STUDENT INDUSTRIAL PROJECT REPORT

TITLE: NETWORK ANOMALY DETECTION WITH MACHINE LEARNING

OCT 2023 – DEC 2023

STUDENT NAME	**SAIRAM BALAMURUGAN**
STUDENT ID	**RAP X7406642**
PROGRAMME	**RESEARCH ATTACHMENT PROGRAM**
SUPERVISOR	**Dr. AZRINA ABD AZIZ**

ABSTRACT

The exponential growth of internet users and interconnected institutions has fostered a thriving online landscape. However, this burgeoning connectivity also presents an attractive target for cybercriminals, with the number of cyberattacks escalating rapidly. While traditional signature-based detection methods offer a degree of protection, they remain ineffective against novel "zero-day" attacks unknown to existing security systems. To address this critical gap, anomaly-based detection approaches offer a promising alternative, capable of identifying even the most nascent attacks.

This study delves into the application of machine learning for network anomaly detection. Leveraging the CICIDS2017 dataset, renowned for its up-to-date nature and diverse attack landscape, the research explores effective feature selection techniques. Employing the Random Forest Regressor algorithm, the study identifies the most informative features that best capture anomalous network behavior.

Subsequently, the analysis incorporates seven distinct machine learning algorithms, each offering unique strengths and weaknesses. Remarkably, all algorithms achieve impressive performance, exceeding 85% success rates in identifying network anomalies. Notably, K Nearest Neighbors emerges as the top performer with an exceptional 97% success rate, followed closely by ID3 at 95%.

Keywords: Machine Learning, Network Security, Network Anomaly Detection, CICIDS2017, Naive Bayes, QDA, Random Forest, ID3, AdaBoost, MLP, KNN

TABLE OF CONTENTS

1)INTRODUCTION:

1.1 MOTIVATION:

The exponential growth of internet users, exceeding 4 billion today, has driven an unprecedented surge in online communication. This digital landscape, however, is constantly under threat from cyberattacks, necessitating robust security solutions.

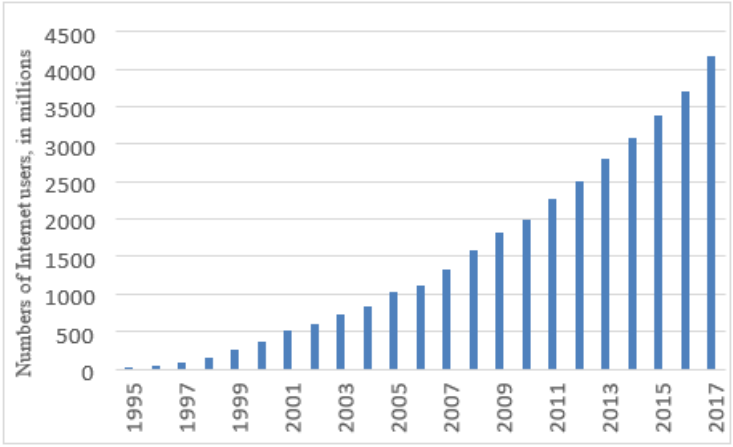


Figure 1 Increase in The Number of the Internet Users According to Years

Existing detection methods primarily rely on two approaches: signature-based and anomaly-based. While signature-based systems excel at identifying known attacks, they struggle with previously unseen threats (zero-day attacks) and are rendered ineffective against encrypted traffic, which now constitutes over half of all internet activity. Anomaly-based detection, however, offers a compelling alternative. By analyzing network flow characteristics such as packet size, connection time, and volume, it can identify anomalous behavior indicative of potential attacks, even when the data is encrypted. This makes it particularly effective against zero-day attacks, where the lack of pre-existing signatures renders traditional methods helpless. Furthermore, the ongoing rise of encrypted communication underscores the limitations of signature-based detection. As more information becomes encrypted, signature databases become increasingly inadequate, leaving networks vulnerable. Anomaly detection, however, bypasses this vulnerability by focusing on the inherent properties of network traffic, enabling it to analyze both encrypted and unencrypted data.

Recognizing the significant advantages of anomaly-based detection, this study aims to contribute to the field by developing a system that leverages machine learning techniques to achieve rapid and effective network anomaly detection. This innovative approach promises to enhance cybersecurity by

safeguarding networks against both known threats and emerging, zero-day attacks, ensuring a more secure online environment.

1.2 GOALS AND OBJECTIVE:

1.2.1 GOALS:

The goals that are aimed to achieve at the end of this study are as follows

1. **Explore and evaluate machine learning algorithms for network anomaly detection.** This objective delves into the capabilities and limitations of various algorithms in identifying anomalous network behavior.

2. **Develop a machine learning-based system for fast and effective detection of network attacks.** This objective aims to leverage the power of machine learning to achieve real-time identification of malicious activity within network traffic.

3. **Benchmark the performance of the developed system against existing research.** This objective involves a thorough comparison of the system's effectiveness with established approaches, allowing for a critical assessment of its contributions.

4. **Contribute to the field of network anomaly detection by achieving competitive results.** This objective aims to advance the knowledge base and state-of-the-art in anomaly detection through the development of a highly performant system.

1.2.2 OBJECTIVES:

The objectives that are aimed to achieve at the end of this study are as follows

1. Conduct a comprehensive literature review to identify existing research gaps and potential avenues for innovation in network anomaly detection.
2. Evaluate and select a suitable dataset that adequately represents real-world network traffic and supports the chosen machine learning algorithms.

3. Perform a thorough investigation and comparison of various machine learning algorithms for network anomaly detection, considering factors such as accuracy, efficiency, and computational cost.
4. Optimize and finalize the selection of machine learning algorithms through rigorous experimentation and analysis.
5. Choose an appropriate software platform that provides the necessary functionalities and resources for implementing and deploying the anomaly detection system.
6. Identify and configure suitable hardware/equipment that meets the performance and scalability requirements of the system.
7. Establish clear evaluation criteria that accurately assess the effectiveness of the developed system in detecting network anomalies.
8. Select relevant benchmark studies for performance comparison, enabling a comprehensive evaluation of the system's capabilities against existing approaches.

2) BACKGROUND AND RELATED WORK:

2.1 DATASETS:

In the realm of network anomaly detection using machine learning, accessing vast datasets of both malicious and benign network traffic is crucial for robust training and testing procedures. However, due to privacy concerns, directly utilizing real-world network traffic publicly is often infeasible. This necessitates the creation and ongoing development of dedicated datasets specifically tailored for anomaly detection research. This section delves into several prominent datasets, comparing and evaluating their suitability for the present research.

2.1.1 DARPA 98:

Developed by MIT Lincoln Laboratory with DARPA funding, this dataset aims to provide a training and testing environment for intrusion detection systems (IDS). It simulates a local computer network of the United States Air Force, capturing network traffic encompassing various

activities like file transfer, internet browsing, and email communication. Alongside benign traffic, it features 38 diverse attack types categorized as Denial-of-Service (DoS), User to Remote (U2R), Probe, and Remote to Local (R2L).

Despite its initial popularity, DARPA 98 has faced criticism for several shortcomings. Notably, it has been deemed outdated and fails to accurately reflect real-world network traffic patterns. Additionally, it lacks false positives, potentially impacting the effectiveness of machine learning algorithms trained on the dataset. Nevertheless, DARPA 98 holds significant historical value as it served as the foundation for widely utilized datasets like KDD Cup 99 and NSL-KDD.

2.1.2 KDD 99:

Created by the University of California, Irvine, this dataset was specifically designed for The Third International Knowledge Discovery and Data Mining Tools Competition (The KDD Cup '99). It leverages the data packets from DARPA 98, transforming them into 21 features via a feature extraction process optimized for machine learning algorithms. Divided into training and testing sets, it comprises 4,898,431 and 3,110,29 data streams respectively, encompassing 38 attack types, 14 of which are exclusive to the test set, representing unknown attacks. This allows for evaluating the system's ability to detect previously unseen threats.

Compared to DARPA 98, KDD 99 offers a more robust platform for machine learning applications with its new feature system and organized training and test sets. Consequently, it has become the preferred choice in numerous studies

2.1.3 CAIDA:

The Centre of Applied Internet Data Analysis (CAIDA) provides another valuable dataset for network anomaly research. Compiled from a few hours of OC48 backbone connection data recorded in San Jose, it also incorporates a simulated hourly DDoS attack. However, the dataset suffers from limitations in diversity, as it primarily focuses on specific applications and attack types. Additionally, the absence of labels in the data streams makes it challenging for machine learning applications.

2.1.4 NSL-KDD:

Despite KDD 99's advantages over DARPA 98, researchers observed significant redundancies within the dataset, potentially skewing results and hindering the performance of machine learning algorithms. Additionally, its large size prompted attempts to utilize subsets, but random selection often failed to capture the full spectrum of data characteristics.

To address these issues, Tavallaee et al. (2009) created the NSL-KDD dataset. This revised version eliminates redundancies and errors found in KDD 99. It comprises four parts under two main categories: training and testing. The training data includes KDDTrain+, a 20% subset of KDDTrain+ (KDDTrain+_20Percent), and the test data includes KDDTest+ and a smaller version with various difficulty levels (KDDTest-21).

2.1.5 ISCX 2012:

Concerns regarding outdatedness and the synthetic nature of existing datasets led to the development of ISCX 2012 (Intrusion Detection Evaluation Dataset) in 2012. Utilizing a seven-day real-world internet stream captured on a testbed built by the Canadian Institute for Cybersecurity, this dataset offers several key advantages:

- Realism: Employing real devices, it accurately captures both normal and malicious network traffic encompassing various protocols like FTP, HTTP, IMAP, POP3, SMTP, and SSH.
- Comprehensive Labeling: All data streams are meticulously labeled, facilitating efficient analysis and model training.
- Extensive Attack Variety: The dataset encompasses a wide range of attack types, including infiltration, denial-of-service, distributed denial-of-service, and brute-force SSH attacks.

2.1.6 CICIDS 2017:

The final dataset under consideration is CICIDS 2017 (Intrusion Detection Evaluation Dataset) developed by the Canadian Institute for Cybersecurity at the University of New Brunswick. This dataset captures a five-day network traffic stream (3rd July - 7th July 2017) from a

network comprised of real computers running contemporary operating systems like Windows Vista/7/8.1/10, Mac, Ubuntu 12/16, and Kali.

Flow Recording Day (Working Hours)	**pcap File size**	**Duration**	**CSV File Size**	**Attack Name**	**Flow Count**
Monday	10 GB	All Day	257 MB	No Attack	529918
Tuesday	10 GB	All Day	166 MB	FTP-Patator, SSH-Patator	445909
Wednesday	12 GB	All Day	272 MB	DoS Hulk, DoS GoldenEye, DoS slowloris, DoS Slowhttptest, Heartbleed	692703
Thursday	7.7GB	Morning	87.7 MB	Web Attacks (Brute Force, XSS, Sql Injection)	170366
		Afternoon	103 MB	Infiltration	288602
Friday	8.2GB	Morning	71.8 MB	Bot	192033
		Afternoon	92.7 MB	DDoS	225745
		Afternoon	97.1 MB	PortScan	286467

Table 2 Details of CICIDS 2017 Dataset

Several advantages distinguish CICIDS 2017 from previously mentioned datasets:

Real-world Data: Obtained from a testbed of actual computers, the data provides an accurate representation of real-world network traffic compared to simulated datasets.

Modern Operating Systems: The data reflects contemporary network behavior by capturing traffic from computers equipped with current operating systems. Additionally, it features operating system diversity across both attacker and victim machines.

Labeled Data: Each data stream is meticulously labeled, facilitating efficient analysis and machine learning model training. Furthermore, the pre-applied feature extraction process simplifies data preparation, resulting in 85 readily available features (listed in Appendix A).

Data Availability: Researchers have the flexibility to choose between working with raw data (pcap files) or processed data (CSV files).

Comprehensive Attack Coverage: The dataset incorporates a broad spectrum of attacks, leveraging the 2016 McAfee Security Report to ensure its relevance and comprehensiveness.

Protocol Diversity: CICIDS 2017 boasts a wider range of protocols compared to other datasets, including the increasingly prevalent HTTPS protocol alongside FTP, HTTP, SSH, and email protocols.

However, CICIDS 2017 also presents some limitations:

Large File Size: Both raw and processed data files are substantial in size, requiring significant storage capacity (47.9 GB and 1,147.3 MB respectively).

User-Defined Training and Testing Sets: Unlike KDD99 and NSL-KDD datasets, CICIDS 2017 does not offer pre-defined training and testing data files. Users must manually create these partitions, as explained in the "Creation of Training and Test Data" section.

Limited Literature and Potential Errors: Due to its recent development, CICIDS 2017 has not undergone extensive research and may harbor undiscovered minor errors. The "Data Cleansing" section addresses these potential issues and proposes solutions.

Selection Justification and Potential Contribution:

Following a thorough comparative analysis, the research chooses CICIDS 2017 processed data (CSV files) as the dataset for the implementation phase. This decision primarily stems from the dataset's key advantages:

- **Contemporary Relevance:** Its current network traffic and attack coverage ensure practical applicability to modern security challenges.
- **Broad Protocol and Attack Landscape:** The diverse range of protocols and attacks allows for a more comprehensive evaluation of the proposed anomaly detection system.
- **Limited Research:** Employing a relatively new dataset offers the potential for a significant contribution to the field by exploring its capabilities and addressing any existing limitations.

2.2 ANOMALIES AND ATTACK TYPES:

2.2.1 ANOMALY:

The concept of an anomaly refers to a data sample that deviates significantly from the well-defined characteristics of a normal sample. To effectively analyze an anomaly, it's crucial to have clear and valid rules defining the normal behavior. Network anomalies can be categorized into three primary types:

Point Anomaly: This type characterizes a single data sample that appears drastically different from the rest of the dataset due to its unique properties. For example, a credit card user who typically makes small purchases experiencing a sudden, significant increase in spending would be considered a point anomaly.

Contextual Anomaly: In this case, the abnormal behavior of a data sample is dependent on specific circumstances or conditions. For instance, the same credit card user experiencing the aforementioned sudden increase in spending would be classified as a contextual anomaly if the high expenditure occurred on a holiday, such as Valentine's Day, where larger purchases are expected.

Collective Anomaly: This type indicates a group of similar data points that collectively exhibit abnormal behavior compared to normal data. An example would be a significant rise in credit card spending across multiple users on Valentine's Day, which deviates from the usual spending patterns.

2.2.2 NETWORK ATTACK TYPES:

Network security focuses on protecting networks from attacks that attempt to violate three fundamental principles: confidentiality, integrity, and availability:

- **Confidentiality:** Protecting information from unauthorized access, ensuring its visibility only to legitimate users.
- **Integrity:** Guaranteeing that information remains unaltered, preventing unauthorized modifications or deletions.
- **Availability:** Ensuring that legitimate users have consistent access to the system and its resources.

Network attacks can be broadly classified into four main categories based on their attempts to compromise these principles:

Denial-of-Service (DoS) Attacks:

These attacks aim to disrupt legitimate user access to services by overwhelming the system with excessive resources. A simple DoS attack involves sending a barrage of requests to a web server, causing it to become overloaded and unavailable. DoS attacks can be further categorized into two types:

- Bandwidth Depletion: Aims to saturate the victim's bandwidth by flooding it with massive data flows.
- Resource Depletion: Targets the victim's resources like memory and CPU by sending a large volume of packets, rendering the system unresponsive.

Probe (Information Gathering) Attacks:

These attacks focus on acquiring information about the targeted network. Attackers can gain valuable insights such as network structure, operating systems used, and device types. While seemingly harmless, probe attacks pave the way for more damaging attacks by providing attackers with critical intelligence.

User to Root (U2R) Attacks:

U2R attacks aim to gain control of an administrator account, enabling access and theft of sensitive resources. Attackers often exploit system vulnerabilities or utilize brute-force attacks to achieve their objective.

Remote to User/Local (R2U/R2L) Attacks:

In these attacks, the attacker infiltrates the victim's network to gain the privilege of sending packets through the victim's computer. Similar to U2R attacks, they typically exploit system vulnerabilities or employ brute-force tactics.

2.2.3. ANOMALY AND ATTACKS – RELATIONSHIP:

Understanding the relationship between network anomalies and attacks can be instrumental in detecting malicious activity. Each attack type manifests itself as a distinct anomaly within network traffic patterns. For example, DoS attacks typically manifest as an

increase in data volume and packet count. These abnormal traffic surges often indicate a collective anomaly.

On the other hand, U2R and R2U attacks are more accurately classified as contextual and point anomalies. This is because they target specific users, ports, and purposes within the network, leading to targeted anomalies rather than widespread disruptions. Similarly, probe attacks, while generating dense packet traffic, also have specific objectives, making them suitable candidates for classification as both contextual and collective anomalies.

Figure 2 effectively summarizes the relationship between these various network anomalies and attack types, providing a valuable visual reference for understanding the connections between them.

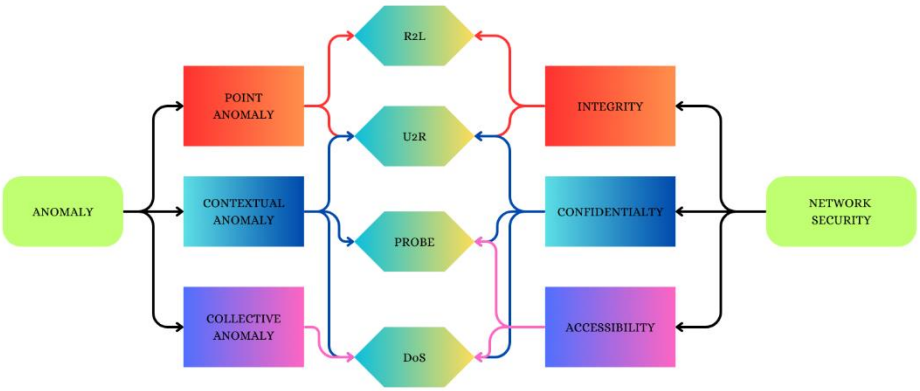


Figure 2 Relationship between Anomaly and Attack

By analyzing network anomalies and understanding their connection to specific attack types, we gain a powerful tool for detecting and mitigating threats to network security. This knowledge allows us to develop effective anomaly detection systems that can identify and respond to these attacks proactively, safeguarding critical information and ensuring the continued operation of vital networks.

2.3 ATTACKS:

The CICIDS2017 dataset provides a valuable window into the diverse and ever-evolving world of cyberattacks. This dataset, encompassing a wide range of attack types, serves as a springboard for our investigation.

