacks. The different frequencies provide a basis for further study and the development of robust intrusion detection models that are able to recognizing and categorizing a wide range of network security threats. These models highlight the variety of attack types and their frequency of occurrence.

c. Prevalence rate using pie chart

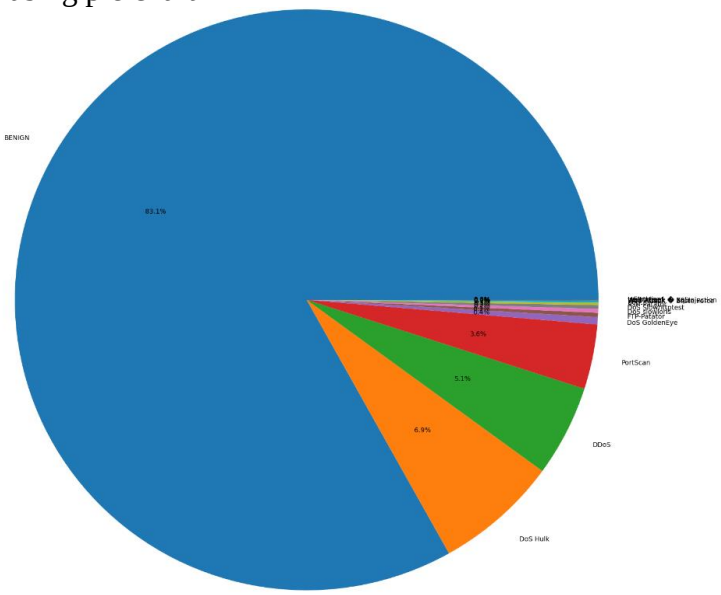

Fig 6: Prevalence rate pie chart

The pie chart in Fig 6, that shows the majority of different categories within the CICIDS2017 dataset providing an overview of the distribution of network instances across various attack types. At 83.1% of the total, the "BEGIN" class is the largest slice on the chart. It also indicates that an important part of the information refers to the first phases or common communication patterns observed in network traffic.

Next that, the class known as "DoS Hulk" accounts for 6.9% of the total, indicating a notable frequency of occurrences associated with Denial-of-Service attacks, which are defined by unusually high traffic volumes. The third-largest amount, or 5.1%, is accounted for by the "DDoS" class, which describes instances of Distributed Denial-of-Service attacks involving several hacked machines.3.6% of all occurrences in the dataset belong to the "PortScan" class, which shows port scanning actions, showing that network ports are occasionally examined. With a smaller slice of 0.4%, the "DoS GoldenEye" class comes in second, indicating instances of attacks that use a particular strategy to create a loss of service.

The attack classes that are still included in the dataset, such "FTP-Patator" and "DoS Slowloris," each show a relatively low prevalence of less than 0.3%. This pie visualization effectively communicates the equal number of instances across several attack categories and offers information on the prevalence and makeup of the main network security risks found in the CICIDS2017 dataset.

d. Attack types to Number of instances with Prevalence rate (%)

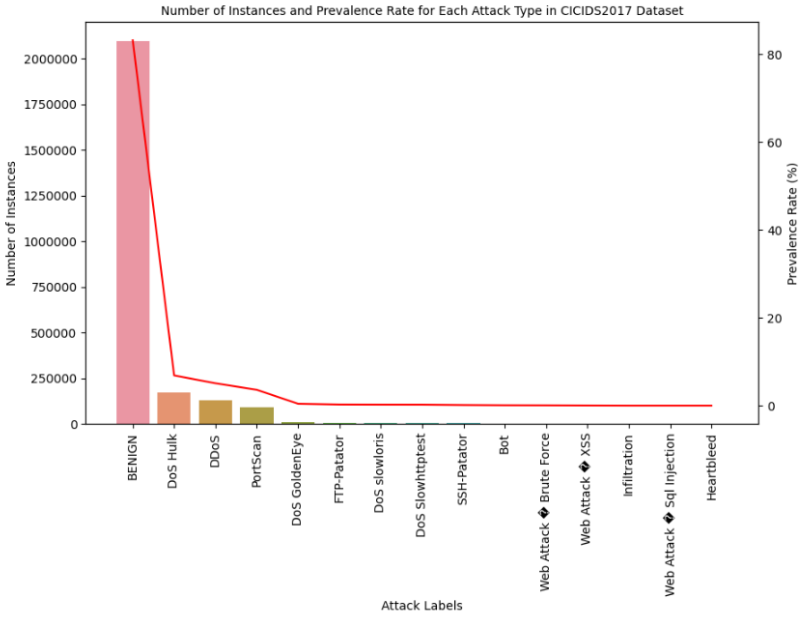

Fig 7: Attack types vs Number of instances and Prevalence rate (%)

