IMPLEMENTATION OF FIREWALL RULES FOR INTRUSION DETECTION SYSTEM

HTAY HTAY YI

UNIVERSITY OF COMPUTER STUDIES, YANGON

JUNE, 2024

IMPLEMENTATION OF FIREWALL RULES FOR INTRUSION DETECTION SYSTEM

Htay Htay Yi

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

June, 2024

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

…..……………………………   ………….......…………………………

Date   Htay Htay Yi

ACKNOWLEDGEMENTS

First of all, I would like to thank the Minister, the Ministry of Science and Technology for full facilities support during the Ph.D. course at the University of Computer Studies, Yangon.

Secondly, I would like to express my deepest gratitude to Dr. Mie Mie Khin, Rector of the University of Computer Studies, Yangon, for allowing me to develop this thesis and providing general guidance during my study period.

I extend my special appreciation to Professor Dr. Nilar Aye and the Dean of the Ph.D. 3rd batch (Research Only) at the University of Computer Studies, Yangon, for their invaluable comments, advice, and insights.

I would like to express special thanks to my external examiner, Professor Dr. Aung Htein Maw, for useful comments and suggestions.

I am profoundly grateful to my supervisor, Professor Dr. Khaing Khaing Wai, at the University of Computer Studies, Yangon, for her excellent guidance, caring, and patient supervision and for providing me with outstanding ideas throughout the development of this thesis.

I am immensely grateful to my advisors and teachers, whose expertise and guidance have been invaluable. Special thanks to teacher Daw Aye Aye Khine, Professor, English Department, who took the time to meticulously review and correct the English in my thesis. Your feedback and suggestions significantly improved the clarity and quality of my work.

I extend my thanks to my colleagues and friends who provided helpful comments and encouragement. Your support has been essential in this journey.

Thank you all for your unwavering support and contributions.

ABSTRACT

Network security plays a pivotal role in safeguarding sensitive data from unauthorized access and malicious activities. This work addresses the challenge by proposing a Selected Features Based Intrusion Detection System (SFBIDS) that apply a firewall with an Intrusion Detection System (IDS). In the system, the firewall is a crucial part of network security and it applies especially in used software-based open source firewall that minimizes complication, time, often adaptable in their configuration, and mostly in cost. The filtering rules themselves might cause a security hole due to the complex nature of their configuration and the order of rules. If there are many firewall filtering rules, many policy anomalies can be caused in the desired network. In the SFBIDS system, twenty-seven firewall rules are manually created in the software-based firewall. An IDS typically operates using one of two primary methods: signature-based detection and anomaly-based detection. In the system employing a signature-based detection method, the approach focuses on identifying known threats by comparing network traffic or files against a database of known signatures. The SFBIDS is evaluated through the generation of a dataset comprising typical network traffic, as well as simulated Denial-of-Service (DoS) attacks and PortScan attacks. Feature selection is a critical component of intrusion detection systems, influencing their effectiveness in detecting malicious activities while minimizing false alarms. It presents a detailed analysis of two feature selection methods: Correlation-Based Feature Subset (CBFS) and Gain Ratio Feature Selection (GRFS), focusing on their efficacy in selecting the most relevant attributes for intrusion detection. Effective feature selection is critical for enhancing the performance of intrusion detection systems. The SFBIDS compare its performance with the widely used CICIDS 2017 dataset. The results demonstrate that by excluding flag features, the performance of intrusion detection algorithms improves significantly. It uses a technique for determining the minimum boundary value in the Correlation Attribute (CA) method by computing the average value from two datasets. It conducts a comparative analysis of attribute reduction in both the SFBIDS dataset and the CICIDS 2017. The SFBIDS system goal is to enhance the adequacy of performance by identifying and eliminating redundant attributes.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

Security issues are becoming more critical in network systems. Firewalls offer an important defense and protection for network, permit to strengthen security aspect. The system emphasizes the implementation of a secure network architecture. It involves strategically configuring devices, protocols, and security measures to mitigate vulnerabilities and potential threats. The system creates datasets to validate the effectiveness of intrusion detection. These datasets are crucial for testing and demonstrating the accuracy of the proposed security measures. The system surpasses the limitations of existing intrusion detection systems achieving over 99% accuracy.

In recent years, Intrusion Detection System (IDS) is becoming one of the essential technologies in any computer network organization. Intrusion Detection System can implement as a software-based or hardware-based. Hardware-based is more expensive and maintenance. A Network Intrusion Detection System (NIDS) is designed to detect malicious users and malicious traffic and then to monitor network traffic. The weakness of IDS is that it misses alert rate higher. One of the main problems with intrusion detection system is that they tend to generate a lot of false positive. A false positive occurs when the system generates an alert based on what it thinks is bad or suspicious activity but is actually normal traffic for that LAN. And then it can detect attacks, but is not preventing intrusions. Most network intrusion detection systems have large default databases of thousands of signatures of possible suspicious activities. When configuring firewall rules, it is crucial to ensure that the rules are ordered and typed. Misconfigurations can lead to vulnerabilities in the network [47]. By integrating firewalls, IDS/IPS, and machine learning-based detection into a cohesive network security system, organizations can achieve a robust and reliable defense against a wide range of threats. This integrated approach ensures that each component complements the others, enhancing the overall security posture and resilience [30].