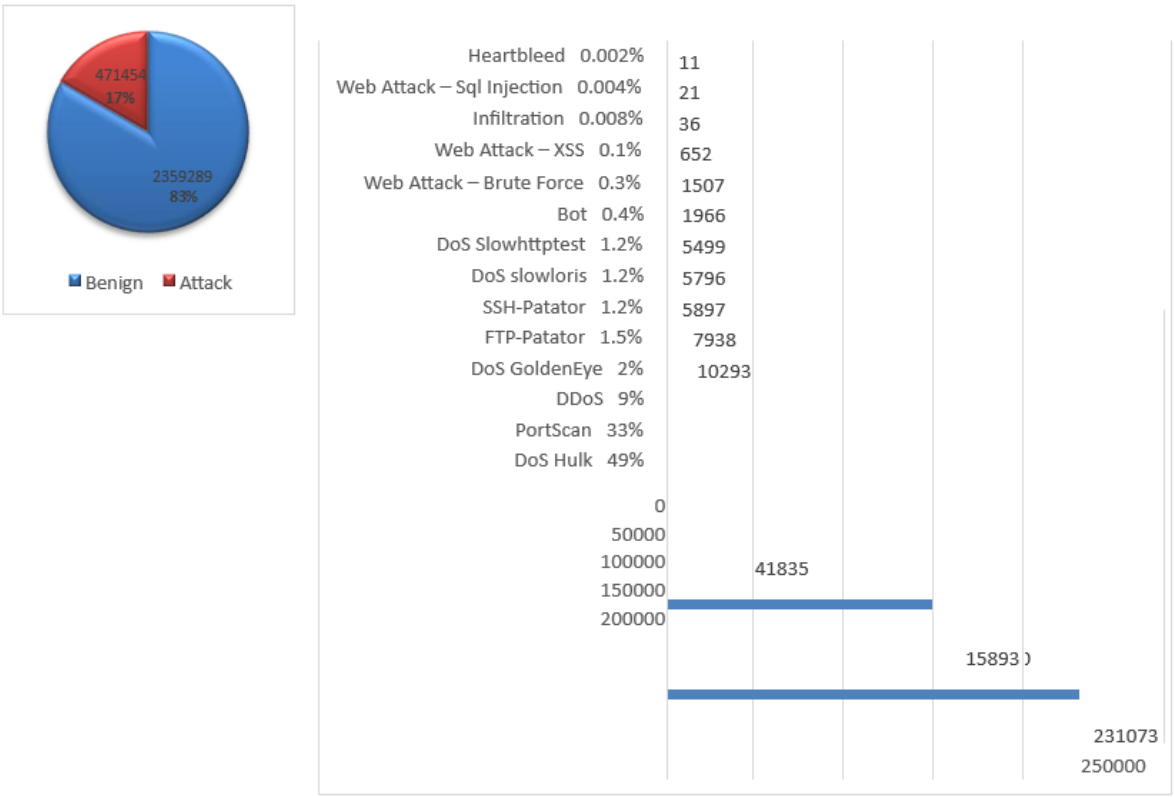


Figure 3 Distribution of Attack types in Dataset

2.3.1 DoS Attacks: A Forceful Disruption

DoS (Denial-of-Service) attacks lead the pack, constituting a significant portion of the dataset. These attacks aim to overwhelm a system with excessive traffic, rendering it inaccessible to legitimate users.

- DoS HULK: This HTTP-based attack leverages TCP-SYN floods and multiple HTTP-GET requests to cripple server resources. It can remain concealed through user agent manipulation and varying templates.
- DoS Goldeneye: This multithreaded Python-based attack exploits the HTTP Keep-Alive feature and disables HTTP caching to maximize resource consumption.
- DoS Slowloris: Using incomplete TCP packets with SYN flags, this attack aims to consume server resources by keeping connections open for extended periods.
- DoS SlowHTTPTest: This attack abuses the TCP window size feature, sending normal requests but setting the window size near zero to significantly slow down the receiving process.

- Botnet: This attack utilizes a network of infected computers (bots) to launch various attacks, including DDoS and SPAM. Identifying bots is possible only during active attack phases.

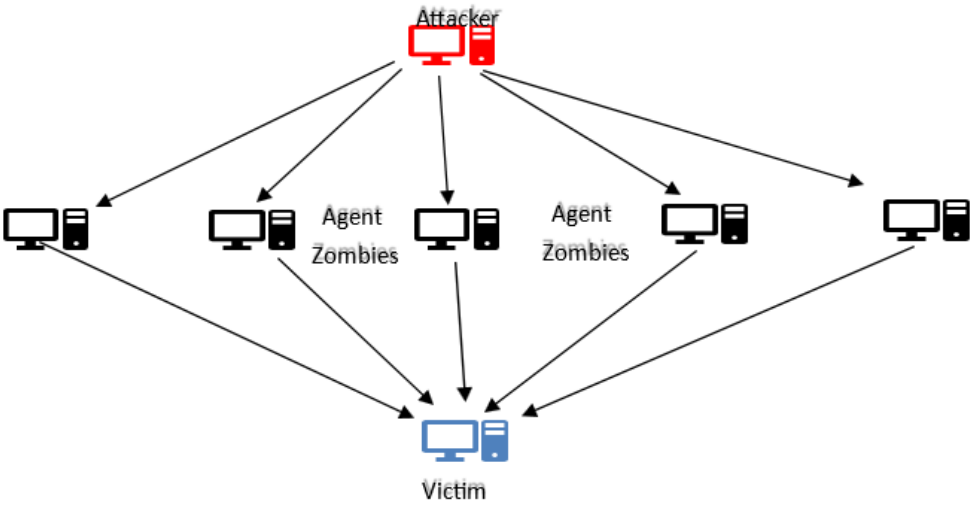


Figure 4 Distributed Denial of Service

2.3.2 Brute-Force Attacks: Cracking the Code

Brute-force attacks attempt to gain unauthorized access by systematically trying various username and password combinations.

- FTP-Patator: This multithreaded attack targets FTP servers, employing a dictionary-based approach to crack usernames and passwords.
- SSH-Patator: This three-phased attack begins with scanning, followed by a brute-force attempt and finally gaining full control.

2.3.3 Web Attacks: Exploiting the Web's Vulnerabilities

Web attacks target web applications, often exploiting vulnerabilities to steal data or gain unauthorized access.

- Web Attack - Brute Force: This attack attempts to log in to a web application using various username and password combinations.

- Web Attack - XSS: This attack injects malicious code into a web page, allowing the attacker to steal data, hijack sessions, or execute unauthorized code.
- Web Attack - SQL Injection: This attack exploits vulnerabilities in web applications to inject malicious SQL commands, potentially enabling data theft, modification, or deletion.

2.3.4 Specialized Attacks: Tailored for Specific Scenarios

- Infiltration: This scenario-specific attack targets a network compromised by a virus, exploiting its vulnerabilities to launch port scans.
- Heartbleed: This attack exploits a vulnerability in OpenSSL libraries, enabling the attacker to steal confidential information such as usernames and passwords.

2.4 MACHINE LEARNING:

The vast array of machine learning algorithms can be categorized into four distinct groups based on their training data and supervision:

1. Supervised Learning:

In supervised learning, the training data is meticulously labeled with the desired outcome. This allows the algorithm to learn the association between the input data and the corresponding output, enabling it to make accurate predictions on new, unseen data. Popular examples of supervised learning algorithms include decision trees, k-nearest neighbors, and support vector machines.

2. Unsupervised Learning:

Unsupervised learning tackles the challenge of unlabeled data. Here, the algorithm must identify inherent patterns and relationships within the data without any prior guidance. This makes it ideal for tasks like anomaly detection, clustering, and dimensionality reduction. Common unsupervised learning algorithms include k-means clustering, principal component analysis (PCA), and autoencoders.

3. Semi-supervised Learning:

Semi-supervised learning bridges the gap between supervised and unsupervised learning by employing a small set of labeled data along with a large amount of unlabeled data. This approach leverages the power of supervised learning while mitigating the laborious task of labeling large datasets.

4. Reinforcement Learning:

Reinforcement learning takes a more interactive approach. Here, the algorithm learns through trial and error in an environment, receiving rewards for desired actions and penalties for undesirable ones. This allows the algorithm to gradually optimize its behavior towards achieving a specific goal. Reinforcement learning has seen significant success in areas like robotics, game playing, and resource allocation.

This project prioritizes supervised learning algorithms to leverage the advantage of readily available, well-labelled data. This approach aims to achieve high performance without incurring the typical costs associated with data labelling.

To accomplish this, the project employs a diverse set of popular algorithms: Naive Bayes, QDA, Random Forest, ID3, AdaBoost, MLP, and K Nearest Neighbours. Choosing such a wide range of algorithms allows for a comprehensive comparison of their characteristics and performance within the context of the project.

2.4.1 Naive Bayes:

Simple yet powerful, Naive Bayes relies on the assumption of conditional independence between features. This efficient algorithm excels in situations with large data sets but may struggle with complex relationships between features.

2.4.2 Decision Trees:

Structured like a branching tree, decision trees offer a clear and interpretable approach to classification. They efficiently navigate complex data sets by splitting them into smaller, more manageable segments based on specific criteria.

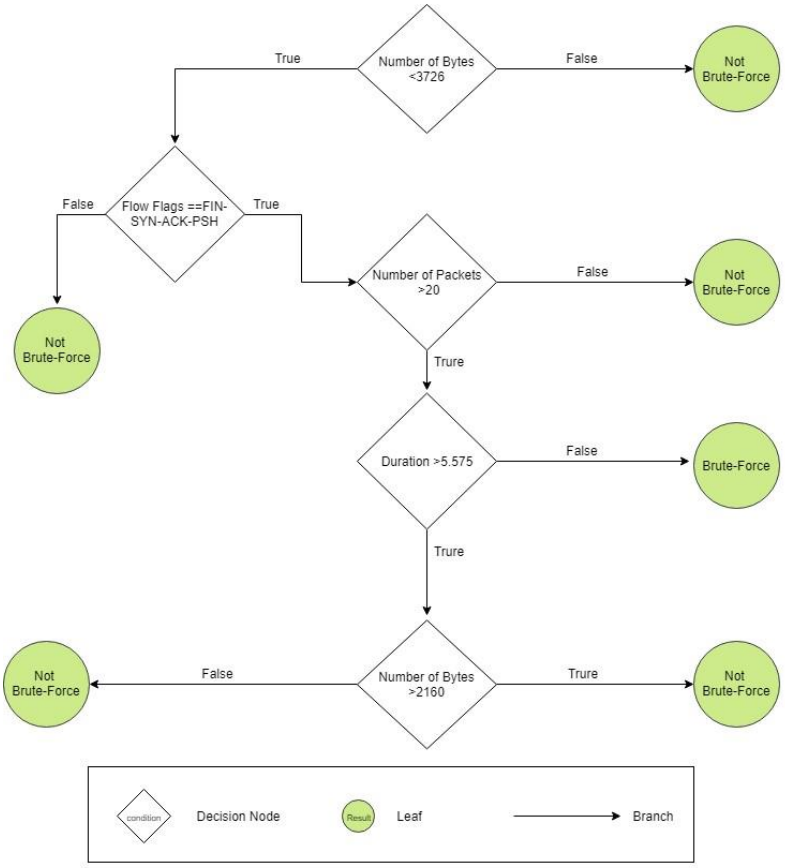


Figure 5 Detection of Brute Force using Decision tree

2.4.3 Random Forest:

Harnessing the collective wisdom of a multitude of decision trees, Random Forest boasts superior accuracy and robustness. This ensemble method mitigates the overfitting tendencies of individual trees, making it a potent weapon for tackling diverse ML challenges.

2.4.4. K-Nearest Neighbors:

This intuitive algorithm classifies new data points by analyzing their proximity to existing data points. By identifying the "k" nearest neighbors, KNN effectively navigates data sets with complex, non-linear relationships.

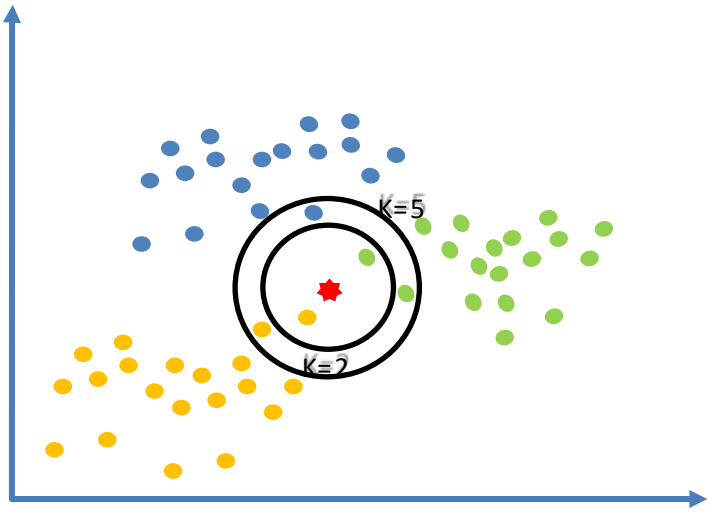


Figure 6 Operation of KNN Algorithm for K=2 and K=5

2.4.5 AdaBoost:

Transforming a group of weak learners into a powerful ensemble, AdaBoost leverages the collective intelligence of multiple algorithms. This adaptive approach iteratively refines its focus, boosting the performance of weak learners and achieving remarkable accuracy.

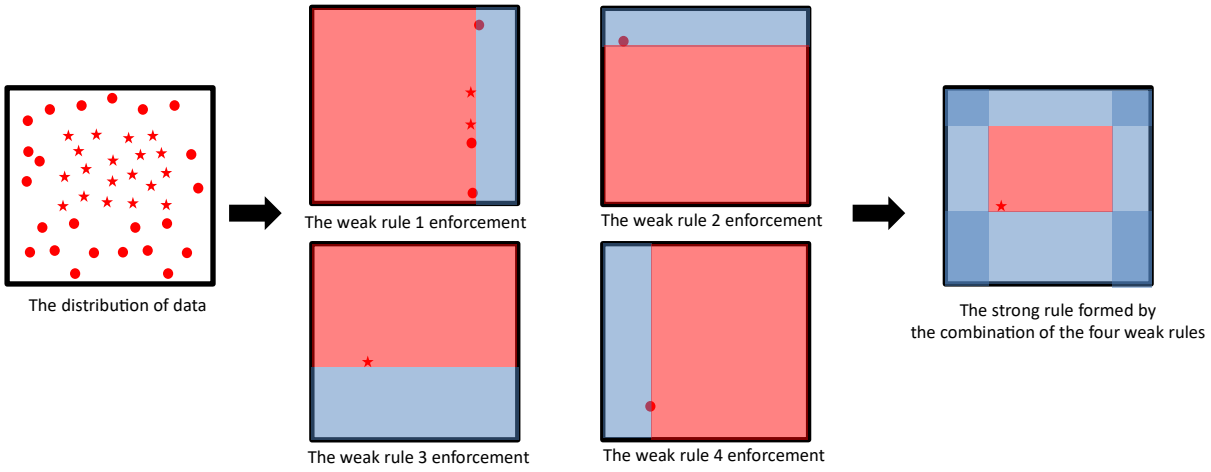


Figure 7 Demonstration of Operation of AdaBoost Algorithm

2.4.6 Multi-Layer Perceptron (MLP):

Inspired by the human brain, MLP employs an interconnected network of artificial neurons to process information. This versatile architecture excels at handling complex, non-linear relationships, making it a valuable tool for a diverse range of ML applications.

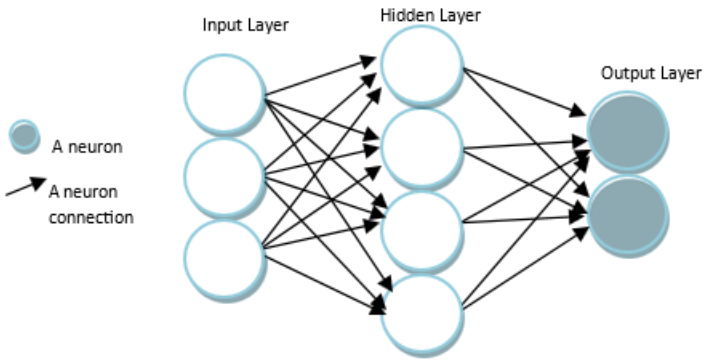


Figure 8 Three Layer NLP

2.4.7 Quadratic Discriminant Analysis (QDA):

For data sets with unequal group variances, QDA offers a superior alternative to linear discriminant analysis. This method accounts for the inherent differences between groups, achieving more accurate classification results.

3) METHODOLOGY:

This project leverages a carefully chosen software and hardware platform, along with established performance evaluation methods, to achieve optimal results.

3.1 SOFTWARE TOOLS:

3.1.1 Python:

This open-source, object-oriented programming language forms the bedrock of the project. Its simple syntax, dynamic structure, and extensive documentation make it ideal for both code development and analysis. Additionally, numerous libraries cater specifically to machine learning applications, solidifying its position as the chosen tool.

3.1.2 Scikit-learn:

This comprehensive machine learning library for Python offers a vast array of algorithms, encompassing all those required for this project. Its extensive documentation and user-friendly interface further enhance its appeal.

3.1.3 Pandas:

When dealing with large datasets, Pandas proves invaluable. This powerful data analysis library facilitates seamless operations like filtering, manipulating columns and rows, and performing calculations, making it an indispensable asset for the project.

3.1.4 Matplotlib:

Visualizing data plays a crucial role in understanding and interpreting results. Matplotlib, a Python library dedicated to data visualization, enables the creation of insightful graphs and charts that illuminate the project's findings.

3.1.5 NumPy:

Mathematical and logical operations are performed swiftly and efficiently with the help of NumPy. This Python library provides the computational power necessary for the project's calculations.

3.2 PERFORMANCE EVALUATION:

The project's results are evaluated based on four key criteria: accuracy, precision, F-measure, and recall. All of these metrics range from 0 to 1, with higher values indicating better performance.

- **Accuracy:** This metric reflects the proportion of correctly categorized data points relative to the total data.
- **Recall (Sensitivity):** This measure indicates the ratio of attack data accurately classified as attacks.
- **Precision:** This metric reveals the percentage of data classified as attacks that are truly attacks.
- **F-measure (F-score/F1-score):** This metric serves as the harmonic mean of sensitivity and precision, providing a single measure of overall performance. This score is particularly important for evaluating the project's effectiveness.

These four metrics are calculated using the following values:

- **TP:** True Positive (Correct Detection) – Attack data classified as attacks.
- **FP:** False Positive (Type-1 Error) – Benign data classified as attacks.

- **FN:** False Negative (Type-2 Error) – Attack data classified as benign.
- **TN:** True Negative (Correct Rejection) – Benign data classified as benign.