The bar graph in the above figure 7 with the line graph overlay that displays attack types vs. incident count and prevalence rate (%) is an essential instrument for intrusion detection system(IDS).

The complete dataset can be viewed because the range of occurrences is from 0 to 2,000,000. The BENIGN label, which suggests typical network traffic, is the most common among the over 2,500,000 examples. The line graph starts at this point, and all subsequent attack types are visually compared to this baseline. The data visualization shows the distribution of different attack types with reference to the most frequent BENIGN cases. It's interesting to see that DoS Hulk has more than 200,000 instances; its line on the graph is at 250,000, indicating how frequent it is relative to normal traffic. Next is DDoS, with approximately 180,000 instances, or a line at 215,000. There are 150,000 PortScan instances shown by a line at 175,000. As the graph expands, DoS GoldenEye shows a distinct dominance with about 18,000 occurrences, matching a line at 150,000. FTP-Patator, on the other hand, has a higher frequency in the dataset as seen by its line stabilizing at roughly 135,000 and its lower count of approximately 3,000 occurrences.

3.5 Data Pre-processing

The next step is to prepare the data using Standard Scaler. It is less probable that issues will arise from different scales when the characteristics are uniformly defined, making sure the model can reliably predict outcomes and generalize over a range of features. When machine learning models are applied to the specified dataset, these preprocessing steps significantly improve their precision and robustness.

3.6 Feature Selection

Feature selection, which aims to identify and keep the most relevant features from the less useful ones, is an important step in the machine learning process. Through this process, the model's interpretability is enhanced, generalization is increased, and computing complexity is

decreased. In the context of our research challenge, when increasing accuracy is the main objective, so the feature selection step is crucial.

I have chosen four feature selection technique given below.

1. Variance Threshold Feature Selection Technique
   This method eliminates features with low variance on a presumption that they are less informative. It is especially useful when interacting with binary features that are mostly constant.
2. SelectKBest Feature Selection Technique
   Using data analysis, a unitary feature selection method known as SelectKBest selects the top k features. This method is effective in selecting characteristics based on their relative importance and ranking them.
3. SelectKBest with ANOVA Feature Selection Technique
   A statistical test that can be performed to evaluate how group means differ from one another is called a study of variance, or ANOVA. It becomes important for classification tasks by aiding to identify features that have a significant impact on the target variable when combined with SelectKBest.
4. Select from Model Feature Selection Technique
   With this technique, features are selected based on significance scores after the initial model has been trained. For instance, a tree-based method allows features to be prioritized based on their level of impact on impurity reduction. This is effective in locating non-linear relationships within the data.

Our research aims to explore and utilize the advantages of each strategy through the use of the different feature selection methods, possibly finding areas of overlap that result in improved model accuracy. It is simpler to identify which variables are most important to our specific research problem when these approaches are continuously evaluated and compared, and it also offers an in-depth understanding of feature importance which plays important role in the performance of the model.