A Denial of Service (DoS) or Distributed Denial of Service (DDoS) attack makes a network service unavailable to legitimate users by overwhelming the target with a flood of illegitimate requests or traffic. When applying machine learning classifiers and appropriate sampling techniques, researchers can develop robust models that effectively detect and mitigate DoS/DDoS attacks, enhancing network security and reliability. In this system, the implementation of a network testbed environment with software-based, open-source solutions can be a cost-effective and flexible approach for analyzing DoS traffic [3].

This work uses the small number of features that reduce the complexity time, and computer resources consumption as CPU and memory when the data analysis. Generating a new dataset for detecting different types of attacks and analyzing it using various machine learning algorithms involves several key steps. This process includes feature selection, dataset generation, and evaluation using machine learning models.

1.1 Research Problem

The research problem in the related to impact of redundancy features, long running time and use many unnecessary features in dataset [24]. When features interact with other features, their combined effect on the model's performance can be impacted by their individual effects. By systematically evaluating and incorporating feature interactions, it can improve model performance and derive more meaningful insights from their data. But, it is hard to get it right when the group features and calculates performance. Because studies show that one feature gets better when paired with another feature. Therefore, due to the lack of connected features, the group of features may not be correct just by not using the method. In a dataset, the number of features and instances that do not significantly affect performance increases the computation time.

The performance of IDS is influenced by the efficiency and effectiveness of the features used in machine learning models. Redundant Features are unnecessary and duplicate features in the dataset that do not contribute to detection accuracy but increase computational complexity. Long Running Time is for model training and detection due to many features leading to inefficiencies. Unnecessary Features in the dataset may not be relevant for the intrusion detection task, thereby increasing the dimensionality and computational load without improving detection performance. The focus of this work is to be a system with good network performance and security.

1.2 Motivation of the Research

The previous researchers [7], [58] have used various methods to reduce misconfiguration and rule inconsistency, but there are no consideration of the performance. When filtering the multiple firewall rules that cause rules inconsistency, and their network can be very complicated and network performance may be compromised by examining rules anomalies. The firewall and IDS are used in one of rulesets, using the machine learning classifiers to apply snort IDS. This system is designed to balance security and performance in the system on the tradeoff of both. Instead of isolation firewall and IDS, this work will implement them in conjunction with better network performance. The study leverages the CICIDS-2017 dataset, a comprehensive dataset for intrusion detection, to perform feature analysis using the Information Gain (IG) metric [24]. The most relevant features contribute to distinguishing between normal and malicious network activities. By applying the IG method, they rank the features based on their significance in anomaly detection, which assists in reducing the dimensionality of the dataset and improving the performance of machine learning models.

1.3 Objectives of the Research

The intend of this research is to implement the network topology with the secure policies that match the desire organization. The limitation of the firewall is combined with the strength of the IDS to build a secure LAN. The next objective is to choose a feature selection method to be a good feature. The another one is for attack detection, and instead of taking previous dataset and taking a processing time to build a dataset that matches the organization need. The last one is to compare with the existing dataset as CICIDS2017 and to show that the performance of the SFBIDS system is good.

The other objectives of this work are as follows:

- To select the correct values for the specified features and to define the classes
- To indicate the best set of features for detecting the certain attack categories
- To know the correctness of the features in the constructed dataset using the feature selection methods
- To evaluate the performance of the system with compare an existing Dataset CICIDS2017

1.4 Contributions of the Research

All the time of catching all the packets, the features and their attributes are considered based on the main destination. In main contribution, when extracting the capture packet, it is considered based on the destination address, regardless of the IP pair as source and destination address. The data extract from packet header and set attributes and values of features based on packet groups as SYN, ACK-SYN, RST and Retransmission.

The research work has several contributions as follows:

1) The system implement a small dataset with sixteen features relevant to DoS and PortScan attacks, and it helps the performance of IDS at a false negative rate and saves the computer resource consumption.
2) The performance of the system will be compared with the existing dataset CICIDS-2017.
3) In both datasets, it is proven that removing the flag feature does not affect the performance of the system.
4) The system applies Collection Based Feature Selection (CBFS), Gain Ration (GR), and Correlation Attribute (CA) methods to determine the quality of the features. In comparison with CICIDS2017, it uses these methods to prove the good features.

The SFBIDS system focuses on sixteen essential features and offers a streamlined and efficient approach to detecting DoS and Probe attacks. By leveraging machine learning techniques and integrating with existing network security tools, this system aims to reduce resource consumption, and improve overall network security.

The system presents network-based IDS that not only can efficiently detect but also can classify network data into three categories which are normal, DoS and PortScan.