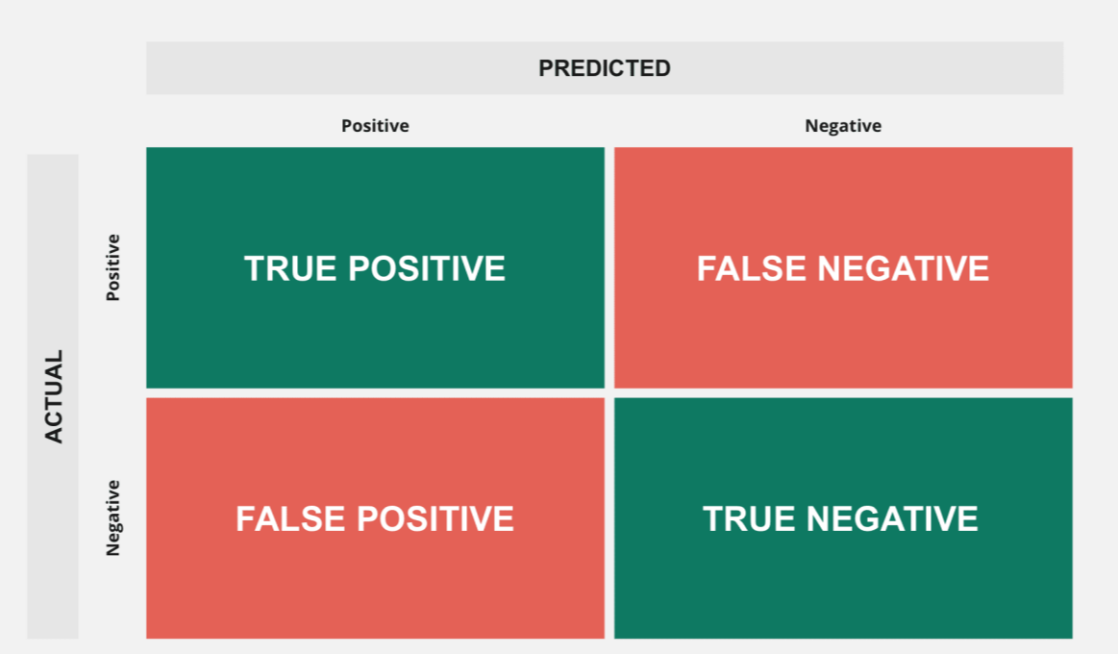


Figure 9 Confusion Matrix Structure

In addition to these four established metrics, processing time is also considered, recognizing its importance in selecting the optimal algorithm. While not strictly a success criterion, processing time plays a significant role in determining the project's efficiency and scalability.

By leveraging this carefully chosen software and hardware platform, coupled with robust performance evaluation methods, the project aims to achieve a comprehensive and objective assessment of the employed algorithms.

3.3 IMPLEMENTATION

This section delves into the intricate process of anomaly detection through machine learning techniques. A multi-pronged approach, encompassing data cleansing, training and test data creation, feature selection, and finally, the application of machine learning algorithms, paves the way for effective anomaly identification.

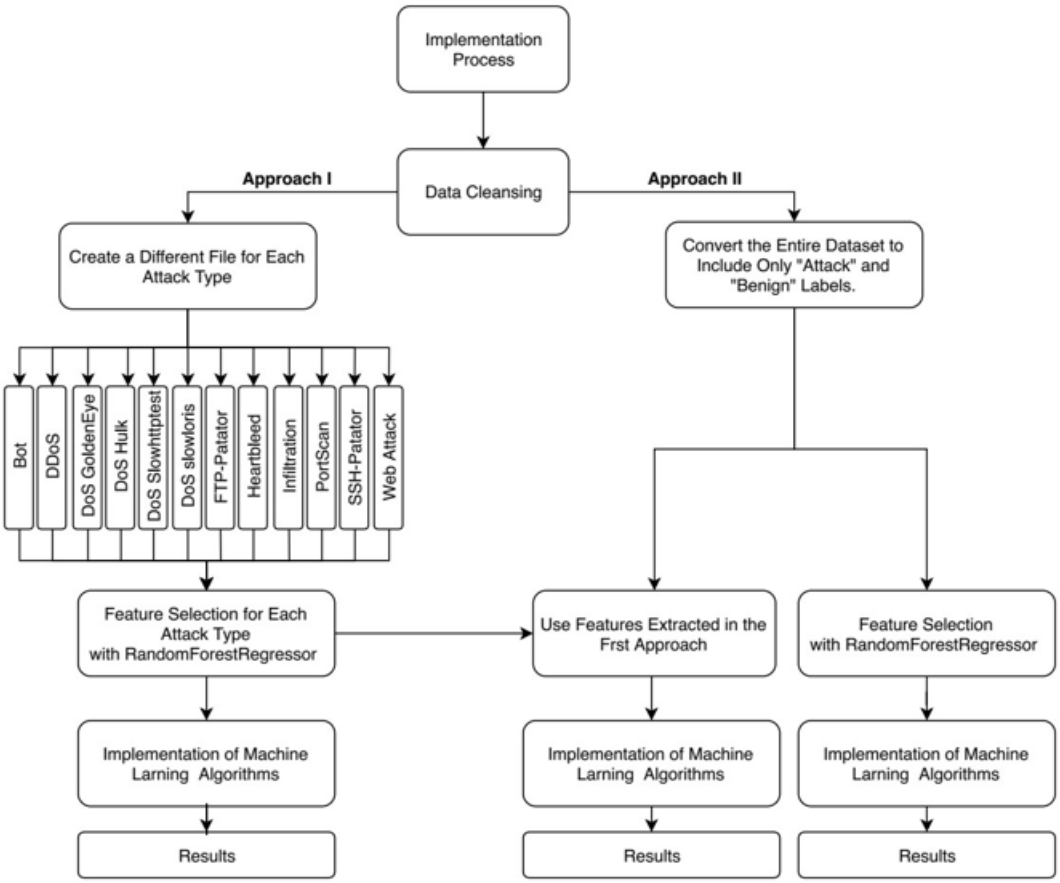


Figure 10 The implementation process

3.3.1 Data Cleansing: Purifying the Source

Prior to utilizing the CICIDS2017 dataset for practical purposes, certain adjustments are necessary to optimize its efficacy. This section delves into the identification and rectification of deficiencies within the dataset, paving the way for efficient and accurate analysis.

1. Identifying and Addressing Imperfections

Inspection of the dataset reveals 288,602 incomplete records, constituting a significant portion of the overall data. These incomplete records introduce inconsistencies and potentially hinder the effectiveness of subsequent analysis. Therefore, the first step in the pre-processing process involves meticulously removing these flawed records, ensuring a clean and reliable foundation for further exploration.

2. Distribution Analysis: Understanding the Data Landscape

Table 2 provides a comprehensive overview of the distribution of stream records within the dataset.

Label Name	Number
Benign	2359289
Faulty	288602
DoS Hulk	231073
PortScan	158930
DDoS	41835
DoS GoldenEye	10293
FTP-Patator	7938
SSH-Patator	5897
DoS slowloris	5796
DoS Slowhttptest	5499
Bot	1966
Web Attack – Brute Force	1507
Web Attack – XSS	652
Infiltration	36
Web Attack – SQL Injection	21
Heartbleed	11

Table 2 Distribution of Stream records in CICIDS 2017 Dataset

This analysis allows for a deeper understanding of the data landscape and facilitates informed decision-making regarding subsequent pre-processing steps.

3. Optimizing for Efficiency: Tailoring the Data for Analysis

Following the removal of incomplete records, further optimization of the dataset may be necessary depending on the specific analysis objectives. This could involve addressing additional inconsistencies, handling missing values, or transforming categorical data into numerical formats.

The CICIDS2017 dataset presents certain challenges in terms of feature representation, which necessitate adjustments to optimize its usability for machine learning algorithms.

1. Eliminating Redundancies: Streamlining Feature Space

Upon examination, a redundancy is identified within the dataset - the "Fwd Header Length" feature appears twice, occupying both the 41st and 62nd columns. This redundancy can potentially skew analysis and introduce inconsistencies. To address this issue, the duplicate feature in the 62nd column is meticulously removed, ensuring a streamlined and consistent feature space.

2. Encoding Categorical Data: Bridging the Gap for Machine Learning

Several features within the dataset, including "Flow ID," "Source IP," "Destination IP," "Timestamp," and "External IP," contain categorical and string values. These values are incompatible with machine learning algorithms, which require numerical input. To facilitate seamless integration with machine learning algorithms, the LabelEncoder from Sklearn is employed. This powerful tool transforms each unique string value into an integer code within the range 0 to n-1, rendering the data suitable for machine learning operations.

3. Preserving Categorical Context: Balancing Transformation with Interpretation

While other categorical features undergo numerical encoding, the "Label" tag remains untouched. This deliberate decision stems from the need to preserve the original categories during attack classification and analysis. By retaining the original categorical labels, different attack types can be readily identified and categorized, enabling diverse approaches and interpretations of the data.

Finally, some minor structural changes should be made to the dataset, including:

1. Addressing Character Encoding Discrepancies: Enhancing Compatibility

The "Label" feature utilizes the character "–" (Unicode Decimal Code &#8211) to identify web attack subtypes. However, this character is not recognized by utf-8, the default codec of the

Pandas library. To ensure compatibility and avoid errors, this character must be replaced with the standard hyphen "-" (Unicode Decimal Code &#45).

2. Converting Outliers for Seamless Integration with Machine Learning Algorithms

Both the "Flow Bytes/s" and "Flow Packets/s" features contain certain outliers, including "Infinity" and "NaN" values, alongside numerical data. To facilitate seamless integration with machine learning algorithms, which require consistent and interpretable data, these outlier values are strategically transformed. Specifically, "Infinity" values are replaced with -1, representing the lowest possible value, while "NaN" values are replaced with 0, representing the absence of data.

3.3.2 Creation of test and Train Dataset:

The efficacy of any machine learning algorithm hinges on the quality and structure of the training data utilized. The CICIDS2017 dataset, unfortunately, lacks dedicated training and testing partitions, necessitating its pre-processing to facilitate effective model development and evaluation.

1. Addressing the Absence of Predefined Partitions: Splitting the Data

To address this challenge, the Sklearn library's train_test_split function is employed. This powerful tool facilitates the division of the unpartitioned CICIDS2017 dataset into distinct training and testing subsets based on user-defined ratios.

2. Balancing Training and Testing Data: Optimizing Performance

Generally, a 80/20 ratio is preferred for partitioning data, allocating 80% of the data to the training set and the remaining 20% to the testing set. This ratio ensures the algorithm acquires sufficient knowledge through the training set while retaining a representative portion of data for robust performance evaluation.

3. Leveraging Cross-Validation for Robust Results

To ensure the reliability andgeneralizability of the results obtained, the training and testing data split is performed ten times in succession through an iterative process known as cross-validation. This rigorous approach ensures a comprehensive evaluation of the algorithm's performance across diverse data subsets, mitigating potential biases and inconsistencies.

4. Consolidating Results: Aggregating Insights

The performance metrics obtained from each iteration of the cross-validation process are then averaged, yielding a comprehensive and statistically robust evaluation of the machine learning algorithm's effectiveness.

3.3.3 Feature Selection

This section delves into the crucial task of feature selection, aiming to identify the most informative features that effectively distinguish between different attack types. A comprehensive list of features and their descriptions can be found in Appendix A.

3.3.3.1 Feature Selection According to Attack Types

This section delves into the critical process of identifying the most informative features that contribute to accurate attack classification. To achieve this, distinct files are created for each attack type, isolating them from other attacks. These files contain both the attack streams and a randomly selected subset of benign streams, maintaining a 30/70 attack-to-benign ratio.

The powerful Random Forest Regressor algorithm from Sklearn is employed to calculate the importance weights of each feature. This algorithm constructs a decision forest, wherein each feature receives a weight based on its contribution to the structure of the decision tree. Upon completion, these importance weights are compared and sorted, revealing the most influential features in attack classification.

However, eight features (Flow ID, Source IP, Source Port, Destination IP, Destination Port, Protocol, Timestamp,

External IP) are deliberately excluded from this process. While traditionally utilized in classical approaches, these features can potentially be manipulated by attackers to avoid detection. For instance, attackers might intentionally select obscure ports to evade monitoring or utilize spoofed IP addresses to obfuscate their origin. Additionally, dynamic port allocation and multi-application sharing of the same port further limit the reliability of port-based analysis.

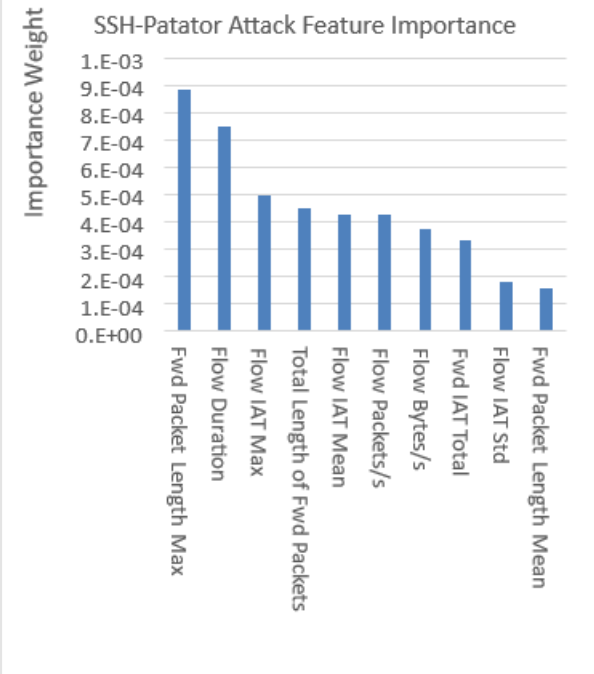
The distribution of features and four attributes with the most significance value for each attack can be seen from Table 3.

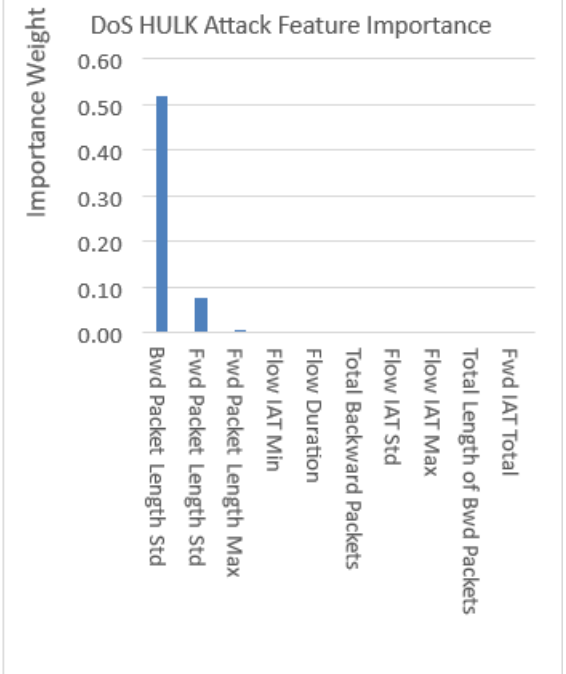
Attack / Feature Name	**Importance Weight**		**Attack / Feature Name**	**Importance Weight**
Bot			**FTP-Patator**	
Bwd Packet Length Mean	0.304823		Fwd Packet Length Max	0.063671
Flow IAT Max	0.034495		Fwd Packet Length Std	0.022751
Flow IAT Std	0.019464		Fwd Packet Length Mean	0.002179
Flow Duration	0.010129		Total Length of Bwd Packets	0.000746
DDoS			**Heartbleed**	
Bwd Packet Length Std	0.468089		Bwd Packet Length Mean	0.064
Total Backward Packets	0.094926		Total Length of Bwd Packets	0.056
Fwd IAT Total	0.012066		Flow IAT Min	0.056
Total Length of Fwd Packets	0.006438		Bwd Packet Length Std	0.044
DoS GoldenEye			**Infiltration**	
Flow IAT Max	0.442727		Total Length of Fwd Packets	0.05238
Bwd Packet Length Std	0.091185		Flow IAT Max	0.036096
Flow IAT Min	0.053795		Flow Duration	0.016453
Total Backward Packets	0.041583		Flow IAT Min	0.015448
DoS Hulk			**PortScan**	
Bwd Packet Length Std	0.514306		Flow Bytes/s	0.313402
Fwd Packet Length Std	0.069838		Total Length of Fwd Packets	0.304917
Fwd Packet Length Max	0.008542		Flow Duration	0.000485
Flow IAT Min	0.001716		Fwd Packet Length Max	0.00013
DoS Slowhttptest			**SSH-Patator**	
Flow IAT Mean	0.64206		Flow Bytes/s	0.000846
Fwd Packet Length Min	0.075942		Total Length of Fwd Packets	0.000814

Fwd Packet Length Std	0.022194		Fwd Packet Length Max	0.000749
Bwd Packet Length Mean	0.020857		Flow IAT Mean	0.000734
DoS slowloris			**Web Attack**	
Flow IAT Mean	0.465561		Total Length of Fwd Packets	0.014697
Bwd Packet Length Mean	0.075633		Bwd Packet Length Std	0.00536
Total Length of Bwd Packets	0.049808		Flow Bytes/s	0.00257
Total Fwd Packets	0.01868		Bwd Packet Length Max	0.001922

Table 3 The Distribution of features and and four attributes with the most significant value for each attack

Analysis of feature distributions reveals a compelling pattern: for most attack types, one or two features exhibit distinct prominence. In stark contrast, the Heartbleed and SSH-Patator attacks exhibit significantly different characteristics. Their feature distributions lack dominant features, instead displaying several features with closely clustered values. (For detailed visualizations and values, refer to Appendix B.)





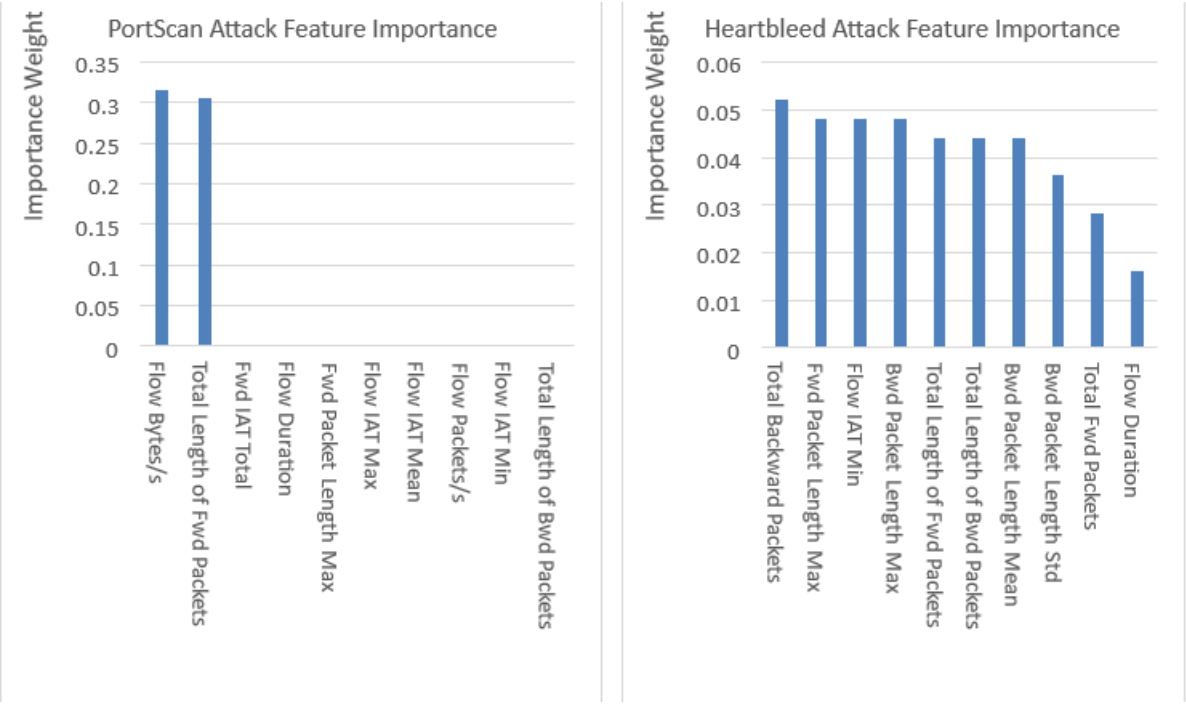


Figure 11 Graphs of feature importance of SSH-Patator, DoS HULK, PortScan, Heartbleed attacks

In the case of PortScan attacks, two features - "Flow Bytes/s" and "Total Length of Fwd Packets" - exhibit significant prominence. This pattern reflects the attacker's strategy of sending numerous packets with minimal payloads (typically around 40 bytes) to expedite the scanning process and optimize bandwidth usage. This approach maximizes efficiency and effectiveness, making PortScan readily identifiable through these dominant features.