3.7 Model Building

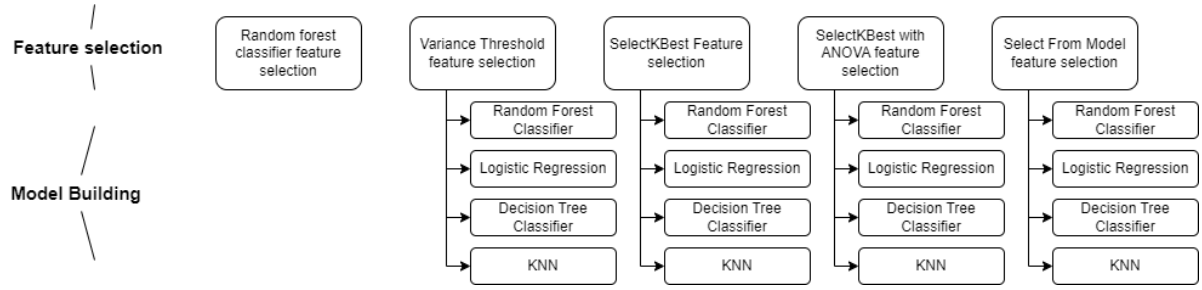

Fig 8: Model building with different feature selection technique

In order to increase the precision and effectiveness of intrusion detection systems, my study method digs into the field of machine learning models (IDS). It specifically focuses on several feature selection methods through the use of a number of classifiers. The Random Forest (RF) Classifier, Logistic Regression, Decision Tree, and k-Nearest Neighbours (KNN) are the main models chosen for our IDS using CICIDS2017 dataset.

1. Random Forest Classifier
   The RF Classifier was selected because of its reliability, adaptability, and ability to handle complex, non-linear connections in the data. The ensemble nature, which combines several decision trees, enables it to precisely represent complex patterns.

Because of the complexity of network intrusion patterns, RF help in feature selection by providing insights on feature significance.

2. Logistic Regression
   Considering its growing popularity as a solution for binary classification problems, logistic regression is especially helpful for differentiating between malicious and benign network operations. Its readability, ease of use, and effectiveness with large datasets seem good for IDS applications, which have practical requirements.
3. Decision Tree
   The Decision Tree model was selected because it offers an accurate representation of the model's modelling process and has an open decision-making process. Network administrators need to understand the decision process in an IDS setting in order to comprehend the basis of threats that have been knew and take the appropriate measures.
4. K-Nearest Neighbours (KNN)
   KNN is a proximity-based algorithm that specializes at recognizing patterns based on similarities. When it comes to intrusion detection systems (IDS), KNN's preference for neighbour proximity helps identify a variety of intrusion cases when anomalous patterns do not correspond to defined parameters.

Decision on model selection

1. Random Forest with Ensemble approach
   Because of their ensemble nature, RF is more suited to detecting the complex and dynamic nature of network intrusions. Feature importance insights enhance the understanding of the relevance of features.
2. Efficiency and Simplicity with Logistic Regression
   Logistic regression is a useful tool for processing large-scale datasets that are commonly utilized in network traffic studies because of its simplicity of usage. Because of its linear nature, it can be used to identify basic patterns of malevolent conduct.
3. Interpretability using Decision Tree
   Decision trees are inherently interpretable because the decision processes that result in alerts in intrusion detection systems (IDS) are crucial to comprehend. It provides a clear illustration of how the model distinguishes between behaviour that is acceptable and annoying.
4. KNN – Proximity based learning using KNN
   Because regulations may not always be complied with, KNN's exceptional ability to identify patterns based on similarity is helpful in capturing a range of infiltration scenarios. It functions well in dynamic network situations due to its adaptability.

By combining a wide range of models with different feature selection strategies, my work seeks to improve accuracy and identify the optimal mix of algorithms and feature subsets for intrusion detection systems. This approach's several facets guarantee a thorough investigation of the correlation between the model and attributes, ultimately enhancing intrusion detection abilities.

3.8 Model Evaluation

The Intrusion Detection System (IDS) project must go through a model evaluation phase in order to assess the results of various feature selection strategies as well as the efficacy of various machine learning models. The evaluation measures employed, which comprise precision, accuracy, F1 score, macro-average, and weighted average, give a comprehensive understanding of each model's effectiveness.

Data Pre-Processing

The model was thoroughly pre-processed before the model evaluation process was initiated. The CICIDS2017 dataset, a sizable collection of network traffic statistics, underwent thorough inspection and cleaning. Descriptive analysis helped to understand data distribution by revealing significant statistical insights. To maintain consistency across the dataset, duplicate rows were removed, missing values were handled, and NaN was used in place of null values. A comprehensive analysis of class labels was performed to ensure a uniform distribution for effective model training.