1.5 Organization of the Research

The composition is arranged with seven chapters. This chapter includes an introduction, the motivation of the thesis, the problem statements, objectives, focuses and contribution of the research work. Chapter 2 observes the anomalies of firewall rules from the literature and related work. Good security is by combining firewall and IDS. The datasets were analyzed using machine learning classifiers and the detection rate and performance of the IDS were reported. Chapter 3 presents the theory part, Firewall and IDS rules, Machine learning classifiers, and attacks. A comparison of two feature selection methods is calculated to highlight the feature quality of the proposed system in Chapter 4. The implementation of the proposed system is represented in Chapter 5. Chapter 6 describes the evaluation of the experimental results by measuring the proposed dataset features and comparing them with CICIDS2017 to know good features by using two feature selection methods. Finally, Chapter 7 presents the conclusion extracted from this research works and depicts the future research lines.

CHAPTER 2
LITERATURE REVIEW AND RELATED WORK

The chapter categorizes the firewall rules misconfiguration, analysis and detection of the firewall rules anomalies, automatically detecting and resolving conflicts rules, Intrusion Detection System (IDS) and their rules, DoS/DDoS and Port Scan attacks that how to create dataset, CICIDS2017 dataset, machine learning classifiers, and its related research. The challenges and application will be discussed with respect to its literature review and related works.

2.1 Deployment of Firewalls

Firewalls are a cornerstone of network security architecture, a critical defense against unauthorized access and cyber threats. Firewalls inspect network traffic based on rules defined by administrators. These rules can specify which types of traffic are allowed or denied based on various criteria, such as IP addresses, port numbers, protocols, and content [17]. Firewalls protect against various threats, including unauthorized access, malware, viruses, worms, and other malicious activities. By filtering incoming and outgoing traffic, they help prevent unauthorized access to sensitive data and resources. Many operating systems come with built-in firewall functionality. For example, Windows includes Windows Firewall, macOS includes macOS Firewall, and Linux distributions often include iptables or firewall. These built-in firewalls can configure to meet the security requirements.

Firewalls use in both personal and enterprise settings. In personal use, they help protect individual computers or home networks from online threats. In enterprise environments, firewalls deploy to protect entire networks, including servers, workstations, and other network devices.

Proper configuration and regular maintenance of firewalls are essential to ensure they provide constructive protection. It includes defining and updating firewall rules, monitoring firewall logs for suspicious activity, and keeping firewall software up-to-date with security patches and updates. The types of firewalls are network based firewalls, host based firewalls, and application layer firewalls. Network-based firewalls are typically hardware or software-based devices that filter traffic at the network level. Host-based firewalls run on individual computers or devices and monitor traffic specific to that device. Application layer firewalls provide more granular control over network traffic.

Firewalls play a vital role in defending against malware infections by establishing a proactive barrier against malicious activities, controlling network access, and continuously monitoring and responding to security threats in real-time [20]. Firewalls are multi-functional security tools that serve various purposes in safeguarding systems and networks. The logging and audit functions of firewalls are essential for maintaining visibility, detecting security incidents, supporting incident response efforts, ensuring compliance, enforcing security policies, and enhancing overall network security posture.

Firewall logs are an essential component of network security. They offer administrators a detailed view of the traffic passing through the network perimeter, which is crucial for monitoring and analyzing potential security threats. By examining these logs, administrators can identify suspicious activities, track down the sources of attacks, and take appropriate measures to protect the network.

2.1.1 Firewall Security Policy

A firewall security policy is a comprehensive set of rules and configurations that govern the behavior of a firewall in managing network traffic. These policies are typically based on an organization's security requirements, compliance regulations, and risk management strategies. Firewall policies often specify which IP addresses or ranges permitted to communicate through the firewall. Exactly, firewall policies are foundational to network security. They define the rules and guidelines for managing inbound and outbound traffic, acting as a barrier between a trusted internal network and untrusted external networks, such as the internet. Policies often include provisions for logging firewall activity and generating alerts for suspicious or unauthorized traffic. Some firewalls can inspect traffic at the application layer and enforce policies based on specific applications or services (e.g., HTTP, FTP, SMTP).

If explicitly defining which services, protocols, and IP addresses are allowed through the firewall, organizations can maintain better control over their network traffic and enforce security policies more effectively. This strategy also enhances visibility into network activity, as administrators can more easily identify and monitor authorized traffic while flagging any attempts to access restricted resources. However, implementing a default-deny policy requires careful planning and ongoing maintenance to ensure that legitimate traffic is not inadvertently blocked. Organizations must review and update firewall rules regularly to accommodate changes in network requirements, application usage, and security threats. Additionally, thorough testing and validation are essential to verify that the firewall policy accurately reflects the organization's security objectives without disrupting legitimate business operations.

2.1.1.1 Default Security Policy and Matching Rules

In firewall configuration, the default policy typically occupies a fixed position at the end of the policy list. It ensures that put in last after all other specific rules can evaluated. Additionally, the order of security policies is often displayed based on their creation time, with the newest policies appearing at the top of the list and the default policy appearing last.

By maintaining the default policy at the end of the list, administrators can effectively enforce a default-deny approach, blocking all traffic is not explicitly allowed by preceding rules. It ensures that the default policy serves as a last line of defense that purpose to provide an additional layer of preservation against unauthorized access and potential security breaches. The default security policy serves as a fallback mechanism. If none of the manually created security policies match the criteria of incoming traffic (such as source IP, destination IP, protocol, port, etc.), the default security policy can use as a catch-all rule. By relying on the default security policy as a safety net, administrators can maintain control over their network traffic while allowing flexibility to create specific rules for different types of traffic and services. In Table 2.1, it can see the example rules of the firewall.