Unlike other attacks, SSH-Patator lacks obvious distinguishing features. This can be attributed to its multi-phased nature, encompassing a complex chain of scanning, brute-force, and termination stages. As it blends characteristics of both PortScan and Brute-Force attacks, its feature distribution lacks a single dominant signature, resulting in closely clustered importance values across various features.

3.3.3.2 Feature selection according to attack or benign

An alternative approach to feature selection involves applying the Random Forest Regressor to the entire dataset, consolidating all attack types under a single "attack" label. This approach results in a data file containing only attack and benign streams. The feature list obtained through this

analysis is presented in Table 4, while Figure 11 provides visual representations of the features.

Feature Name	Priority Weight		Feature Name	Priority Weight
Bwd Packet Length Std	0.246627		Flow IAT Mean	0.003266
Flow Bytes/s	0.178777		Total Length of Bwd Packets	0.001305
Total Length of Fwd Packets	0.102417		Fwd Packet Length Min	0.000670
Fwd Packet Length Std	0.063889		Bwd Packet Length Mean	0.000582
Flow IAT Std	0.009898		Flow Packets/s	0.000541
Flow IAT Min	0.006946		Fwd Packet Length Mean	0.000526
Fwd IAT Total	0.005121		Total Backward Packets	0.000169
Flow Duration	0.004150		Total Fwd Packets	0.000138
Bwd Packet Length Max	0.004007		Fwd Packet Length Max	0.000125
Flow IAT Max	0.003579		Bwd Packet Length Min	0.000084

Table 4 The Priority weight for attack or benign approach

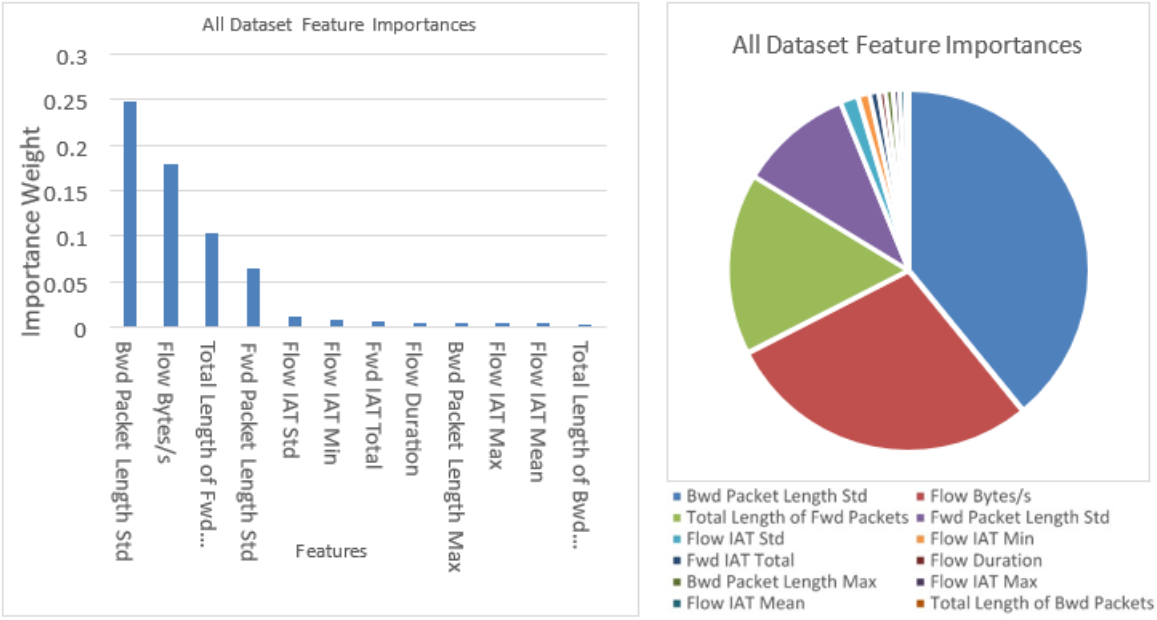


Figure 12 Graphs of priority weight according to attack or benign list

3.3.4 Implementation of Machine Learning Algorithm

In this section, we delve into the application of machine learning algorithms to the CICIDS2017 dataset, employing two distinct approaches to extract valuable insights:

1. Attack-Specific Analysis:

This approach focuses on individual attack types, leveraging the feature sets identified in the Feature Selection section. Each attack type is represented by a dedicated file containing 30% attack data and 70% benign data, facilitating targeted analysis. Subsequently, seven machine learning algorithms are applied to each of these files, repeated ten times for robust evaluation. This methodology allows for detailed observation of the effectiveness and performance of different algorithms against specific attack types, providing valuable insights into their strengths and weaknesses in diverse scenarios.

2. Holistic Evaluation:

In this approach, the entire dataset is utilized as a single unit, encompassing all attack types under a single "attack" label. This consolidated file solely contains attack and benign data, enabling a comprehensive analysis of the overall effectiveness of machine learning algorithms across the entire spectrum of attack types. To ensure effective feature selection, the four features with the highest importance-weight for each individual attack type, identified in the first approach, are combined into a single pool. This results in a preliminary pool of 48 candidate features, which is then refined by eliminating redundancies, leading to a final set of 18 features. This set of features captures the most critical indicators of anomalous behavior across all attacks, enabling the development of robust and generalizable machine learning models for anomaly detection.

Bwd Packet Length Max	Flow IAT Mean	Fwd Packet Length Min
Bwd Packet Length Mean	Flow IAT Min	Fwd Packet Length Std
Bwd Packet Length Std	Flow IAT Std	Total Backward Packets
Flow Bytes/s	Fwd IAT Total	Total Fwd Packets
Flow Duration	Fwd Packet Length Max	Total Length of Bwd Packets
Flow IAT Max	Fwd Packet Length Mean	Total Length of Fwd Packets

Table 5 Feature list created for all attack type

While the combination of attack-specific features offers valuable insights, a data-centric approach utilizing features with high importance scores across the entire dataset can also be implemented. This approach prioritizes efficient feature selection, focusing on maximizing information gain while minimizing dimensionality.

1. Identifying Feature Significance Threshold:

To achieve this objective, a threshold value of 0.8% is established for feature weight. This threshold ensures that the selected features collectively capture 97% of the total feature importance weight, effectively focusing on the most informative attributes that contribute significantly to attack identification.

2. Refined Feature Set:

By applying this threshold, seven features emerge as the most significant contributors to attack classification. These features capture the essence of anomalous behavior across all attack types, enabling the development of robust and generalizable machine learning models with minimal feature redundancy. The selection of these seven features prioritizes both effectiveness and efficiency, resulting in a concise yet informative representation of the data.

Feature Name	Importance Weight	Percentage
Bwd Packet Length Std	0.246627	38.97%
Flow Bytes/s	0.178777	28.25%
Total Length of Fwd Packets	0.102417	16.18%
Fwd Packet Length Std	0.063889	10.10%
Flow IAT Std	0.009898	1.56%
Flow IAT Min	0.006946	1.10%
Fwd IAT Total	0.005121	0.8 %

Table 6 the priority weight obtained in "feature selection in attack of benign"

4) RESULTS AND DISCUSSION

In this section, the results of the studies done in the implementation section are presented. In this context, in the assessment carried out, the evaluation criteria are presented via the data of the F-measure. However, all the evaluation data obtained can be accessed from the Appendix.

The performance evaluation procedures are repeated 10 times for each machine learning algorithm. The numbers given in the tables are the arithmetic mean of these 10 processes. Box and whisker graphs are created to illustrate the consistency of the results and the change between them.

4.1 Approach 1 – Using 12 attack types

This section presents the results of applying seven machine learning algorithms to twelve diverse attack types, with the findings summarized in Table 7. To highlight both the best and worst performers, outstanding scores are bolded while the least effective scores are both underlined and *italicized*. In cases where multiple algorithms achieve the same F-measure, additional metrics such as accuracy, precision, recall, and execution time are used to break the tie (see Appendix C for detailed results).

Attack Names	**F-Measures**						
	NB	RF	KNN	ID3	AB	MLP	QDA
Bot	*0.54*	0.96	0.95	0.96	**0.97**	0.64	0.68
DDoS	0.77	0.96	0.92	**0.96**	0.96	0.76	*0.34*
DoS GoldenEye	0.81	0.99	0.98	**0.99**	0.99	*0.64*	0.71
DoS Hulk	*0.23*	0.93	0.96	**0.96**	0.96	0.95	0.36
DoS Slowhttptest	*0.35*	0.98	0.99	0.98	**0.99**	0.78	0.38
DoS slowloris	*0.37*	0.95	0.95	**0.96**	0.95	0.74	0.46
FTP-Patator	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00
Heartbleed	**1.00**	0.99	1.00	0.95	0.93	*0.66*	1.00
Infiltration	0.78	0.92	0.88	0.89	**0.92**	*0.52*	0.83
PortScan	*0.39*	1.00	1.00	**1.00**	1.00	0.61	0.85
SSH-Patator	*0.33*	0.96	0.95	**0.96**	0.96	0.83	0.41
Web Attack	0.74	0.97	0.93	**0.97**	0.97	*0.60*	0.84

Table 7 Distribution of results according to type of attack and machine learning algorithm

When analyzing the results presented in Table 7, a compelling pattern emerges: Random Forest, KNN, ID3, and Adaboost algorithms consistently achieve an impressive success rate exceeding 90% for diverse attack types. Among these high performers, ID3 distinguishes itself as the most successful, achieving the highest F-measure in 7 out of 12 tasks. Interestingly, in 6 of these 7 tasks (DDoS, DoS GoldenEye, DoS Hulk, PortScan, SSH-Patator, and Web Attack), ID3 shares the top spot with at least one other algorithm. However, its low processing time places it ahead of its competitors in these scenarios.

In stark contrast, Naive Bayes exhibits the lowest F-measure across all attack types, ranking last in 6 out of 12 tasks. Notably, QDA, another statistical method, closely mirrors Naive Bayes' performance in these tasks. This finding aligns with the established notion that statistical methods like Naive Bayes generally perform less effectively compared to other machine learning algorithms. Further scrutiny reveals that DoS resource depletion attacks (DoS Hulk, DoS Slowhttptest, and DoS Slowloris) comprise half of these 6 tasks where Naive Bayes performs poorly.

Interestingly, Naive Bayes stands out as the top performer for the FTP-Patator attack. While it underperforms on most other attack types, its execution speed shines through in this specific scenario, achieving perfect performance scores across all evaluation metrics. This suggests that the features used to describe the FTP-Patator attack might be highly distinctive, falling within a narrow range relative to normal data, thus making them easily identifiable by Naive Bayes.

Furthermore, all machine learning algorithms achieve flawless performance on the FTP-Patator attack. This could be attributed to the inherent characteristics of the features used to define this attack. These features might be confined to a very limited range compared to normal data, resulting in a clear distinction between normal and attack traffic. As depicted in Figure 13, this unique feature distribution likely facilitates flawless identification of the FTP-Patator attack by all algorithms.

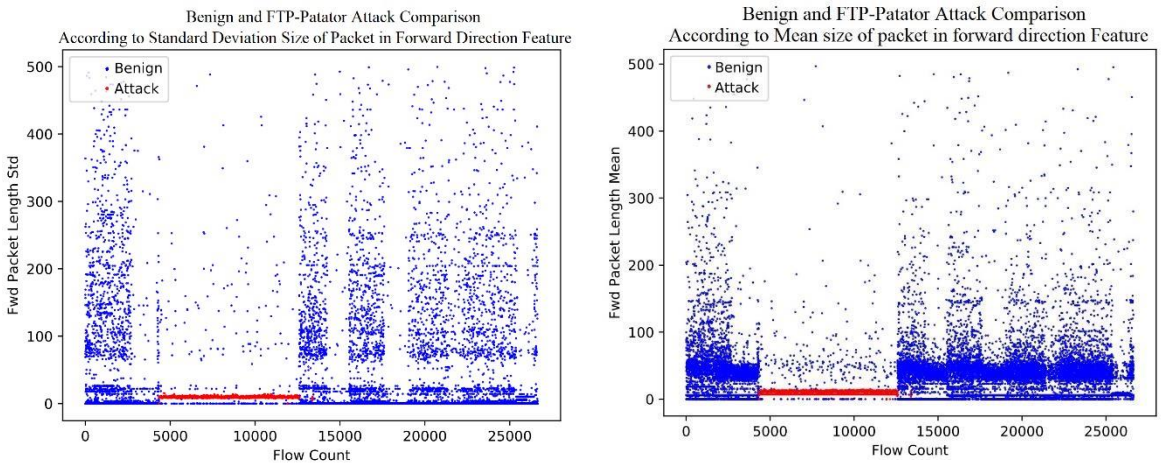


Figure 13 "Fwd Packet Length std" and "Fwd Packet Length mean" features in FTP-Patator and benign Flow

MLP has the second worst performances among the algorithms. It has been the last in the 4 out of 12 tasks. Especially with the Heartbleed and Infiltration attacks, it has a fairly low score. three of the four attacks (Web Attack, Heartbleed, and Infiltration) with the low score have also smallest numbers of records in the dataset. In this context, it can be considered that there is a relationship between the achievement score of MLP and the amount of training data.

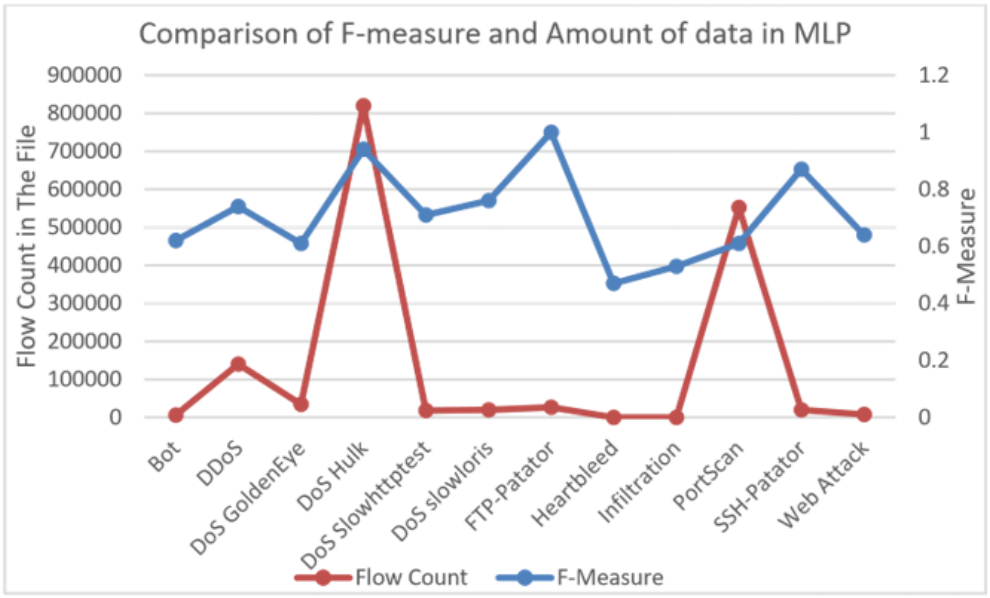


Figure 14 Comparison of the F-measures of the MLP and the flow numbers contained in the attack files.

Figure 14 presents a compelling visual representation of the relationship between data volume and performance for the MLP algorithm. The graph clearly depicts a positive correlation, suggesting that increasing the amount of data positively impacts the algorithm's F-measure. This observation might lead to the assumption that performance gains are directly proportional to data volume for the MLP algorithm. However, this assumption requires further scrutiny and cannot be universally applied.

The limited data availability for specific attack types, such as Heartbleed and Infiltration, presents another critical factor impacting performance evaluation. To gain a deeper understanding of this effect, consider the F-measure data obtained from applying the ID3 algorithm to nine attack files (see Appendix C for all graphical representations). Figure 15 presents the corresponding box-and-whisker plots.

Upon examining these plots, a balanced distribution is observed for most attack types, with minimal variation in F-measure values (between 0% and 2%). However, this balanced distribution deteriorates drastically for two specific attacks: Infiltration and Heartbleed. In Heartbleed, the F-measure range spans 10%, while Infiltration exhibits an even wider range of 8%.

The fundamental reason for this imbalanced distribution lies in the scarcity of data samples for these two attacks. Infiltration comprises only 36 flows, while Heartbleed boasts a mere 11 flows within the entire dataset of 2,830,743 flows.

This significant data imbalance unfortunately renders the results for these two attacks unreliable. Although the observed F-measures for Heartbleed and Infiltration might appear promising, they likely represent overfitting. This assumption is further supported by the substantial fluctuations observed in their respective F-measure values.

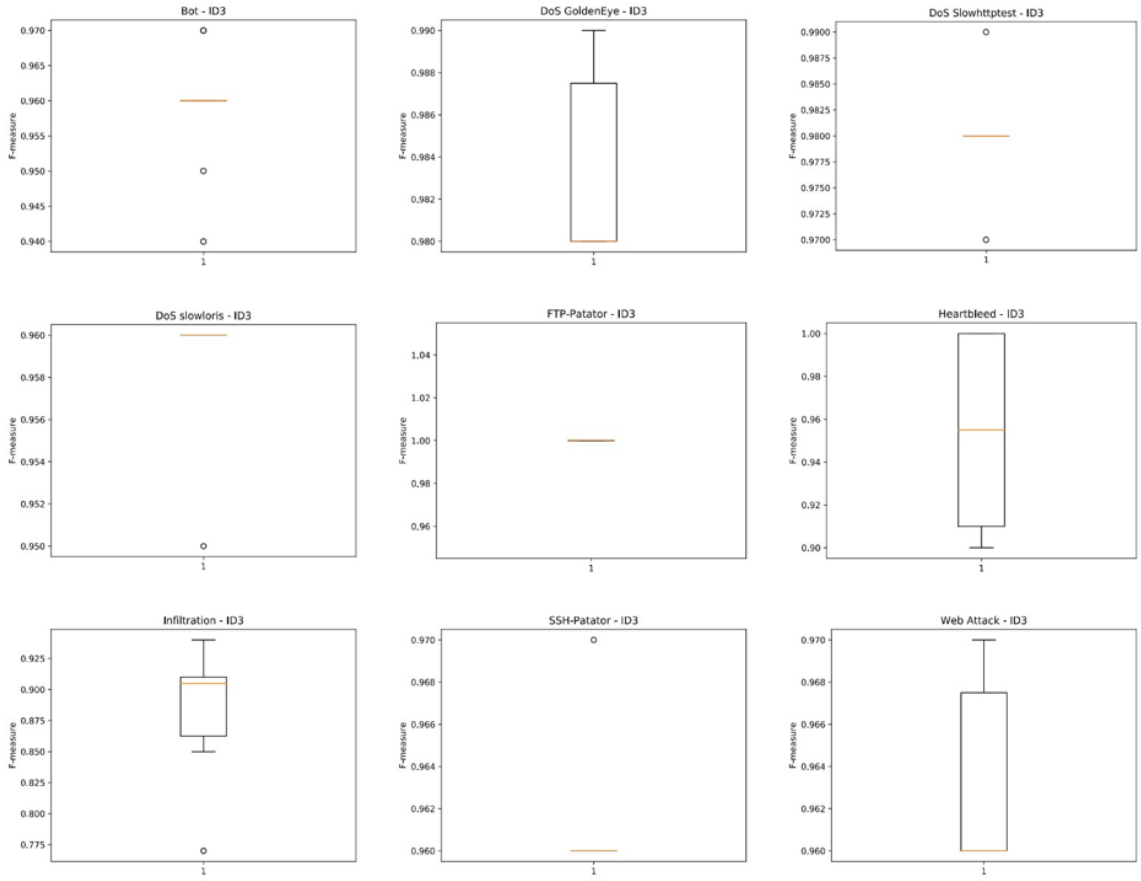


Figure 15 box and whisker graphics containing the results of applying ID3 algorithm to various attack types

4.2 Approach 2 – Using two groups “Attack or Benign”

In this section, the entire data set is used as a single dataset file. All attacks contained in this file are collected under a single common name, "attack". Seven different machine learning methods are applied to this dataset. In this approach, two methods will be used, the first one, the features created for attack files in approach 1 are used. In the second method, the 7 features obtained in the Feature Selection According to Attack or Benign section are used.