Data Visualization

A significant portion of the dataset's intricacies may be understood because of the visualization. Histogram plots showing the class distribution against the total number of incidents shed light on the prevalence of different types of assaults. Bar charts showed the frequency of each sort of assault, highlighting the dominance of particular classes. A pie chart was used to visually represent the prevalence rate of various assault categories, giving users a thorough grasp of the dataset composition. The bar graph with a line overlay provided a dynamic picture of attack occurrences, emphasizing the significance of each assault type in respect to the benign class.

Feature Selection

Feature selection is an important step in the process that tries to increase the model's interpretability, generalization, and computing performance. The four different procedures that were used were Select from Model, SelectKBest, SelectKBest with ANOVA, and Variance Threshold. Through a systematic process of identifying and preserving the most critical traits, these tactics ensured optimal model performance. The advantages of each selection technique were based on the capacity to capture non-linear relationships and statistical significance.

Model Building

Four distinct machine learning models—Random Forest Classifier, Logistic Regression, Decision Tree, and k-Nearest Neighbors (KNN)—were selected to be tested in the following step, which was model building. Every model was carefully connected to one of the four feature selection processes in order to produce an extensive collection of model-feature combinations. Random Forest was chosen over Logistic Regression because of its ensemble technique's ability to capture intricate patterns while maintaining efficiency and simplicity. Decision Trees' well-defined decision-making process made them interpretable, but KNN excelled at proximity-based learning, which is a crucial task in dynamic network scenarios.

Model Evaluation

The models were thoroughly assessed using metrics including weighted average, macro-average, F1 score, accuracy, and precision. Following that, a comprehensive examination of the model's performance for each class was included in a classification report that was generated. Precision was used to gauge the model's ability to correctly identify samples; accuracy was used to indicate overall correctness; the F1 score was used to balance precision and recall; and weighted averages and macros were used to indicate both class-specific and overall performance.

In summary, the laborious research procedure involving feature selection, data pre-processing, visualization, and model construction allowed for a complete evaluation of the model. The chosen

metrics and classification report offer a thorough understanding of the benefits and drawbacks of each model, facilitating well-informed decisions for enhancing IDS capabilities.

4 Design Specification

With the primary objectives of enhancing model performance and reducing training time, Intrusion Detection Systems (IDS) employ feature selection algorithms and classifiers that are covered in detail in this section. Our work focuses on identifying the optimal combination for improved performance by applying multiple classifiers and feature selection strategies.

1. Feature Selection Techniques

a. Variance Threshold:

- **Description:** Variance thresholding eliminates low variability features because it thinks they are less valuable. This is quite useful when working with binary features that exhibit little change.
- **Functionality:** The strategy aims at increasing model efficiency by focusing on the most informative characteristics and reducing the dimensionality of the dataset.
- **Requirements:** This method uses less processing resources, hence it is suitable for initial feature filtering.

b. SelectKBest:

Description: SelectKBest is a univariate feature selection strategy that assigns a unique relevance value to each characteristic in order to determine the top k features.

Functionality: SelectKBest is used to order features according to their significance and helps determine which qualities are most relevant for the model.

Requirements: It requires access to feature importance scores, although it functions well in scenarios when each individual feature's significance is critical.

c. SelectKBest with ANOVA:

Description: By including an analysis of variance (ANOVA) to evaluate how features affect the target variable in classification problems, this method enhances SelectKBest's feature selection procedure.

Functionality: ANOVA provides statistical insights into feature relevance, making it easier to identify features that significantly affect the target variable.

Requirements: It is necessary to have access to pertinent books and computer resources because statistical computations are involved.

d. Select from Model:

Description: Select from After the initial model training, the model comprises selecting features according to their relevance scores. For instance, features are ranked using tree-based approaches depending on how efficiently they minimize contaminants.

Functionality: It is feasible to detect non-linear relationships in the data more easily by ranking features based on how effectively they affect the model's performance using this strategy.

Requirements: It is required to have both model training and access to feature significance scores.

2. **Classifier ML algorithm**
   a. **Random Forest Classifier:**
      Description: By merging multiple decision trees, the RF Classifier is an ensemble learning strategy that handles non-linear correlations in the data and enhances model performance.