Table 2.1 Example of Firewall Rules

Rule	Proto:	Src_IP	Src_port	Dst_IP	Dst_port	Action
r1	UDP	192.168.235.50	any	192.168.137.100	53	allow
r2	TCP	192.168.235.*	any	192.168.137.*	443	deny
r3	TCP	192.168.235.*	any	192.168.137.100	22	deny
r4	TCP	192.168.235.50	any	192.168.137.100	443	allow
r5	TCP	*.*.*.*	any	192.168.137.*	22	deny
r6	ICMP	192.168.235.*	any	192.168.137.*	Ping	deny
r7	ICMP	192.168.235.50	any	192.168.*.*	Ping	allow
r8	UDP	*.*.*.*	any	*.*.*.*	53	deny

2.1.2 Issue of Firewall Policy

As networks grow in complexity and firewall rulesets expand, it becomes increasingly difficult to effectively manage and assess the associated risks, particularly concerning misconfigurations or overly permissive rules. By proactively addressing these challenges and adopting a holistic approach to firewall management, organizations can mitigate the risks associated with misconfigured or overly permissive firewall rules and enhance overall network security posture. As the number of rules increases, it becomes harder to maintain clear visibility into the purpose and function of each rule. Without adequate documentation or management practices, administrators may struggle to understand the rationale behind specific rules or identify redundant or outdated rules. Quantifying the risk introduced by misconfigured or overly permissive firewall rules can be challenging due to the interplay of technical, operational, and business factors. Traditional risk assessment methodologies may struggle to adequately capture the potential impact of firewall rule misconfigurations on overall security posture. By proactively addressing these challenges and adopting a holistic approach to firewall management, organizations can mitigate the risks associated with misconfigured or overly permissive firewall rules and enhance overall network security posture.

The techniques and algorithms outlined in the paper [6] aimed to address this challenge by offering two key capabilities: (1) Automatic Discovery of Firewall Policy Anomalies involves identifying rule conflicts and potential issues within existing firewall policies. Rule conflicts may arise due to overlapping rules, contradictory rules, or unintended consequences of rule ordering. By automatically detecting such anomalies, administrators can pinpoint areas of concern within their firewall configurations and take corrective action. (2) Once anomalies can be identified, the paper proposed techniques for editing firewall policies that avoids introducing new anomalies. It includes methods for safely inserting, removing, or modifying rules without disrupting the general security posture or inadvertently introducing new vulnerabilities.

Model checking techniques, such as Binary Decision Diagrams (BDDs), are powerful tools for analyzing the behavior of complex systems, including firewall policy configurations. In the context of firewall policy analysis, BDDs can be employed to represent the various rules and conditions in a compact and efficient manner, allowing for rapid evaluation of different scenarios and potential rule conflicts [19]. Overall, leveraging BDDs and model checking techniques for firewall policy analysis offers a systematic and automated approach to identifying and addressing potential security risks arising from misconfigurations or ambiguities within the policy. By providing a formal framework for policy analysis, these techniques help enhance the effectiveness and reliability of firewall configurations in ensuring network security. By constructing a BDD representation of the firewall policy, it becomes possible to systematically evaluate the behavior of the policy for different combinations of input parameters. It allows for automated detection of anomalies, such as conflicting rules or unreachable states within the policy.

Researchers often validate their algorithms using real-life firewall policies acquired from organizations or by generating synthetic firewall policies to simulate different scenarios. Validation against real-world data helps ensure that the algorithm performs effectively across a various use case and accurately identifies anomalies that may impact network security [14]. For the latest information on the fastest algorithms and advancements in firewall rule anomaly discovery and resolution, it recommends consulting recent academic publications and research papers in the field of network security and firewall policy analysis.

The work of Al-Shaer [6], [7] represented a significant contribution to the field of network security. Al-Shaer's objective is to tackle the difficulties associated with automatically identifying and resolving policy anomalies within both centralized and distributed legacy firewalls. Centralized and distributed legacy firewalls present unique challenges in terms of policy management. Centralized firewalls typically manage policies for an entire network from a single location, while distributed firewalls may have policies spread across multiple devices or locations. Al-Shaer's techniques likely account for these differences and provide solutions that apply effectively in both scenarios.

The rule merging model proposed by Zhang [58] likely leverages the concept of rule-service, which involves consolidating multiple firewall rules into a smaller set of more efficient rules. This approach reduced redundancy and complexity within firewall configurations, making them easier to manage and more effective at filtering network traffic. Furthermore, the results indicated that the optimized firewall model based on Zhang's rule merging technique achieves fewer filtering hits while processing the identical packets compares to traditional firewalls. It suggests that the optimized model is more efficient at filtering network traffic, resulting in improved performance and reduced resource consumption. Overall, this work represents the main contribution in network security by introducing a novel approach to firewall rule merging. The model enhanced the security posture of organizations while reducing the operational overhead associated with managing firewall configurations.