4.2.1 Using feature extracted from attack files

Table 8 presents the performance results obtained using the 18 features extracted from the Feature Selection section. Notably, KNN emerges as the top performer with an F-measure of 0.96, closely followed by AdaBoost and ID3 at 0.95. However, ID3 significantly outperforms AdaBoost in terms of execution speed, making it a more attractive choice in this scenario. QDA stands as the least effective algorithm, scoring a disappointing 0.30, roughly 0.40 points lower than its closest rivals (Naive Bayes and MLP).

From a speed perspective, Naive Bayes and QDA shine as the fastest algorithms. Conversely, KNN, despite boasting the highest accuracy, suffers from significantly slower execution compared to its peers.

Machine Learning Algorithms	**Evaluation Criteria**				
	F-Measure	Precision	Recall	Accuracy	Time
Naive Bayes	0.79	0.80	0.78	0.78	**4.576**
QDA	*0.30*	0.84	0.31	0.31	6.649
Random Forest	0.94	0.95	0.94	0.94	24.739
ID3	0.95	0.95	0.95	0.95	29.284
AdaBoost	0.95	0.95	0.95	0.95	391.804
MLP	0.79	0.81	0.84	0.84	81.668
K Nearest Neighbours	**0.96**	0.96	0.97	0.97	*1967.054*

Table 8 application of features obtained from the first approach

4.2.2 Using Feature Selection for All Dataset

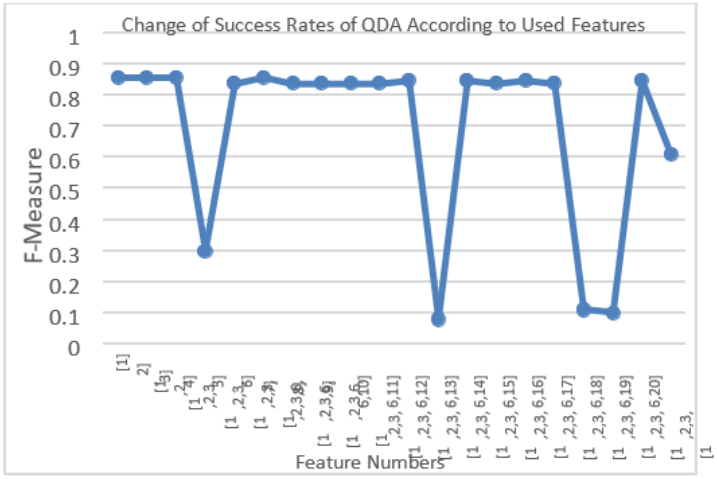
Table 9 presents the performance results using the alternative feature selection method based on attack-specific features. The key changes are highlighted in red for easy comparison. Notably, the algorithms of Random Forest, ID3, Adaboost, and MLP exhibit minimal changes in F-measure when compared to Table 8. However, Naive Bayes and QDA demonstrate substantial improvements, with increases of 2 and 11 points, respectively.

Interestingly, all algorithms exhibit significantly reduced execution times in Table 9 compared to Table 8. This decrease is primarily attributed to the reduced feature count, utilizing only 7 features instead of 18.

Machine Learning Algorithms	Evaluation Criteria				
	F-Measure	Precision	Recall	Accuracy	Time(sec)
Naive Bayes	0.81	0.8	0.82	0.82	1.6258
QDA	0.41	0.83	0.38	0.38	1.925
Random Forest	0.94	0.947	0.94	0.94	20.511
ID3	0.95	0.95	0.95	0.95	11.552
AdaBoost	0.94	0.94	0.94	0.94	144.166
MLP	0.79	0.815	0.84	0.84	51.799
K Nearest Neighbours	0.97	0.97	0.97	0.97	1038.253

Table 9 . Implementation of features obtained using Random Forest Regressor for All Dataset

When looking at the table 9, it is seen that almost all of the results have F-measures above 0.80, but the score obtained in the QDA algorithm is well below this value. In order to investigate the cause of this problem, it can be seen in the figure 16 that illustrates how the QDA algorithm responds to the possibilities of different feature selection.



1-Bwd Packet Length Std	6-Flow IAT Min	11-Flow IAT Mean	16-Fwd Packet Length Mean
2-Flow Bytes/s	7-Fwd IAT Total	12-Total Length of Bwd Packets	17-Total Backward Packets
3-Total Length of Fwd Packets	8-Flow Duration	13-Fwd Packet Length Min	18-Total Fwd Packets
4-Fwd Packet Length Std	9-Bwd Packet Length Max	14-Bwd Packet Length Mean	19-Fwd Packet Length Max
5-Flow IAT Std	10-Flow IAT Max	15-Flow Packets/s	20-Bwd Packet Length Min

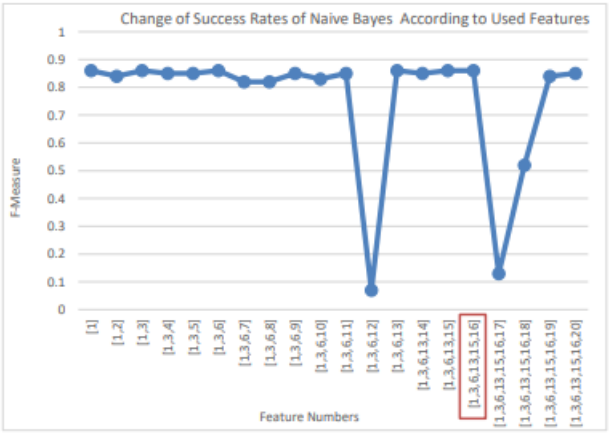
Figure 16 Feature List and F-Measure Changes in different feature selection of QDA Algorithm

To identify the optimal feature set for QDA, a sequential feature selection process is employed. The 20 features are progressively

added to the QDA algorithm based on their importance weight, starting with the most significant. If the F-measure value decreases upon adding a feature, it is removed from the list. This iterative process culminates in the feature list that yields the highest F-measure for QDA.

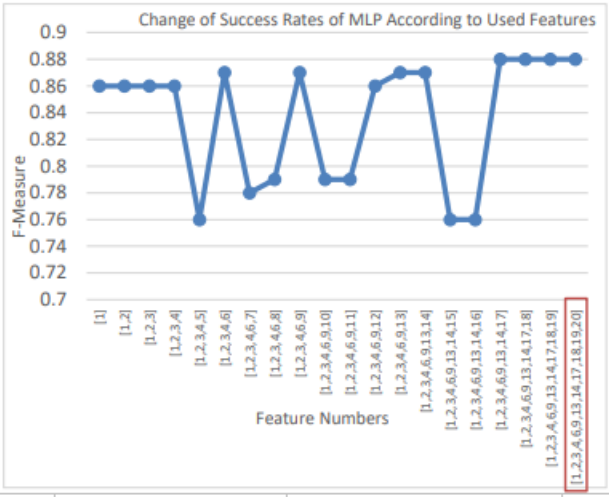
Examination of the resulting graph reveals significant drops in F-measure, particularly when features 3, 11, 16, and 17 are added. Notably, the presence of feature 3 (Fwd Packet Length Std) in both feature sets (Table 8 and 9) potentially contributes to QDA's low performance.

Based on this analysis, the final feature set for QDA is updated to include the feature group with the highest F-measure (features 0, 1, 2, and 5). The application of this sequential selection approach also leads to substantial performance improvements for Naive Bayes and MLP algorithms, as illustrated in Figures 17 and 18.



1-Bwd Packet Length Std	6-Flow IAT Min	11-Flow IAT Mean	16-Fwd Packet Length Mean
2-Flow Bytes/s	7-Fwd IAT Total	12-Total Length of Bwd Packets	17-Total Backward Packets
3-Total Length of Fwd Packets	8-Flow Duration	13-Fwd Packet Length Min	18-Total Fwd Packets
4-Fwd Packet Length Std	9-Bwd Packet Length Max	14-Bwd Packet Length Mean	19-Fwd Packet Length Max
5-Flow IAT Std	10-Flow IAT Max	15-Flow Packets/s	20-Bwd Packet Length Min

Figure 17. F-Measure Changes in Different Feature Selections of Naive Bayes Algorithm



1-Bwd Packet Length Std	6-Flow IAT Min	11-Flow IAT Mean	16-Fwd Packet Length Mean
2-Flow Bytes/s	7-Fwd IAT Total	12-Total Length of Bwd Packets	17-Total Backward Packets
3-Total Length of Fwd Packets	8-Flow Duration	13-Fwd Packet Length Min	18-Total Fwd Packets
4-Fwd Packet Length Std	9-Bwd Packet Length Max	14-Bwd Packet Length Mean	19-Fwd Packet Length Max
5-Flow IAT Std	10-Flow IAT Max	15-Flow Packets/s	20-Bwd Packet Length Min

Figure 18. F-Measure Changes in Different Feature Selections of MLP Algorithm

In this context, if the selected features are updated with the data obtained from the last three graphs according to the algorithms used, the F-Measure scores of these three algorithms (Naive Bayes, QDA and MLP) will increase. Updated features can be seen in Table 10.

Algorithms	**Features**
Naive Bayes	Bwd Packet Length Std, Total Length of Fwd Packets, Flow IAT Min, Fwd Packet Length Min, Flow Packets/s, Fwd Packet Length Mean
QDA	Bwd Packet Length Std, Flow Bytes/s, Total Length of Fwd Packets, Flow IAT Min
MLP	Bwd Packet Length Std, Flow Bytes/s, Total Length of Fwd Packets, Fwd Packet Length Std, Flow IAT Min, Bwd Packet Length Max, Fwd Packet Length Min, Bwd Packet Length Mean, Total Backward Packets, Total Fwd Packets, Fwd Packet Length Max, Bwd Packet Length Min
Random Forest	Bwd Packet Length Std, Flow Bytes/s, Total Length of Fwd Packets, Fwd Packet Length Std, Flow IAT Std, Flow IAT Min, Fwd IAT Total (No changes were made to this feature list. The features obtained in "3.2.3.2 Feature Selection According to Attack or Benign" section are used.)
ID3	
AdaBoost	
K Nearest Neighbours	

Table 10 according to machine learning algorithm updated features

If the application is updated according to the properties contained in Table 10, the following results are obtained. Changed values are highlighted in red.

Machine Learning Algorithms	**Evaluation Criteria**				
	F-Measure	Precision	Recall	Accuracy	Time
Naive Bayes	0.86	0.86	0.87	0.87	**1.8255**
QDA	0.86	0.87	0.88	0.88	2.3696
Random Forest	0.94	0.94	0.94	0.94	19.0899
ID3	0.95	0.95	0.95	0.95	9.5107
AdaBoost	0.94	0.94	0.94	0.94	135.2455
MLP	*0.83*	0.82	0.87	0.87	59.6933
K Nearest Neighbours	**0.97**	0.97	0.97	0.97	*1626.833*

Table 11 The Final Result – Implementation using Table 10

When Table 11, which contains the final results of this section, is examined, according to the previous results, an increase of 0.5 points in Naive Bayes and 0.4 points in MLP is observed. However, the biggest change occurred in the QDA. It reached 0.86 with an increase of 0.45 points.

Thus, the MLP with the lowest performance of 0.83, while the K Nearest Neighbours algorithm has the highest score as in all applications. As for speed, Naive Bayes is the fastest algorithm in all applications, while KNN is the slowest.

5) CONCLUSION AND FUTURE WORK:

5.1 Conclusion:

This research investigates the application of machine learning techniques for detecting network anomalies. To achieve this objective, the CICIDS2017 dataset was chosen due to its contemporary nature, diverse attack range, and inclusion of various network protocols such as mail services, SSH, FTP, HTTP, and HTTPS. The dataset comprises over 80 features defining network flow behavior.

During the implementation, feature importance weights were calculated using the Random Forest Regressor algorithm to determine which features contribute most significantly to anomaly detection. Two distinct approaches were employed:

1. **Attack-Specific Feature Selection:** In this approach, importance weights were calculated individually for each attack type, identifying the features most relevant to specific attack detection strategies.
2. **Holistic Feature Selection:** This approach consolidated all attack types into a single group, calculating importance weights for the entire dataset. This process identified the common features crucial for detecting diverse attack types.

Ultimately, seven diverse machine learning algorithms were applied to the dataset, each exhibiting distinct strengths and characteristics. These algorithms and their corresponding F-measure scores, ranging between 0 and 1, are as follows:

- **Naive Bayes:** 0.86
- **QDA:** 0.86
- **Random Forest:** 0.94
- **ID3:** 0.95
- **AdaBoost:** 0.94
- **MLP:** 0.83
- **K Nearest Neighbors:** 0.97

5.2 Future Work:

While this work has established a strong foundation for network anomaly detection using machine learning, it also presents numerous opportunities for future improvement.

1. Bridging the Gap to Real-Time Applications:

The current approach utilizes pre-recorded data stored in CSV files for training and testing. This setup, while valuable for initial research, lacks practical viability in real-time network security systems. To overcome this limitation, integrating a module capable of capturing and processing live network data for immediate analysis by the machine learning algorithms is crucial.

2. Leveraging Multi-Layered Architectures:

The current study employed independent application of various machine learning algorithms. While this provided valuable insights into individual performance, it may not translate effectively to real-world scenarios. Implementing a multi-layered or hierarchical machine learning architecture offers promising potential. This approach can significantly improve both efficiency and effectiveness.

3. Combining Efficient and Powerful Algorithms:

A two-tiered structure serves as an example of this concept. The first layer would utilize computationally efficient algorithms like Naive Bayes or QDA for continuous, low-cost monitoring of network traffic. Upon detecting an anomaly, the first layer would seamlessly transmit the information to the second layer composed of more powerful algorithms like ID3, AdaBoost, and KNN. This collaborative approach leverages the strengths of each layer, enabling efficient detection coupled with high-performance anomaly classification for informed decision-making.

4. Adapting to Evolving Threats:

The ever-changing landscape of cyber threats necessitates a proactive approach to machine learning algorithms. Continuously incorporating new attack data into the training process allows the models to adapt and evolve, ensuring they remain effective against emerging threats.

5. Exploring Alternative Feature Selection Techniques:

While the utilized feature selection methods yielded promising results, exploring alternative techniques like genetic algorithms or particle swarm optimization algorithms holds the potential for further improvements in accuracy and efficiency.

6) REFERENCES:

"1998 DARPA Intrusion Detection Evaluation Data Set," *Lincoln Laboratory, Massachusetts Institute of Technology*, [Online]. Available: https://www.ll.mit.edu/rd/datasets/1998-darpa-intrusion-detection-evaluation-data-set.

C. Thomas, V. Sharma, and N. Balakrishnan, "Usefulness of DARPA dataset for intrusion detection system evaluation," in *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2008*, 2008, vol. 6973, p. 69730G: International Society for Optics and Photonics.

A. Özgür and H. Erdem, "A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015," *PeerJ PrePrints,* vol. 4, p. e1954v1, 2016.

"CAIDA OC48 Peering Point Traces dataset," *Center for Applied Internet Data Analysis,* [Online]. Available: https://www.caida.org/data/passive/passive_oc48_dataset.xml.

"NSL-KDD dataset," *Canadian Institute for Cybersecurity, University of New Brunswick,* [Online]. Available: http://www.unb.ca/cic/datasets/nsl.html.

"Intrusion detection evaluation dataset (ISCXIDS2012)," *Canadian Institute for Cybersecurity, University of New Brunswick,* [Online]. Available: http://www.unb.ca/cic/datasets/ids.html.

"Intrusion Detection Evaluation Dataset (CICIDS2017)," *Canadian Institute for Cybersecurity, University of New Brunswick,* [Online]. Available: http://www.unb.ca/cic/datasets/ids-2017.html.

"Nmap: *the Network Mapper - Free Security Scanner," Nmap.org. (2018),* [online] Available: https://nmap.org/

R. Christopher, "Port scanning techniques and the defense against them," *SANS Institute,* 2001.

“GoldenEye,” *GitHub, 20-Jun-2018.* [Online]. Available: Available: https://github.com/jseidl/GoldenEye.

“Patator,” *GitHub, 04-Aug-2018.* [Online]. Available: https://github.com/lanjelot/patator.

“A new DOS Perl Program,” *GitHub, 05 Nov 2013.* [Online]. Available: https://github.com/llaera/slowloris.pl.

“slowhttptest,” *GitHub.* [Online]. Available: https://github.com/shekyan/slowhttptest/wiki.

“Slow Read DoS Attack.” *Istanbul Technical University, 07 Sep 2013.* [Online]. Available: https://bidb.itu.edu.tr/eskiler/seyirdefteri/blog/2013/09/07/slow-read-dosattack-(yavaş-okutarak-hizmet-engelleme-saldırısı).

“Ares,” *GitHub, 08-Dec-2017.* [Online]. Available: https://github.com/sweetsoftware/Ares.

"heartleech," *GitHub, 07-Jun-2014.* [Online]. Available: https://github.com/robertdavidgraham/heartleech.

"Python 3.6.6 documentation," *Python Software Foundation.* [Online]. Available: https://docs.python.org/3.6/.

"scikit-learn," scikit-learn 0.19.1 documentation. [Online]. Available: http://scikitlearn.org/stable/.

"Python Data Analysis Library," *pandas: powerful Python data analysis toolkit - pandas 0.22.0 documentation.* [Online]. Available: https://pandas.pydata.org/.

"Matplotlib: *Python plotting - Matplotlib 2.2.2 documentation," Matplotlib.* [Online]. Available: https://matplotlib.org/.

"NumPy,*" NumPy developers.* [Online]. Available: http://www.numpy.org/.