      Functionality: The RF Classifier helps with feature selection by providing information on the relative relevance of each feature in the ensemble.
      Requirements: Perfect for scenarios where complex network intrusion patterns necessitate a group strategy.
   b. **Logistic Regression:**
      Description: Logistic regression is a widely used method for binary classification problems because it is effective and simple to use in distinguishing between harmful and benign network events.
      Functionality: Because of its linear structure, it can be used to find fundamental trends in harmful behaviour across large-scale datasets.
      Requirements: Perfect in circumstances when user-friendliness and understanding are
   c. **Decision Tree:**
      Description: Network administrators can use Decision Tree models, which provide an open representation of the decision-making process, to better understand the basis of risks they have identified.
      Functionality: Because decision tree models are naturally interpretable, they improve the transparency of intrusion detection systems.
      Requirements: helpful when comprehensibility of the decision-making process is a primary concern.
   d. **KNN:**
      Description: KNN is a proximity-based algorithm that can detect patterns based on similarities, making it effective at identifying various kinds of intrusions.
      Functionality: KNN's adaptability makes it effective at spotting non-compliant patterns in dynamic network environments.
      Requirements: Perfect for circumstances where identifying a range of intrusion patterns is crucial.

5 Implementation

The Implementation stage provides various output of the IDS implementation which are listed below.

1. **Outputs from the project**
   a. **Data Importing from Drive where dataset is situated.**
   b. **Transformed data:**

The CICIDS2017 dataset is cleaned and transformed and post transformation it had 2,829,358 instances and 79 features.

c. **Feature selection:**
   Four feature selection technique were finalized and used which is Variance Threshold, SelectKBest, SelectKBest with ANOVA and Select from Model and each selected features uniquely.

d. **Model Building:**
   Four models were decided and used and the reason of selecting this algorithm are explained in the model evaluation section and the models are Random Forest classifier, Logistic Regression, Decision Tree and KNN.

e. **Visualizations:**
   Plotted Number of instances per class in CICIDS2017 dataset against Number of instances, Bar plot explaining class distribution vs frequency, Prevalence rate using pie chart and Attack types to Number of instances with Prevalence rate (%).

f. **Evaluation metrics:**
   The model build is evaluated using accuracy, precision, F1 score, macro average, weighted average and with classification report consisting of these metrics for each class

2. **Tools and Languages used in the project**
   Programming language used: Python
   Libraries used: Pandas, Scikit-Learn, Matplotlib, seaborn, Numpy,
   Models built: Random forest, Logistic Regression, Decision Tree and KNN.
   Data Processing: Google colab

6 Evaluation

By combining a number of feature selection strategies with a variety of classifiers, the study aimed to enhance Intrusion Detection System (IDS) performance. The following analysis and key findings are derived from the model outputs:

6.1 Variance Threshold feature selection with Random Forest classifier

With Variance Threshold feature selection, the Random Forest classification model achieves an impressive 99% accuracy. This implies a high degree of capacity to discriminate between unauthorized and authorized network activities.

6.2 Variance Threshold feature selection with Logistic Regression

Combining Variance Threshold feature selection with logistic regression yields a good accuracy of 89%. It does somewhat worse than the Random Forest model, but it still does a great job in classification tasks.

6.3 Variance Threshold feature selection with Decision Tree

The Decision Tree model, when paired with Variance Threshold feature selection, achieves 99% accuracy. This shows how well the dataset's complex decision boundaries are captured.

6.4 Variance Threshold feature selection with KNN

Utilizing Variance for threshold feature selection, the k-Nearest Neighbors (KNN) model yields 99% accuracy. This illustrates how KNN complies with IDS standards by identifying patterns based on proximity.

6.5 SelectKBest feature selection with Random Forest classifier

When the SelectKBest feature selection technique is combined with the Random Forest classifier, a 99% accuracy rate is attained. This implies that focusing on the top k pertinent features greatly increases the model's accuracy.

6.6 SelectKBest feature selection with Logistic Regression

Combining SelectKBest feature selection with logistic regression yields an accuracy of 89%. This illustrates how effective logistic regression is when feature relevance takes precedence.

6.7 SelectKBest feature selection with Decision Tree

The Decision Tree model exhibits 99% accuracy when SelectKBest feature selection is used. This bolsters the decision tree's efficiency in categorization tasks—particularly when elements that are pertinent are highlighted.