The manual definition of rules in firewall policies can often lead to conflicting, redundant, or overshadowed rules in policy anomalies. These anomalies can create vulnerabilities or inefficiencies in the network security posture, potentially exposing the network to security breaches or performance issues. Researchers have proposed optimization techniques to address policy anomalies by eliminating redundant rules, resolving conflicts, or reordering rules for better performance. It may involve merging overlapping rules, simplifying rule sets, or optimizing rule evaluation order. The simultaneous detection and resolution of policy anomalies offered by Abedin's algorithm is crucial for maintaining the integrity and effectiveness of the firewall policy. By automatically generating an anomaly-free rule set, the algorithm helps to ensure that the firewall can accurately and efficiently enforce security policies without unintended conflicts or vulnerabilities [29]. By introducing a novel algorithm for detecting and resolving firewall policy anomalies, Abedin provides a practical solution for organizations seeking to enhance the effectiveness and efficiency of their firewall configurations. The algorithm proposed by Abedin likely employs a combination of techniques to identify conflicting, redundant, or overshadowed rules within the firewall policy. The algorithm can determine the necessary reorder and split operations to resolve these anomalies effectively.

Hongxin Hu's framework and the (Firewall Anomaly Management Environment (FAME) tool offer a comprehensive solution for managing the anomalies of firewall policy. This work represents a valuable contribution to the area of network security management, providing practitioners with practical tools and techniques for maintaining secure and efficient firewall policies [18].

2.2 Firewall Based Intrusion Detection System

Where a firewall rule fails to block a sophisticated attack or a new form of malware, the IDS can potentially detect the intrusion based on abnormal behavior or signatures associated with the attack. The firewall provides proactive protection by blocking traffic based on predefined rules. It can prevent known threats from entering or leaving the network based on IP addresses, ports, and protocols. It includes detecting new or unknown threats, zero-day exploits, and insider attacks. When the IDS identifies such activities, it generates alerts for further investigation. The combine of a firewall and an IDS that provided a more comprehensive security posture for a network.

The author's analysis underscores the importance of adopting comprehensive security measures beyond traditional firewalls to adequately defend modern networks against a wide range of cyber threats. It involves leveraging multiple data sources and implementing security solutions that balance functionality, reliability, and cost-effectiveness [32]. Functionality refers to the effectiveness of the security solution in addressing the organization's specific security requirements and mitigating potential threats. Reliability pertains to the system's ability to perform its intended functions without interruption or failure. Cost-effectiveness considerations are that the chosen security solution offers value for money and aligns with the organization's budgetary constraints. The approach described by the author highlights the increasing complexity and sophistication required in modern network security strategies. By integrating multiple data sources, such as IDS alerts, SNMP events, and flow records, administrators can create dynamic firewall rules that adapt to real-time threats and network conditions.

Filip Hock's proposed behavior and principles of IDS/IPS systems likely include several concepts and considerations [16]. The choice between commercial and open-source IDS/IPS systems depends on factors such as budget, security requirements, technical expertise, and organizational preferences. Some organizations may opt for commercial solutions for their comprehensive features and support, while others may prefer the flexibility and cost-effectiveness of open-source alternatives.

The Network Defender proposed by N. Akhyari would aim to provide a cost-effective and efficient solution for network security, leveraging the power and flexibility of open-source applications to detect and respond to network attacks effectively [9].

Shah's [40] study would contribute valuable insights into the performance of Snort and Suricata as open-source IDS solutions for network security and help inform decision-making for organizations seeking to implement or enhance their intrusion detection capabilities. The objective of the work would be to assess the effectiveness and efficiency of Snort and Suricata in detecting various types of malicious traffic, including known signatures, anomalies, and emerging threats. The study would analyze the detection rates and accuracy of Snort and Suricata in identifying malicious traffic compared to ground truth labels. This analysis would highlight the strengths and weaknesses of each IDS in detecting different types of attacks. The performance comparison at 10 Gbps is as it assesses the scalability and efficiency of the IDSs under high network traffic loads. IDSs must be capable of processing and analyzing network traffic in real time without introducing significant latency or performance degradation. It is a significant undertaking, as the accuracy and efficiency of IDSs are crucial for effectively identifying and mitigating security threats in high-speed networks.

2.2.1 Intrusion Detection System

IDSs play a role of computer and network systems by continuously monitoring for suspicious or malicious activities, analyzing signals indicative of security threats, and promptly alerting relevant personnel or systems administrators when potential security incidents are detected [50].

IDSs primarily function as reactive rather than proactive security measures [33]. While they excel at detecting and alerting administrators to potential security incidents, they typically do not take direct preventive actions to stop attacks in progress. Instead, they serve as "watchdogs" or "informants," providing valuable information about ongoing or attempted security breaches. While IDSs serve as reactive agents by detecting and alerting administrators to security incidents, they also play proactive security measures and improve overall security resilience. Figure 2.1 likely illustrates the deployment of an IDS within a network environment, with IDS sensors strategically placed for internal and external traffic monitoring.