"sklearn.preprocessing.LabelEncoder," *1.4. Support Vector Machines - scikit-learn 0.19.1 documentation.* [Online]. Available: http://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

"train_test_split," *scikit-learn 0.19.1 documentation.* [Online]. Available: http://scikitlearn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html.

J. Brownlee, "Feature Importance and Feature Selection With XGBoost in Python," Machine Learning Mastery, 10-Mar-2018. [Online]. Available: https://machinelearningmastery.com/feature-importance-and-feature-selection-withxgboost-in-python/.

"sklearn.ensemble.RandomForestRegressor," *scikit-learn.* [Online]. Available: http://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.

7) APPENDIX

Appendix A – The list of features and explanation

This list has been taken from: http://www.netflowmeter.ca/netflowmeter.html

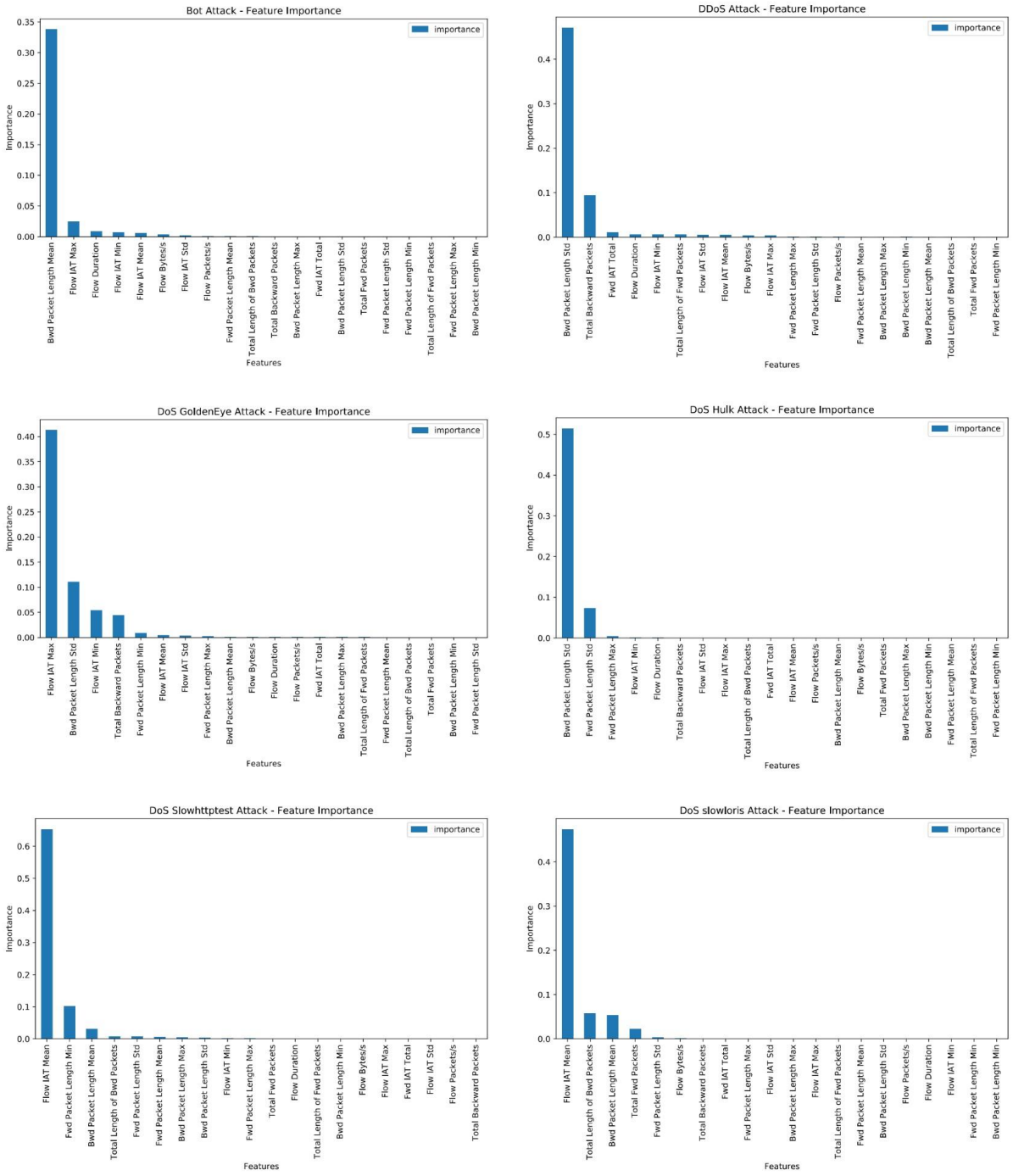
№	Feature Name	Feature Description
1	Flow ID	Flow ID
2	Source IP	Source IP
3	Source Port	Source Port
4	Destination IP	Destination IP
5	Destination Port	Destination Port
6	Protocol	Protocol
7	Timestamp	Timestamp
8	Flow Duration	Duration of the flow in Microsecond
9	Total Fwd Packets	Total packets in the forward direction
10	Total Backward Packets	Total packets in the backward direction
11	Total Length of Fwd Packets	Total size of packet in forward direction
12	Total Length of Bwd Packets	Total size of packet in backward direction
13	Fwd Packet Length Max	Maximum size of packet in forward direction
14	Fwd Packet Length Min	Minimum size of packet in forward direction
15	Fwd Packet Length Mean	Mean size of packet in forward direction
16	Fwd Packet Length Std	Standard deviation size of packet in forward direction
17	Bwd Packet Length Max	Maximum size of packet in backward direction
18	Bwd Packet Length Min	Minimum size of packet in backward direction
19	Bwd Packet Length Mean	Mean size of packet in backward direction
20	Bwd Packet Length Std	Standard deviation size of packet in backward direction
21	Flow Bytes/s	Number of flow bytes per second
22	Flow Packets/s	Number of flow packets per second
23	Flow IAT Mean	Mean length of a flow
24	Flow IAT Std	Standard deviation length of a flow
25	Flow IAT Max	Maximum length of a flow
26	Flow IAT Min	Minimum length of a flow
27	Fwd IAT Total	Total time between two packets sent in the forward direction
28	Fwd IAT Mean	Mean time between two packets sent in the forward direction
29	Fwd IAT Std	Standard deviation time between two packets sent in the forward direction
30	Fwd IAT Max	Maximum time between two packets sent in the forward direction
31	Fwd IAT Min	Minimum time between two packets sent in the forward direction
32	Bwd IAT Total	Total time between two packets sent in the backward direction
33	Bwd IAT Mean	Mean time between two packets sent in the backward direction
34	Bwd IAT Std	Standard deviation time between two packets sent in the backward direction
35	Bwd IAT Max	Maximum time between two packets sent in the backward direction
36	Bwd IAT Min	Minimum time between two packets sent in the backward direction
37	Fwd PSH Flags	Number of packets with PUSH
38	Bwd PSH Flags	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)
39	Fwd URG Flags	Number of times the URG flag was set in packets travelling in the forward direction (0 for UDP)
40	Bwd URG Flags	Number of times the URG flag was set in packets travelling in the backward direction (0 for UDP)
41	Fwd Header Length	Total bytes used for headers in the forward direction
42	Bwd Header Length	Total bytes used for headers in the backrward direction
43	Fwd Packets/s	Number of forward packets per second
44	Bwd Packets/s	Number of backward packets per second

Appendix A Figure 1. List of Features in CICIDS 2017 Dataset and it's explanation

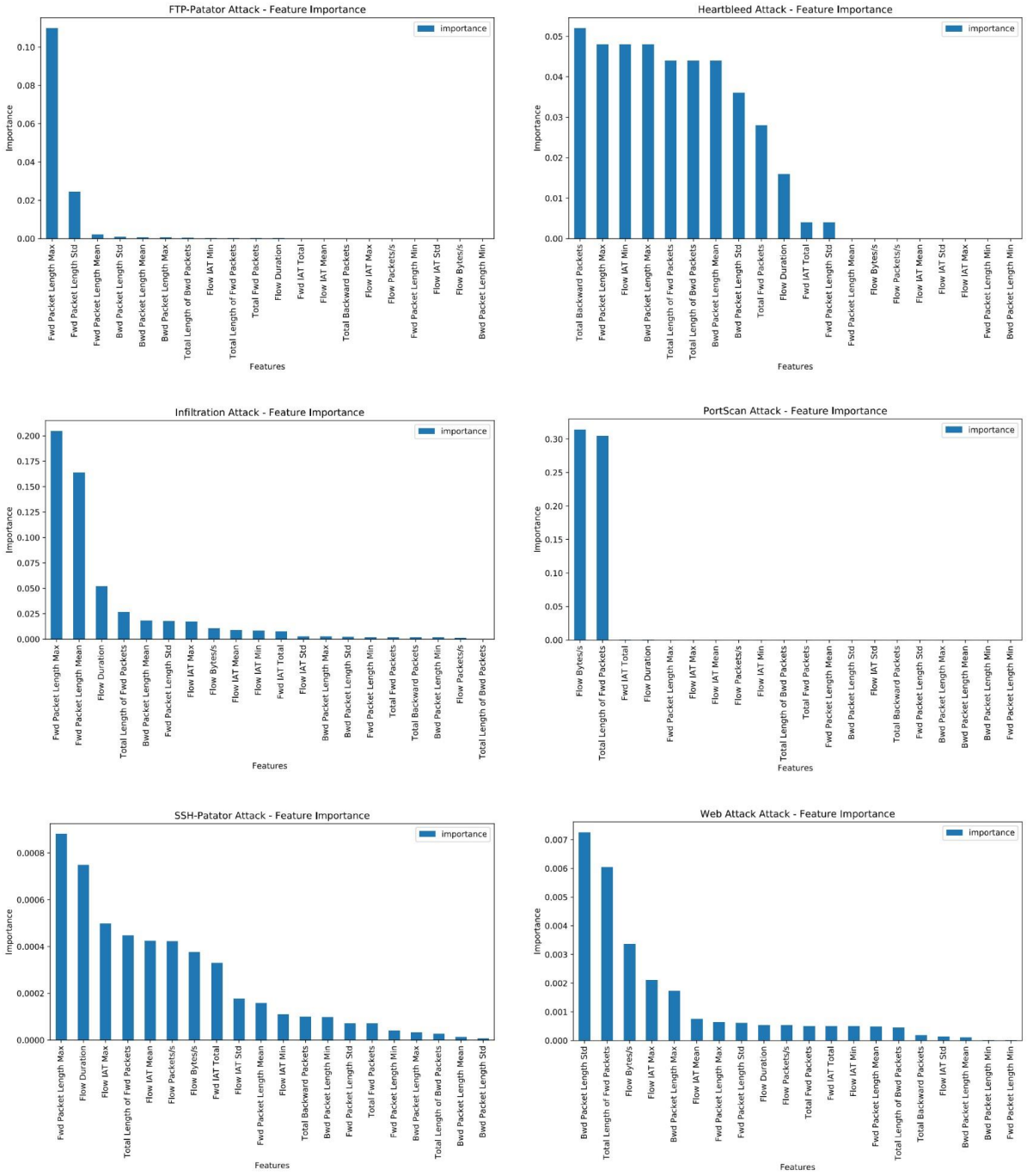
№	Feature Name	Feature Description
45	Min Packet Length	Minimum inter-arrival time of packet
46	Max Packet Length	Maximum inter-arrival time of packet
47	Packet Length Mean	Mean inter-arrival time of packet
48	Packet Length Std	Standard deviation inter-arrival time of packet
49	Packet Length Variance	Packet Length Variance
50	FIN Flag Count	Number of packets with FIN
51	SYN Flag Count	Number of packets with SYN
52	RST Flag Count	Number of packets with RST
53	PSH Flag Count	Number of packets with PUSH
54	ACK Flag Count	Number of packets with ACK
55	URG Flag Count	Number of packets with URG
56	CWE Flag Count	Number of packets with CWE
57	ECE Flag Count	Number of packets with ECE
58	Down/Up Ratio	Download and upload ratio
59	Average Packet Size	Average size of packet
60	Avg Fwd Segment Size	Average size observed in the forward direction
61	Avg Bwd Segment Size	Average size observed in the backward direction
62	Fwd Avg Bytes/Bulk	Average number of bytes bulk rate in the forward direction
63	Fwd Avg Packets/Bulk	Average number of packets bulk rate in the forward direction
64	Bwd Avg Bulk Rate	Average number of bulk rate in the backward direction
65	Bwd Avg Bytes/Bulk	Average number of bytes bulk rate in the backward direction
66	Bwd Avg Packets/Bulk	Average number of packets bulk rate in the backward direction
67	Bwd Avg Bulk Rate	Average number of bulk rate in the backward direction
68	Subflow Fwd Packets	The average number of packets in a sub flow in the forward direction
69	Subflow Fwd Bytes	The average number of bytes in a sub flow in the forward direction
70	Subflow Bwd Packets	The average number of packets in a sub flow in the backward direction
71	Subflow Bwd Bytes	The average number of bytes in a sub flow in the backward direction
72	Init_Win_bytes_forward	The total number of bytes sent in initial window in the forward direction
73	Init_Win_bytes_backward	The total number of bytes sent in initial window in the backward direction
74	act_data_pkt_fwd	Count of packets with at least 1 byte of TCP data payload in the forward direction
75	min_seg_size_forward	Minimum segment size observed in the forward direction
76	Active Mean	Mean time a flow was active before becoming idle
77	Active Std	Standard deviation time a flow was active before becoming idle
78	Active Max	Maximum time a flow was active before becoming idle
79	Active Min	Minimum time a flow was active before becoming idle
80	Idle Mean	Mean time a flow was idle before becoming active
81	Idle Std	Standard deviation time a flow was idle before becoming active
82	Idle Max	Maximum time a flow was idle before becoming active
83	Idle Min	Minimum time a flow was idle before becoming active
84	Label	Label
85	External IP	External IP

Appendix A Figure 2. List of Features in CICIDS 2017 Dataset and it's explanation

Appendix B – Feature priority weights, according to attacks



Appendix B Figure 1. Feature Priority weights according to attacks



Appendix B Figure 2. Feature Priority weights according to attacks

Bot attack importance list:

	Features	importance
1	Bwd Packet Length Mean	0.338129
2	Flow IAT Max	0.024407
3	Flow Duration	0.008495
4	Flow IAT Min	0.007067
5	Flow IAT Mean	0.005609
6	Flow Bytes/s	0.003518
7	Flow IAT Std	0.002213
8	Flow Packets/s	0.000667
9	Fwd Packet Length Mean	0.000418
10	Total Length of Bwd Packets	0.000333
11	Total Backward Packets	0.000223
12	Bwd Packet Length Max	0.000212
13	Fwd IAT Total	0.000141
14	Bwd Packet Length Std	0.000083
15	Total Fwd Packets	0.000077
16	Fwd Packet Length Std	0.000075
17	Fwd Packet Length Min	0.000069
18	Total Length of Fwd Packets	0.000048
19	Fwd Packet Length Max	0.000036
20	Bwd Packet Length Min	0.000005

DoS Slowhttptest attack importance list:

	Features	importance
1	Flow IAT Mean	0.653023
2	Fwd Packet Length Min	0.101739
3	Bwd Packet Length Mean	0.029976
4	Total Length of Bwd Packets	0.007153
5	Fwd Packet Length Std	0.007107
6	Fwd Packet Length Mean	0.006074
7	Bwd Packet Length Max	0.003857
8	Bwd Packet Length Std	0.002934
9	Flow IAT Min	0.002184
10	Fwd Packet Length Max	0.001245
11	Total Fwd Packets	0.000808
12	Flow Duration	0.000802
13	Total Length of Fwd Packets	0.000604
14	Bwd Packet Length Min	0.000545
15	Flow Bytes/s	0.000421
16	Flow IAT Max	0.000327
17	Fwd IAT Total	0.000284
18	Flow IAT Std	0.000252
19	Flow Packets/s	0.000159
20	Total Backward Packets	0.000130

Infiltration attack importance list:

	Features	importance
1	Fwd Packet Length Max	0.204751
2	Fwd Packet Length Mean	0.163696
3	Flow Duration	0.052250
4	Total Length of Fwd Packets	0.026823
5	Bwd Packet Length Mean	0.018539
6	Fwd Packet Length Std	0.017846
7	Flow IAT Max	0.017182
8	Flow Bytes/s	0.010812
9	Flow IAT Mean	0.008816
10	Flow IAT Min	0.008310
11	Fwd IAT Total	0.007726
12	Flow IAT Std	0.002862
13	Bwd Packet Length Max	0.002737
14	Bwd Packet Length Std	0.002030
15	Fwd Packet Length Min	0.001937
16	Total Fwd Packets	0.001862
17	Total Backward Packets	0.001680
18	Bwd Packet Length Min	0.001610
19	Flow Packets/s	0.001406
20	Total Length of Bwd Packets	0.000000

DDoS attack importance list:

	Features	importance
1	Bwd Packet Length Std	0.471368
2	Total Backward Packets	0.093576
3	Fwd IAT Total	0.010827
4	Flow Duration	0.006320
5	Flow IAT Min	0.006110
6	Total Length of Fwd Packets	0.006037
7	Flow IAT Std	0.005439
8	Flow IAT Mean	0.005238
9	Flow Bytes/s	0.004741
10	Flow IAT Max	0.004703
11	Fwd Packet Length Max	0.001647
12	Fwd Packet Length Std	0.001406
13	Flow Packets/s	0.001019
14	Fwd Packet Length Mean	0.000619
15	Bwd Packet Length Max	0.000577
16	Bwd Packet Length Min	0.000405
17	Bwd Packet Length Mean	0.000197
18	Total Length of Bwd Packets	0.000085
19	Total Fwd Packets	0.000042
20	Fwd Packet Length Min	0.000017

DoS slowloris attack importance list:

	Features	importance
1	Flow IAT Mean	0.473856
2	Total Length of Bwd Packets	0.057384
3	Bwd Packet Length Mean	0.053022
4	Total Fwd Packets	0.021895
5	Fwd Packet Length Std	0.002831
6	Flow Bytes/s	0.000965
7	Total Backward Packets	0.000769
8	Fwd IAT Total	0.000755
9	Fwd Packet Length Max	0.000728
10	Flow IAT Std	0.000664
11	Bwd Packet Length Max	0.000608
12	Flow IAT Max	0.000603
13	Total Length of Fwd Packets	0.000579
14	Fwd Packet Length Mean	0.000537
15	Bwd Packet Length Std	0.000528
16	Flow Packets/s	0.000429
17	Flow Duration	0.000375
18	Flow IAT Min	0.000348
19	Fwd Packet Length Min	0.000152
20	Bwd Packet Length Min	0.000021

PortScan attack importance list:

	Features	importance
1	Flow Bytes/s	0.313933
2	Total Length of Fwd Packets	0.304613
3	Fwd IAT Total	0.000427
4	Flow Duration	0.000398
5	Fwd Packet Length Max	0.000150
6	Flow IAT Max	0.000059
7	Flow IAT Mean	0.000054
8	Flow Packets/s	0.000031
9	Flow IAT Min	0.000031
10	Total Length of Bwd Packets	0.000022
11	Total Fwd Packets	0.000021
12	Fwd Packet Length Mean	0.000019
13	Bwd Packet Length Std	0.000018
14	Flow IAT Std	0.000017
15	Total Backward Packets	0.000014
16	Fwd Packet Length Std	0.000009
17	Bwd Packet Length Max	0.000004
18	Bwd Packet Length Mean	0.000002
19	Bwd Packet Length Min	0.000002
20	Fwd Packet Length Min	0.000001