6.8 SelectKBest feature selection with KNN

A 99% accuracy rate is achieved when the k-Nearest Neighbors (KNN) model is used in conjunction with SelectKBest feature selection. This emphasizes how flexible KNN is in concentrating on the dataset's most important attributes.

6.9 SelectKBest with ANOVA feature selection with Random Forest classifier

ANOVA and SelectKBest are coupled to yield a high accuracy of 99.60% for the Random Forest classifier. This demonstrates how the ANOVA statistical test may be used to effectively identify features that have a substantial impact.

6.10 SelectKBest with ANOVA feature selection with Logistic Regression

An amazing accuracy of 99.60% is achieved with logistic regression when paired with SelectKBest and ANOVA. This illustrates how well ANOVA and logistic regression work together.

6.11 SelectKBest with ANOVA feature selection with Decision Tree

Even though it only achieves an accuracy of 79.87% when SelectKBest and ANOVA are applied, the Decision Tree model demonstrates the intricate effects of feature selection techniques on diverse models. Further investigation may shed light on how the components in this scenario interact.

6.12 SelectKBest with ANOVA feature selection with KNN

SelectKBest and ANOVA are utilized to obtain a high accuracy of 98.62% for the KNN model. This illustrates how statistically significant traits discovered by ANOVA may be utilized to customize KNN.

6.13 Select from Model feature selection with Random Forest classifier

With an accuracy of 99.79%, the Random Forest classifier performs exceptionally well when used with the Select from Model feature selection method. This method, which applies knowledge from the trained model, works quite well.

6.14 Select from Model feature selection with Logistic Regression

Choose from When combined with Logistic Regression, the model maintains a respectable 88.73% accuracy. It's not as high as some other models, but it still demonstrates the potential utility of this tactic in specific circumstances.

6.15 Select from Model feature selection with Decision Tree

The Decision Tree model achieves 99.77% accuracy by using Select from Model feature selection. This emphasizes how useful it is to use the model's insights to choose features.

6.16 Select from Model feature selection with KNN

The KNN model's very good accuracy of 99.46% is achieved when paired with Select from Model. This illustrates the value of applying model-based insights for KNN feature prioritization.

6.17 Discussion

Our experiments have provided insightful information about how to enhance intrusion detection systems (IDS); yet, a number of issues need more research to determine potential areas for improvement.

1. **Feature Selection Discussion**
   Variance Threshold: Generally shown to be successful; nevertheless, a closer look at the specific maintained features may yield further information.
   SelectKBest: High accuracy, especially with Random Forest; interpretability could be improved by understanding how accuracy and selected attributes are related.
   Impact of ANOVA: SelectKBest's application of ANOVA yielded outstanding results; additional study into the statistically significant components identified by ANOVA could increase our understanding.
   Select from Model: Using model insights for feature selection proved useful when using Random Forest in particular; focusing on the most crucial features may aid in manual selection.
2. **Model and Evaluation Discussion**
   Random Forest: Consistently outperformed other models, demonstrating adaptability. Analyzing the group's decision-making process can shed light on its continued performance.
   KNN combined with logistic regression: Exhibited adaptability to different feature selection techniques. Future strategies may benefit from a closer look at the attributes that these models perform well with.
   Decision Tree Variability: Accuracy variability was demonstrated, highlighting the need for more meticulous feature selection. Gaining a deeper comprehension of choice boundaries may yield fresh insights.
3. **Further Recommendation**
   Detailed Feature Exploration: Examine the features in more detail to have a better understanding of the features that are kept by different selection processes.

Particular to a model Investigation: Conduct a thorough analysis of decision-making processes, paying particular attention to Random Forest and Decision Tree models.

7 Conclusion and Future Work

Review of the Research Question and Its Objectives:

Our study looked for approaches to use Machine Learning (ML) to increase model efficiency and decrease training times for Intrusion Detection Systems (IDS). To optimize IDS performance, we used multiple classifiers and feature selection techniques. The study started a comprehensive inquiry that covered techniques for selecting features, building models, preparing data, and visualizing the results.

Key Findings: The initiative achieved its objectives with great success. The capacity of Random Forest to continuously outperform other models, the nuanced effects of feature selection strategies on different classifiers, and the utility of the Select from Model method—particularly when combined with Random Forest—are some of the key findings. Numerous feature selection methods, including SelectKBest, Variance Threshold, ANOVA, and Select from Model, each had distinct benefits and provided valuable insights into their applicability.