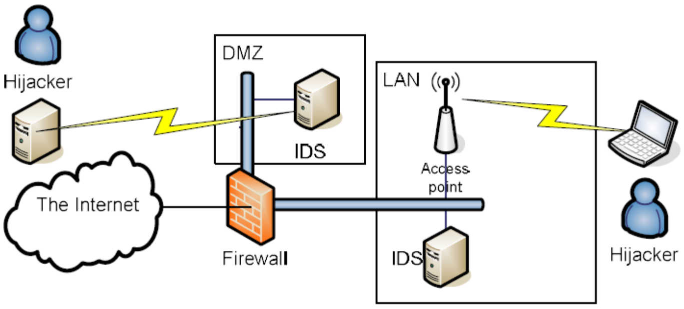


Figure 2.1 IDS in Internal and External Network

2.2.2 Classification Based IDS

Host-based IDS: A Host-based Intrusion Detection System (HIDS) is a security solution that activities on individual hosts or endpoints within a network. Unlike network-based intrusion detection systems (NIDS), which monitor network traffic for signs of suspicious activity, HIDS operates directly on individual hosts, allowing for more granular visibility and control over host-level events and behaviors.

HIDS provides granular visibility into the activities and behavior of each host, allowing security analysts to monitor and analyze events at the individual endpoint level. This level of detail enables the detection of host-specific threats and vulnerabilities that may not be apparent at the network level. HIDS continuously monitors host activities in real-time, analyzing incoming data streams from various sources to identify potential security incidents or anomalies. By detecting unauthorized or suspicious behavior as it occurs, HIDS helps organizations respond promptly to security threats and mitigate risks.

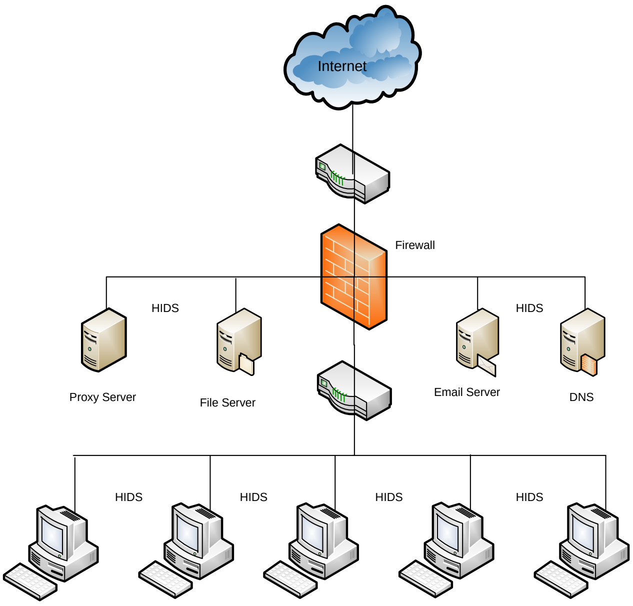


Figure 2.2 Host-Based Intrusion Detection System

Network-based IDS: A Network-based Intrusion Detection System (NIDS) is a security solution that specializes in monitoring and analyzing network traffic for indications of malicious activity or unauthorized access. NIDS compares observed network traffic against a database of pre-defined signatures or behavioral patterns associated with known security threats, such as malware, denial-of-service (DoS) attacks, intrusion attempts, or other suspicious activities. If a match is found, NIDS generates alerts or notifications to alert security personnel. NIDS helps organizations find and respond to security incidents by leveraging advanced detection techniques and comprehensive data collection [50].

NIDS typically operates in a passive mode, meaning it observes network traffic without actively interfering with it. It allows it to analyze traffic without introducing additional latency or disruption to network operations. NIDS examines individual packets of data as they travel across the network. It can analyze various attributes of these packets, such as source and destination address, protocol headers, payload contents, and patterns indicative of known attack signatures. If a equal is found, the system generates an alert to notify administrators of a potential security threat.

NIDS collects data from traffic passing through network segments, including packets transmitted over the internet. This data may include facts such as packet headers, payload content, session information, and traffic flow patterns. NIDS may be deployed at strategic points within a network infrastructure to provide comprehensive coverage and visibility into network traffic. Centralized monitoring allows security analysts to oversee network activity from a single console and respond promptly to detected threats. NIDS can generate alerts or notifications to prompt security personnel to investigate further. This proactive approach allows organizations to respond swiftly to security threats, mitigating potential damage and minimizing the risk of data breaches or network disruptions.

2.3 Machine Learning Based Intrusion Detection System

While firewalls and IDS contribute to network security, IDS offers a more advanced level of attack detection and can help with attacks that might slip past a traditional firewall's defenses. Machine Learning Techniques are commonly employed in IDS to enhance their ability to classify the normal and attack packets in the network.

When contrasting the NSL-KDD and UNSW-NB15 datasets with Monte Carlo simulation, Alhajjar and colleagues likely aimed to evaluate the performance of IDSs trained and tested on real-world datasets against synthetic data generated through a standard perturbation method [4]. This comparison provides real-world versus synthetic data for training and evaluating IDSs and help more robust intrusion detection techniques.