DoS GoldenEye attack importance list:

	Features	importance
1	Flow IAT Max	0.413073
2	Bwd Packet Length Std	0.111039
3	Flow IAT Min	0.054021
4	Total Backward Packets	0.044895
5	Fwd Packet Length Min	0.009531
6	Flow IAT Mean	0.004040
7	Flow IAT Std	0.003863
8	Fwd Packet Length Max	0.002899
9	Bwd Packet Length Mean	0.001228
10	Flow Bytes/s	0.000928
11	Flow Duration	0.000927
12	Flow Packets/s	0.000892
13	Fwd IAT Total	0.000719
14	Bwd Packet Length Max	0.000445
15	Total Length of Fwd Packets	0.000406
16	Fwd Packet Length Mean	0.000393
17	Total Length of Bwd Packets	0.000217
18	Total Fwd Packets	0.000065
19	Bwd Packet Length Min	0.000033
20	Fwd Packet Length Std	0.000027

FTP-Patator attack importance list:

	Features	importance
1	Fwd Packet Length Max	1.098307e-01
2	Fwd Packet Length Std	2.437956e-02
3	Fwd Packet Length Mean	2.200624e-03
4	Bwd Packet Length Std	8.997715e-04
5	Bwd Packet Length Mean	7.081365e-04
6	Bwd Packet Length Max	6.054917e-04
7	Total Length of Bwd Packets	4.015506e-04
8	Flow IAT Min	2.611238e-04
9	Total Length of Fwd Packets	1.766284e-04
10	Total Fwd Packets	1.634693e-04
11	Flow Duration	1.330083e-04
12	Fwd IAT Total	9.598814e-05
13	Flow IAT Mean	9.187611e-05
14	Total Backward Packets	8.884482e-05
15	Flow IAT Max	8.359459e-05
16	Flow Packets/s	7.030109e-05
17	Fwd Packet Length Min	4.619749e-05
18	Flow IAT Std	2.987659e-05
19	Flow Bytes/s	2.244798e-05
20	Bwd Packet Length Min	4.764849e-07

SSH-Patator attack importance list:

	Features	importance
1	Fwd Packet Length Max	0.000881
2	Flow Duration	0.000748
3	Flow IAT Max	0.000497
4	Total Length of Fwd Packets	0.000448
5	Flow IAT Mean	0.000425
6	Flow Packets/s	0.000423
7	Flow Bytes/s	0.000375
8	Fwd IAT Total	0.000329
9	Flow IAT Std	0.000177
10	Fwd Packet Length Mean	0.000158
11	Flow IAT Min	0.000111
12	Total Backward Packets	0.000100
13	Bwd Packet Length Min	0.000099
14	Fwd Packet Length Std	0.000070
15	Total Fwd Packets	0.000070
16	Fwd Packet Length Min	0.000040
17	Bwd Packet Length Max	0.000032
18	Total Length of Bwd Packets	0.000027
19	Bwd Packet Length Mean	0.000014
20	Bwd Packet Length Std	0.000008

DoS Hulk attack importance list:

	Features	importance
1	Bwd Packet Length Std	5.148222e-01
2	Fwd Packet Length Std	7.321079e-02
3	Fwd Packet Length Max	4.515548e-03
4	Flow IAT Min	1.675778e-03
5	Flow Duration	1.218072e-03
6	Total Backward Packets	3.813481e-04
7	Flow IAT Std	2.572354e-04
8	Flow IAT Max	2.517998e-04
9	Total Length of Bwd Packets	1.778769e-04
10	Fwd IAT Total	1.739909e-04
11	Flow IAT Mean	9.875828e-05
12	Flow Packets/s	8.114421e-05
13	Bwd Packet Length Mean	5.449508e-05
14	Flow Bytes/s	2.752602e-05
15	Total Fwd Packets	1.227050e-05
16	Bwd Packet Length Max	1.004453e-05
17	Bwd Packet Length Min	9.303096e-06
18	Fwd Packet Length Mean	8.013636e-06
19	Total Length of Fwd Packets	4.604820e-06
20	Fwd Packet Length Min	1.810830e-08

Heartbleed attack importance list:

	Features	importance
1	Total Backward Packets	0.052
2	Fwd Packet Length Max	0.048
3	Flow IAT Min	0.048
4	Bwd Packet Length Max	0.048
5	Total Length of Fwd Packets	0.044
6	Total Length of Bwd Packets	0.044
7	Bwd Packet Length Mean	0.044
8	Bwd Packet Length Std	0.036
9	Total Fwd Packets	0.028
10	Flow Duration	0.016
11	Fwd IAT Total	0.004
12	Fwd Packet Length Std	0.004
13	Fwd Packet Length Mean	0.000
14	Flow Bytes/s	0.000
15	Flow Packets/s	0.000
16	Flow IAT Mean	0.000
17	Flow IAT Std	0.000
18	Flow IAT Max	0.000
19	Fwd Packet Length Min	0.000
20	Bwd Packet Length Min	0.000

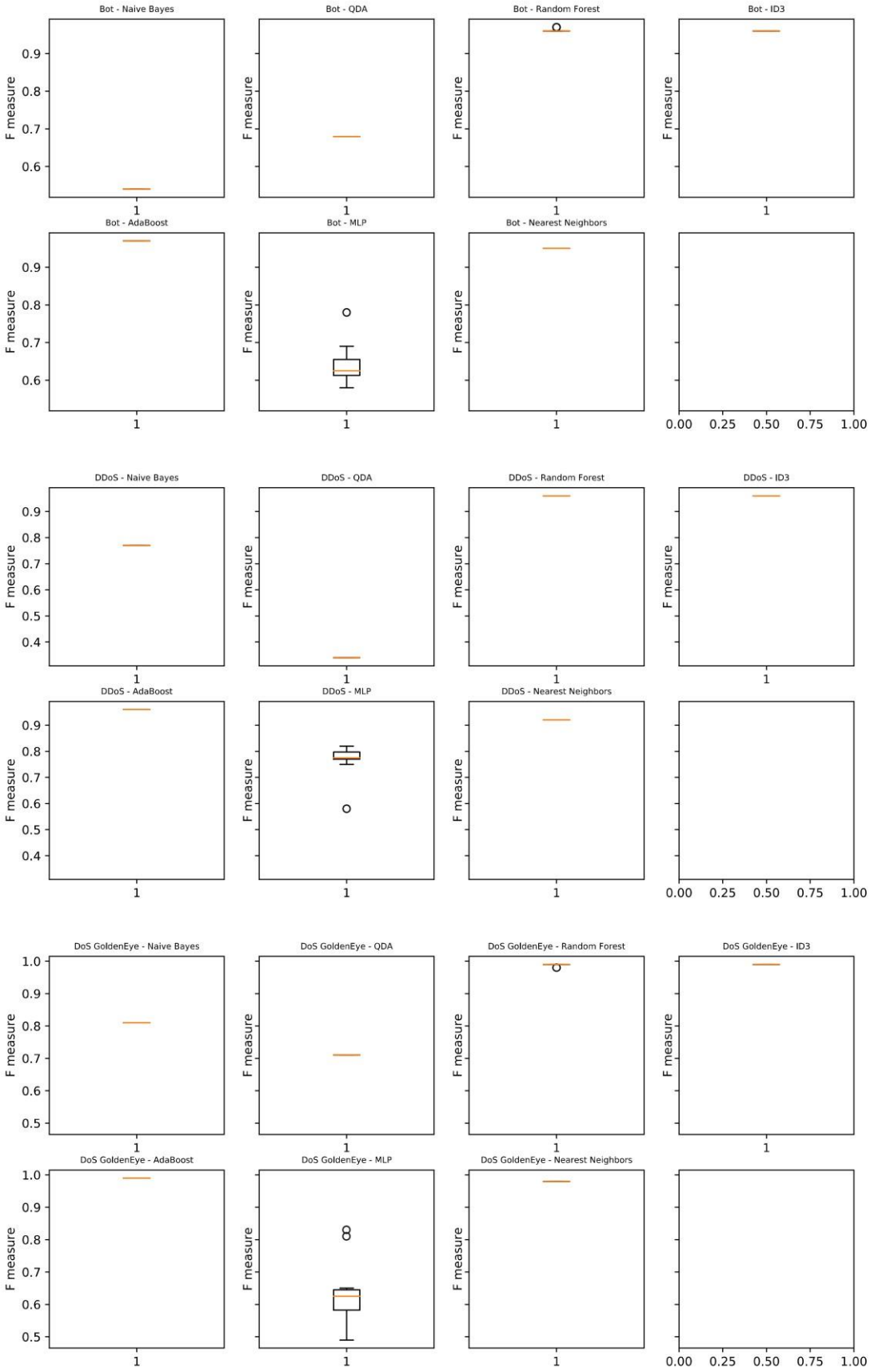
Web Attack attack importance list:

	Features	importance
1	Bwd Packet Length Std	0.007255
2	Total Length of Fwd Packets	0.006046
3	Flow Bytes/s	0.003366
4	Flow IAT Max	0.002102
5	Bwd Packet Length Max	0.001728
6	Flow IAT Mean	0.000760
7	Fwd Packet Length Max	0.000638
8	Fwd Packet Length Std	0.000616
9	Flow Duration	0.000541
10	Flow Packets/s	0.000526
11	Total Fwd Packets	0.000506
12	Fwd IAT Total	0.000505
13	Flow IAT Min	0.000499
14	Fwd Packet Length Mean	0.000490
15	Total Length of Bwd Packets	0.000454
16	Total Backward Packets	0.000177
17	Flow IAT Std	0.000140
18	Bwd Packet Length Mean	0.000102
19	Bwd Packet Length Min	0.000016
20	Fwd Packet Length Min	0.000008

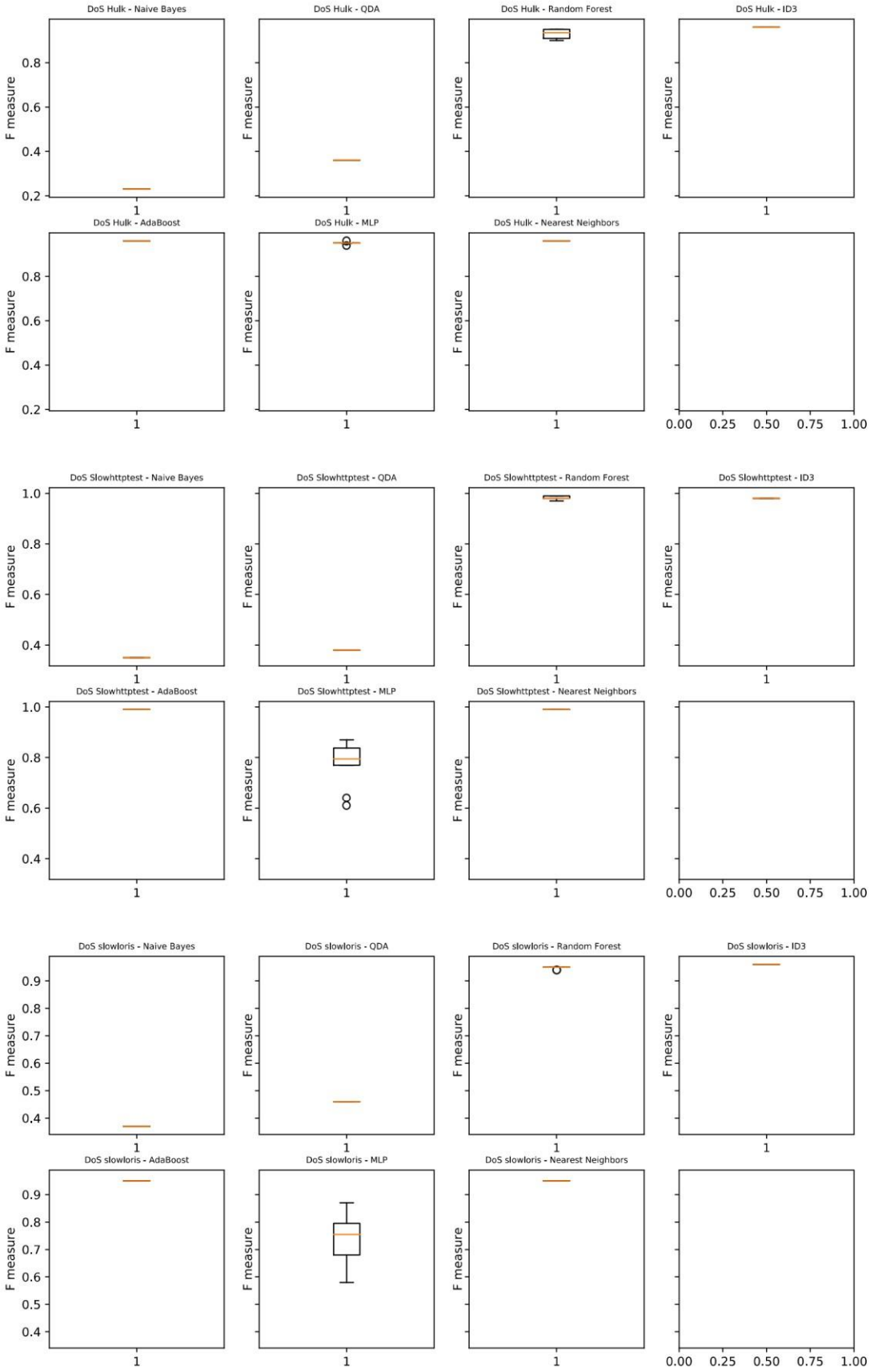
Appendix B Figure 3. Feature Priority weight values, according to attacks

Appendix C. The Machine Learning Implementation Results (According to Attacks)

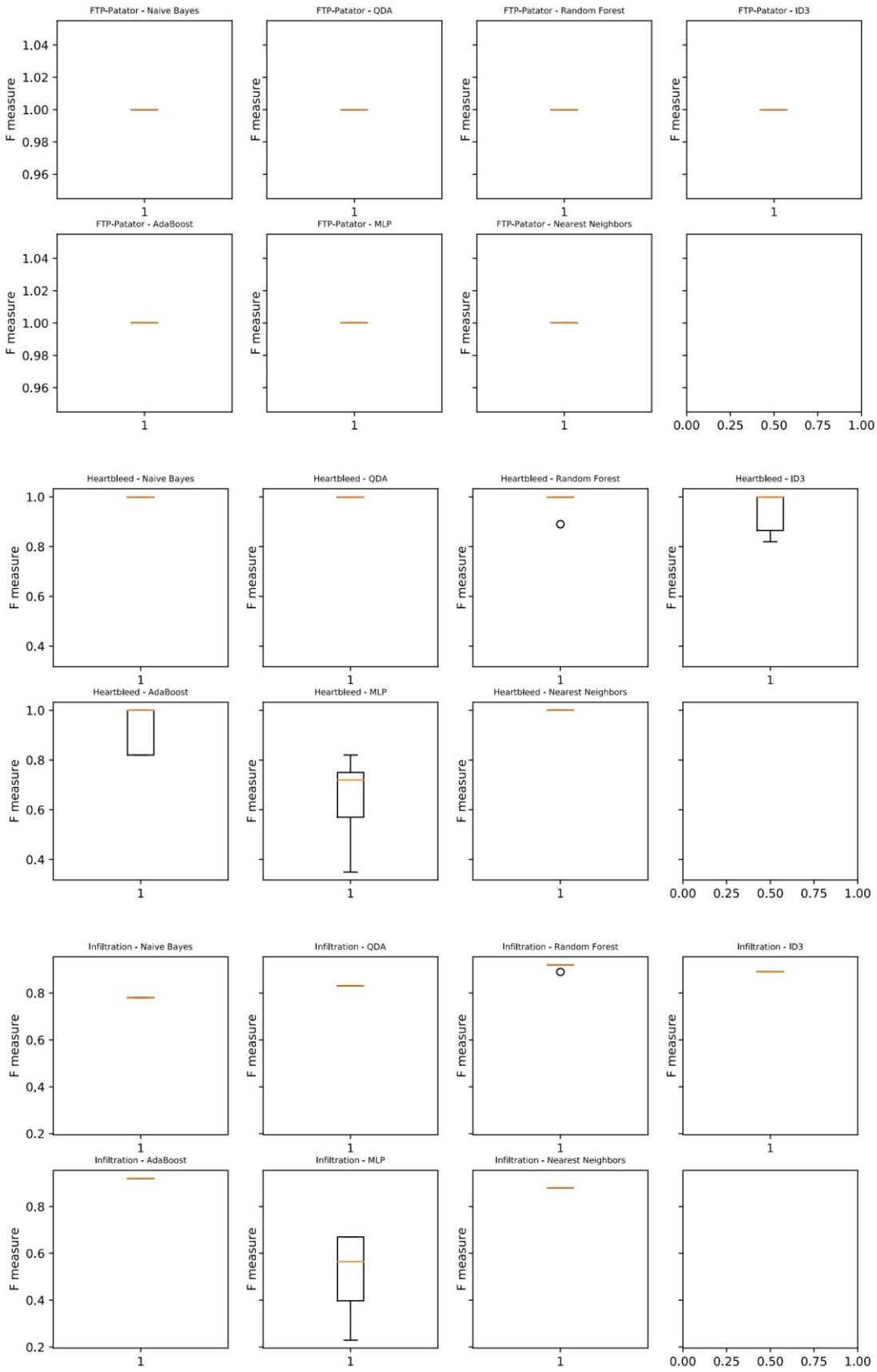
	NaiveBayes					RandomForrest					KNN					ID3					Adaboost					MLP					QDA				
	Acc	F1	Pr	Rc	Time	Acc	F1	Pr	Rc	Time	Acc	F1	Pr	Rc	Time	Acc	F1	Pr	Rc	Time	Acc	F1	Pr	Rc	Time	Acc	F1	Pr	Rc	Time	Acc	F1	Pr	Rc	Time
Bot	0.56	0.55	0.82	0.56	0.003	0.97	0.97	0.97	0.97	0.030	0.96	0.96	0.96	0.96	0.014	0.97	0.97	0.97	0.97	0.008	0.98	0.98	0.98	0.98	0.170	0.69	0.62	0.61	0.69	0.125	0.68	0.68	0.84	0.68	0.003
DDoS	0.77	0.76	0.76	0.77	0.045	0.96	0.96	0.97	0.96	0.430	0.93	0.93	0.93	0.93	1.361	0.96	0.96	0.97	0.96	0.200	0.96	0.96	0.96	0.96	2.818	0.77	0.74	0.76	0.77	4.962	0.42	0.35	0.80	0.42	0.054
DoS GoldenEye	0.82	0.80	0.82	0.82	0.011	0.99	0.99	0.99	0.99	0.096	0.98	0.98	0.98	0.98	0.093	0.98	0.98	0.98	0.98	0.043	0.98	0.98	0.98	0.98	0.674	0.62	0.61	0.72	0.62	0.711	0.95	0.95	0.95	0.95	0.013
DoS Hulk	0.34	0.23	0.80	0.34	0.298	0.94	0.93	0.94	0.94	3.738	0.96	0.96	0.96	0.96	254.4	0.96	0.96	0.96	0.96	0.909	0.96	0.96	0.96	0.96	22.16	0.94	0.94	0.94	0.94	25.63	0.41	0.36	0.81	0.41	0.319
DoS Slowhttptest	0.41	0.36	0.73	0.41	0.006	0.98	0.98	0.98	0.98	0.056	0.99	0.99	0.99	0.99	0.058	0.98	0.98	0.99	0.98	0.020	0.99	0.99	0.99	0.99	0.313	0.72	0.71	0.83	0.72	0.387	0.42	0.38	0.74	0.42	0.006
DoS slowloris	0.42	0.36	0.80	0.42	0.006	0.95	0.94	0.94	0.94	0.055	0.95	0.95	0.95	0.95	0.035	0.96	0.96	0.96	0.96	0.022	0.95	0.95	0.95	0.95	0.372	0.77	0.76	0.80	0.77	0.494	0.48	0.46	0.79	0.48	0.008
FTP-Patator	1.00	1.00	1.00	1.00	0.006	1.00	1.00	1.00	1.00	0.057	1.00	1.00	1.00	1.00	0.214	1.00	1.00	1.00	1.00	0.014	1.00	1.00	1.00	1.00	0.412	1.00	1.00	1.00	1.00	2.683	1.00	1.00	1.00	1.00	0.008
Heartbleed	1.00	1.00	1.00	1.00	0.004	1.00	1.00	1.00	1.00	0.011	1.00	1.00	1.00	1.00	0.002	0.95	0.95	0.98	0.95	0.001	0.95	0.94	0.94	0.95	0.003	0.52	0.47	0.47	0.52	0.011	1.00	1.00	1.00	1.00	0.005
Infiltration	0.82	0.79	0.82	0.82	0.002	0.93	0.93	0.95	0.93	0.011	0.85	0.86	0.87	0.85	0.003	0.91	0.91	0.94	0.91	0.002	0.90	0.90	0.93	0.90	0.051	0.59	0.53	0.53	0.59	0.008	0.83	0.82	0.85	0.83	0.002
PortScan	0.44	0.39	0.80	0.44	0.185	1.00	1.00	1.00	1.00	2.554	1.00	1.00	1.00	1.00	54.72	1.00	1.00	1.00	1.00	0.784	1.00	1.00	1.00	1.00	15.01	0.72	0.61	0.63	0.72	13.77	0.84	0.84	0.89	0.84	0.205
SSH-Patator	0.41	0.34	0.80	0.41	0.008	0.96	0.96	0.96	0.96	0.069	0.96	0.96	0.96	0.96	0.045	0.96	0.96	0.97	0.96	0.027	0.96	0.96	0.97	0.96	0.411	0.87	0.87	0.88	0.87	0.324	0.47	0.43	0.80	0.47	0.006
Web Attack	0.73	0.75	0.86	0.74	0.005	0.97	0.97	0.97	0.97	0.029	0.93	0.94	0.94	0.94	0.014	0.96	0.96	0.96	0.96	0.009	0.97	0.96	0.96	0.97	0.170	0.69	0.64	0.68	0.69	0.111	0.83	0.84	0.89	0.84	0.003