Efficacy and Consequences: The implications of our investigation go beyond its immediate conclusions. IDS can achieve higher accuracy and efficiency when the efficaciousness of machine learning models is paired with meticulous feature selection. The fact that models like Random Forest are easily adaptable to complicated decision restrictions and that the observed performance regularly agrees with prior research lends additional credence to the robustness of our approach.

Limitations: Acknowledging one's limitations is essential to having a complete understanding. The focus of the study on accuracy metrics made it possible to conduct a more thorough analysis that considered precision, recall, and F1-score. Moreover, despite Random Forest's remarkable performance, further study should be done on the intricate decision-making process that takes place within its group.

Future work: Future studies could delve into detailed feature analysis, highlighting the significance of certain properties retained by different selection techniques. Improved interpretability can be achieved by looking at a model's specific decision-making processes, especially for Random Forest and Decision Tree models. Combined with other evaluation criteria, precision, recall, and F1-score will yield a more comprehensive assessment of performance.

To sum up, our research successfully navigated the complex field of intrusion detection system optimization. The outcomes demonstrate the efficacy of machine learning models, with Random Forest demonstrating particularly strong performance. While we are happy about this accomplishment, I also humbly acknowledge the limitations of our research and suggest future directions. The ML/IDS interface has immense promise for enhancing cybersecurity measures, and our research significantly advances this emerging subject.

References

1. Robiah, Nazrulazhar, Salama – “Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset.”

2. Sridevi Subbiah, Kalaisrasi Sonai, Saranya Thangaraj – “Intrusion Detection Technique in Wireless Sensor Network Using Grid Search Forest with Boruta Feature Selection Algorithm.”

3. Ibrahim Hayatu Hassan, Mohammed Abdullahi, Mansur Masama Aliyu – “An Improved Binary Manta Ray Foraging Optimization Algorithm Based Feature Selection and Forest Classifier for Network Intrusion Detection.”

4. Li Yang, Abdallah Shami – “IDS-ML: An Open Source Code for Intrusion Detection System Development using Machine Learning.”

5. HJ Liao, CHR Lin, YC Lin, KY Tung – “Intrusion Detection System: A Comprehensive Review.”

6. Anish Halimaa A, K. Sundarakantham – “Machine Learning Based Intrusion Detection System.”

7. R Panigrahi, S Borah – “A Detailed Analysis of CICIDS2017 Dataset for Designing Intrusion Detection System.”
8. A Novel Statistical Technique for Intrusion Detection System.
9. Enamul Kabir, Jiankun Hu, Hua Wang, Guangping Zhuo - "A novel statistical technique for intrusion detection systems"
10. R. Vinayakumar, Mamoun Alazab, K. P. Soman, Prabaharan Poornachandran - "A Deep Learning Approach to Network Intrusion Detection"
11. Preeti Mishra, Vijay Varadharajan, Uday Tupakula, Emmanuel S. Pilli - "A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection"
12. Sunanda Gamage, Jagath Samarabandu - "Deep learning methods in network intrusion detection: A survey and an objective comparison"
13. R Vinaykumar, M Alazab, KP Sonam – “Deep learning approach for intelligent intrusion detection system”
14. Gozde Karatas, Onder Demir, Ozgur Koray Sahingoz - "Deep Learning in Intrusion Detection Systems"
15. M Adullah, A Alshannaq, A Balamash, S Almabdy – “Enhanced Intrusion Detection System using Feature Selection Method and Ensemble Learning Algorithms.”
16. J Cai, J Luo, S Wang, S Yang – “Feature Selection in Machine Learning: A New Perspective.”
17. S Subbiah, KSM Anbananthen, S Thangaraj, S Kannan, D Chelliah – “Intrusion Detection Technique in Wireless Sensor Network using Grid Search Random Forest with Boruta Feature Selection Algorithm.”
18. M Di Mauro, G Galatro, G Fortino, A Liotta – “ Supervised Feature Selection Techniques in Network Intrusion Detection: A Critical Review.”
19. A Khraist, I Gondal, P Vamplew, J Kamruzzaman – “Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges.”