Alireza Osareh is known for his work in intrusion detection systems (IDSs), particularly in machine learning applications enhancing the effectiveness of IDSs. Conducting a study to compare the efficiency of machine learning methods, such as artificial neural networks (ANNs) and Support Vector Machines (SVMs), in the context of IDSs would align well with his expertise and research interests [33]. The Decision Trees gives better overall performance then SVM with 1998 DARPA Intrusion Detection dataset [2].

C.-F Tsai [44] presented a comprehensive review of the application of machine learning techniques to IDS. The primary goal of this review is to examine and understand the current status of using machine learning techniques to solve intrusion detection problems. Despite numerous studies in this area, there has not been a systematic review to consolidate the findings and methodologies used in these studies until this work. The review covered 55 studies published between 2000 and 2007. It focuses on single, hybrid, and ensemble classifiers for IDS. This paper [27] introduces an innovative approach to enhancing the performance of IDS using SVM combined with a feature selection strategy called the Gradually Feature Removal Method (GFRM). The Gradually Feature Removal Method optimizes feature selection to improve classification performance and reduce computational complexity. This work highlights the importance of feature selection in machine learning and its impact on the effectiveness of intrusion detection systems.

2.3.1 Appertain of Machine Learning Classifiers

SVMs are known for their effectiveness in multiclass classification tasks, making them a valuable component in the overall performance of IDSs that utilize machine learning algorithm. It has its strengths and weaknesses, and employing a variety of them can help improve the overall accuracy and robustness of the classification system [35]. The algorithm revolutionized the training of Support Vector Machines (SVMs). Le Cesie and Van Houwelingen [11] contributed to the development of Logistic Regression.

J48, also known as the C4.5 algorithm for decision tree learning [2]. It's for classification tasks in machine learning. It constructs a tree by recursively partitioning the data based on the attribute that provides the best split at each node, typically using measures like information gain or Gini impurity. Random Tree, on the other hand, constructs a decision tree by considering a random subset of attributes at each node. This randomness can help reduce overfitting and improve generalization. In machine learning tasks where a multiclass classifier is needed, and these decision tree algorithms can be employed. Multiclass classification involves predicting one of several possible classes for each instance.

JRip is a data mining designed by Cohen for classification tasks on accuracy. However, it's worth noting that JRip is a rule-based classifier, not a decision tree algorithm like J48 or Random Tree. It generates sets of rules to classify instances based on their attribute values. It's a common concern in machine learning models, including decision trees and rule-based classifiers like JRip. The k-fold cross-validation is a valuable tool for evaluating and comparing machine learning models, selecting hyperparameters, and assessing the robustness of a model's performance across different data subsets.

2.3.2 Overview of Existing Dataset

The KDD Cup 99 dataset, while well applied to evaluate IDS, has limitations that make it less representative of current network environments and attack trends [11]. It contains a predefined set of attack types, but it may not cover all the types of attacks that are prevalent in modern networks. These features include various attributes such as protocol type, service, duration, number of failed login attempts, etc. [45]. Moreover, the attacks included in the dataset might not accurately represent the techniques used by contemporary attackers. Some attack instances were artificially generated, which may not accurately reflect real-world attack scenarios. There is redundancy in the dataset, with multiple instances of the same attack pattern.

The Kyoto 2006+ dataset is an extension and improvement upon the original Kyoto 2006 dataset, designed to evaluate NIDSs. The dataset was developed by Kyoto University's Honeypot Project and provides a more diverse and realistic set of network traffic data for intrusion detection research.

The dataset consists of various features extracted from network traffic, and it contains a total of 24 features [22]. These features collectively provide valuable information about network traffic patterns, communication protocols, and potential indicators of malicious activity, which can be utilized by intrusion detection algorithms to distinguish between normal and abnormal behavior [43], [38]. Additionally, ten more features are included in the analysis of NIDSs. These features have been identified as valuable in the effectiveness of performance or robustness in IDS based on the specific characteristics of the Kyoto 2006+ dataset or other considerations. Fourteen features are selected based on the KDD Cup 1999 dataset. These features are likely chosen because of their relevance and importance in intrusion detection, as identified by previous research and analysis.

The NSL-KDD dataset is an improvement over the original KDD Cup 1999 dataset in several aspects. It retains the same 41 features but addresses some of the limitations of the original dataset, particularly in terms of data balance and redundancy. By removing duplicates and redundant records, NSL-KDD aims to provide a more realistic representation of network traffic and intrusions [22], [24]. It is done to create a more balanced and representative dataset for evaluating intrusion detection systems (IDS).

The CICIDS-2017 is a comprehensive dataset that used for detection system, offering a rich source of network traffic data for analyzing and developing anomaly detection algorithms. It contains significant network traffic data, providing researchers with substantial information and models. With 78 features for each observation, the dataset offers diverse variables that capture various aspects of network behavior. These features may include attributes such as packet size, protocol type, source and destination IP addresses, and more [24] [36].

The Kyoto 2006+ dataset is another valuable resource in intrusion detection, particularly for network intrusion detection systems (NIDS). The dataset includes 24 features, with 14 initially selected based on the KDD Cup 1999 dataset. These features likely capture aspects of network traffic behavior relevant to intrusion detection [54]. The ten additional features were included in the analysis of NIDSs, possibly to address specific requirements or challenges encountered in intrusion detection research. The Kyoto 2006+ dataset consists of two main types of traffic: benign (normal) and attack traffic [43] [1].