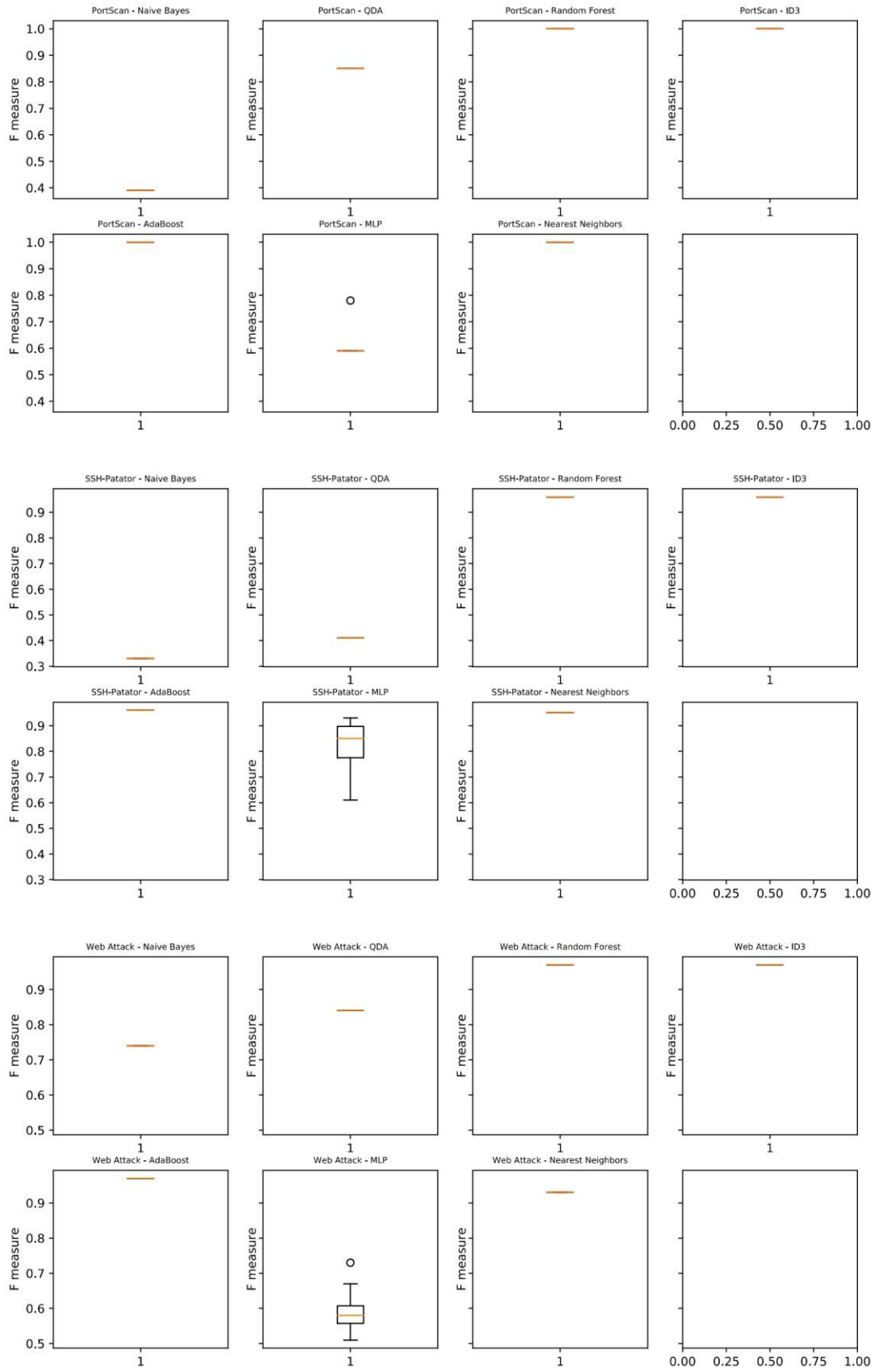
Appendix C Figure 2. *Machine Learning Implementation Box and Whisker Graph (F-Measure)*



Appendix C Figure 3. *Machine Learning Implementation Box and Whisker Graph (F-Measure)*

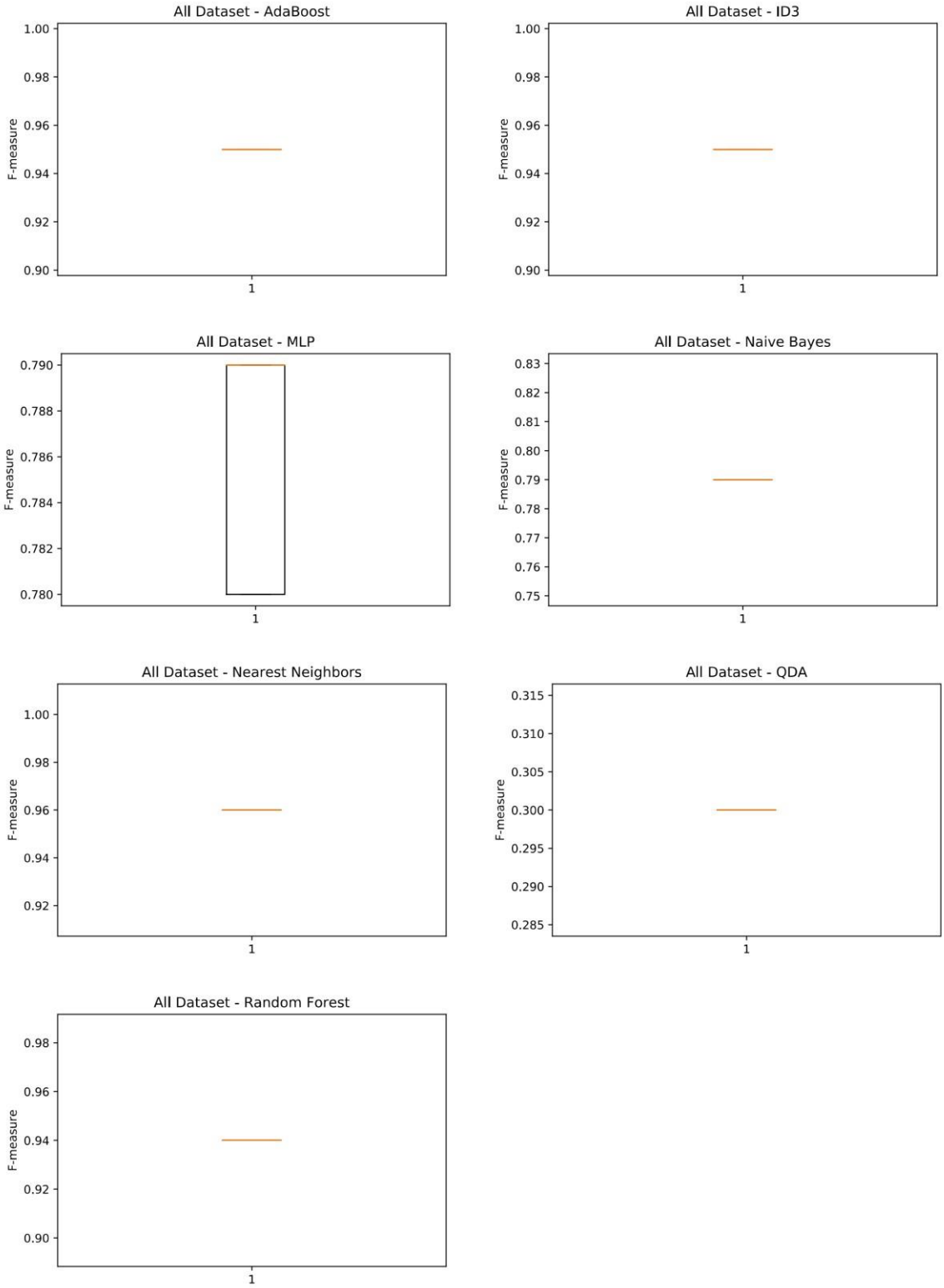


Appendix C Figure 4. *Machine Learning Implementation Box and Whisker Graph (F-Measure)*

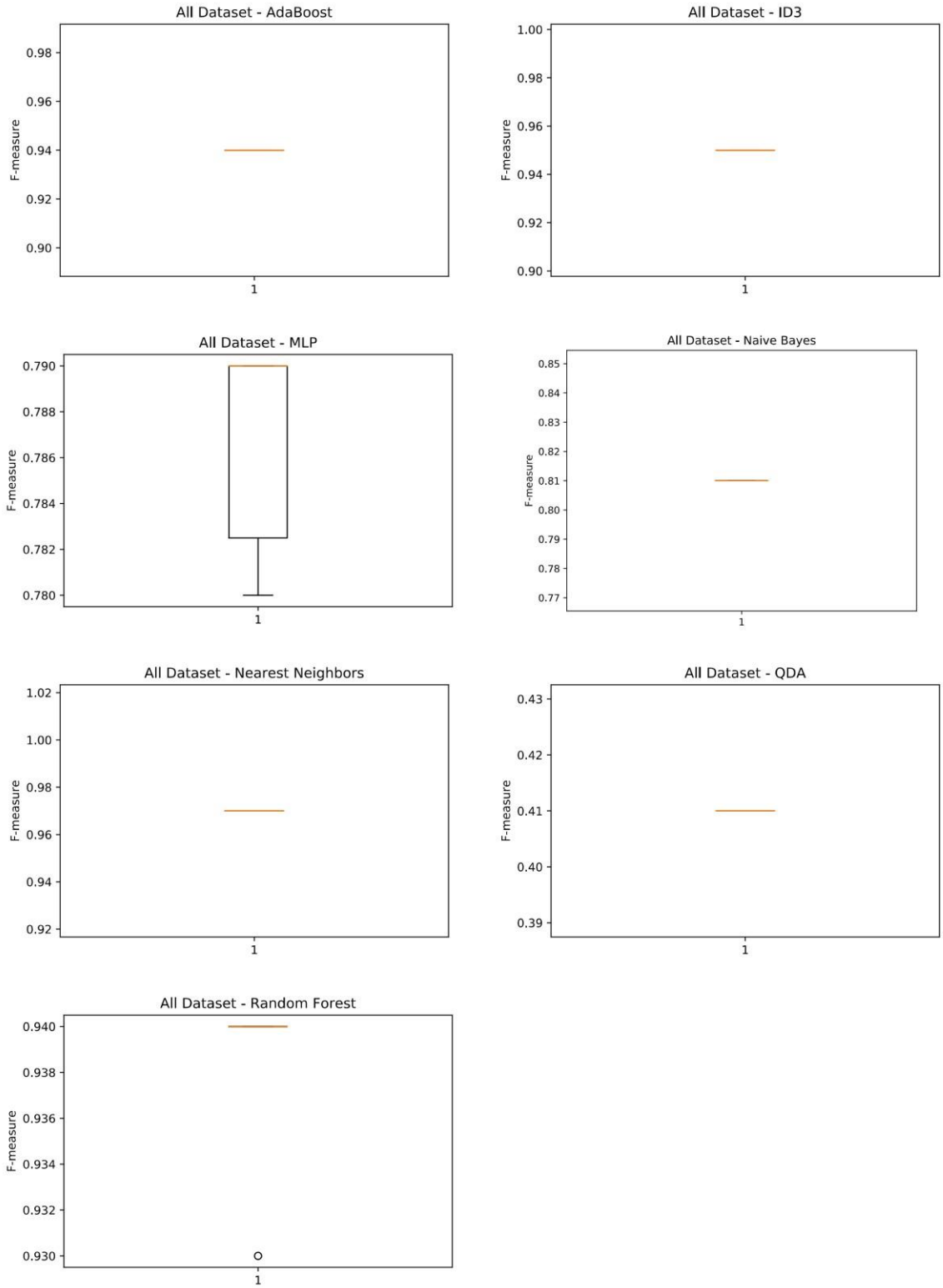


Appendix C Figure 5. *Machine Learning Implementation Box and Whisker Graph (F-Measure)*

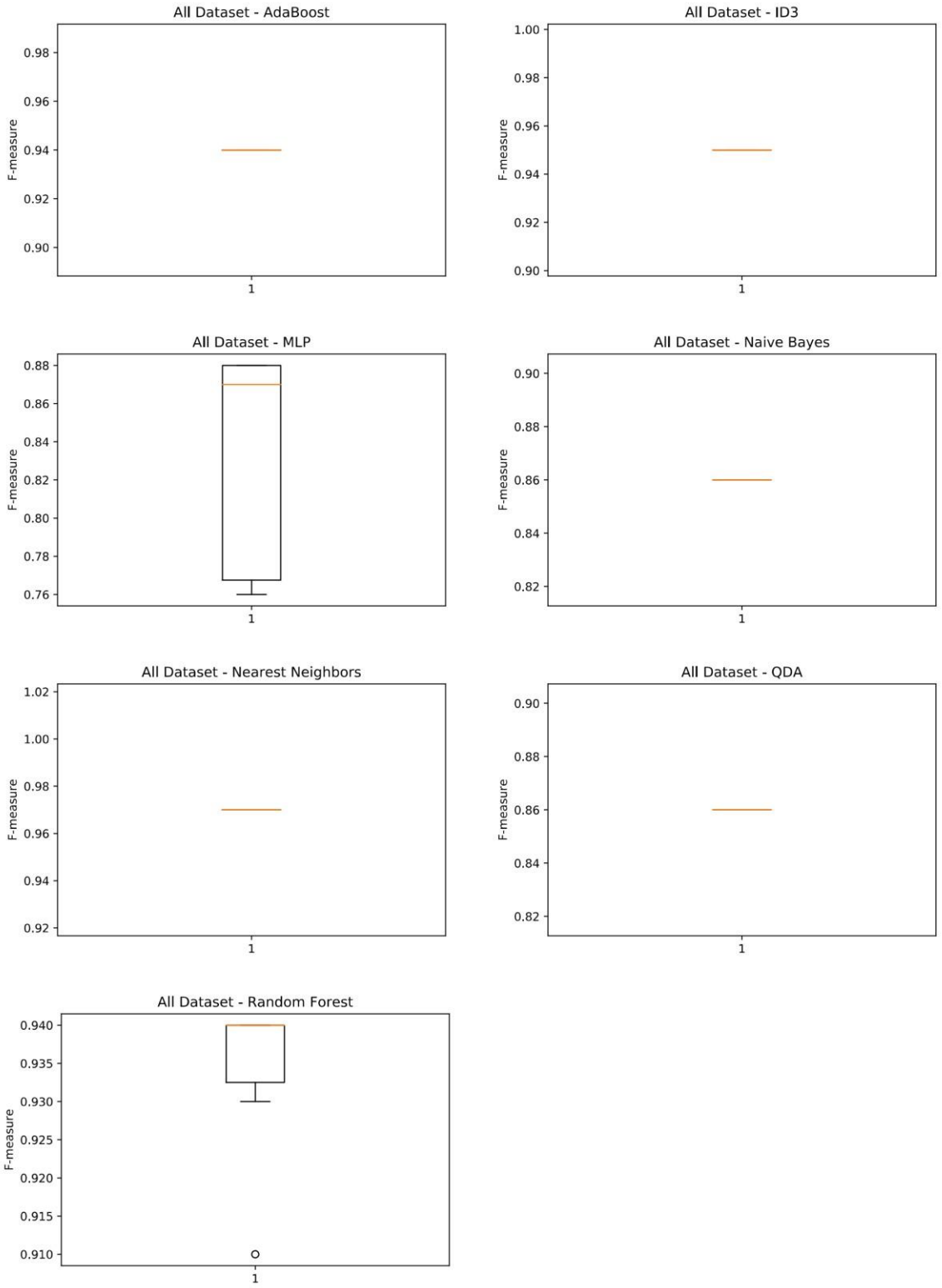
Appendix D. The Machine Learning Implementation Results (According to All Dataset)



Appendix D Figure 1. Method I - Machine Learning Implementation Box and Whisker Graph (F-Measure)



Appendix D Figure 2. Method II - Machine Learning Implementation Box and Whisker Graph (F-Measure)



Appendix D Figure 3. Final - Machine Learning Implementation Box and Whisker Graph (F-Measure)