The NSL-KDD dataset, derived from the KDD Cup 1999 dataset, is a popular benchmark dataset used in intrusion detection research. It is to address some of the limitations of the original KDD Cup 1999 dataset, including redundant records and unrealistic traffic patterns [54] [24]. The NSL-KDD dataset is composed of five main classes, representing different types of network traffic: Normal, Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L). Each class corresponds to a specific type of network activity, with DoS attacks targeting the availability of resources, Probe attacks attempting to gather information about the target system, and U2R and R2L attacks focusing on unauthorized access to the system [36] [38].

The CICIDS-2017 dataset is indeed a widely used benchmark dataset in the field of intrusion detection. It contains network traffic data, with 78 features recorded for each observation. These features covered network traffic behavior, providing a rich source of information for anomaly detection algorithms [54]. The dataset consists of two main classes of traffic: benign traffic and different types of attacks. The attacks in the dataset are diverse and representative of real-world cybersecurity threats, making the dataset suitable for estimating the performance of IDS [24], [43]. With seven distinct attack types, including Brute force, Portscan, Botnet, DOS (Denial of service), DDoS (Distributed Denial of Service), Web, and Infiltration, the dataset covers a wide range of common threats.

2.3.3 Useful of Existing Datasets with Machine Learning

The SVM compared with the decision tree performance on the KDD Cup 99. Using decision trees as binary classifiers and employing multiple classifiers for multi-class classification is a common strategy in machine learning and classification tasks. The Decision Tree gives better performance the SVM, that Decision Tree worked in the small training data [2]. Multiclass classification is a superior for Intrusion Detection. The SVM is not capable of Multiclass which work with Decision Tree. K. Kumar's proposal involves enhancing the performance of Naive Bayes classifiers for intrusion detection in IDS [23]. The study compares the three different classifiers—Naive Bayes, J48, and REPTree—using 10-fold cross-validation. Enhancing the performance of Naive Bayes classifiers can have significant implications in intrusion detection, particularly in improving accuracy and reducing false alarms. By addressing these challenges, Kumar's work could potentially lead to more reliable and efficient systems for detecting and mitigating attacks, ultimately bolstering cybersecurity efforts. It would be interesting to learn more about the specific techniques or methodologies.

The research [22] addresses the need for updated datasets that reflect the current threat landscape, as older datasets like KDD Cup 99 are modern network environments. By providing a more accurate and comprehensive dataset, the paper contributes to developing more effective NIDS, which are crucial for protecting networks against cyber threats. The statistical analysis presented in the paper offers important insights into the nature of modern network attacks, supporting the development of more robust and effective intrusion detection systems. Arash Habibi Lashkari's work on characterizing Tor traffic using time-based features provides valuable insights into distinguishing anonymized traffic from regular traffic [25]. The effective use of machine learning techniques and the identification of significant time-based features contribute to advancements in network traffic analysis and security.

2.3.4 Detection with IDS

In the work, Benedetto Marco Serinelli [39] presented the challenge of zero-day anomaly detection in the context of IDS. Zero-day attacks refer to previously unknown vulnerabilities before the vendor has issued a patch or solution, making them particularly difficult to detect using traditional signature-based methods. This study explores a platform designed to detect such zero-day anomalies. Three DoS Attacks aim to disrupt the normal functioning of network service by overwhelming it with a flood of illegitimate traffic. Three Scanning Attacks involve probing the network to identify open ports and services for further attacks. The results indicate a misclassification prediction error, which suggests that the system incorrectly identifies some of the anomalies. This misclassification error poses a significant challenge as it inhibits the application of automatic attack responses. In other words, the system's inability to accurately classify these zero-day attacks prevents it from taking appropriate automated actions to mitigate the threats.

In the study conducted by Kinam Park, the focus was on evaluating the effectiveness of the Random Forest algorithm for intrusion detection by analyzing its performance on datasets derived from the Kyoto 2006+ dataset [35]. This dataset is notable for being a comprehensive and recent collection of network packet data to develop and test IDS. While Random Forest performs well, there are inherent challenges in handling imbalanced data (where the number of packets significantly outweighs the number of malicious packets), which can affect the detection performance. The study also notes the computational complexity and resource requirements for training and deploying Random Forest models, especially with large-scale network data.

2.4 Summary

This chapter encompasses a wide range of topics in network security, aiming to address critical challenges in firewall management, intrusion detection, and attack detection through the development of methods and techniques informed by literature review and related works. Additionally, the methods and techniques will be evaluated in real-world applications to assess their practical feasibility and effectiveness in improving network security. This research aims to develop methods for detecting DoS attacks, and Portscan attacks, which are common threats to network security. It may involve developing machine learning classifiers trained on labeled datasets to distinguish between legitimate and malicious traffic.

CHAPTER 3
THEORETICAL BACKGROUND

In the realm of cybersecurity, protecting network infrastructure is paramount. Two critical components in this chapter are Firewalls and Intrusion Detection Systems (IDS). This chapter explores the theoretical foundations of these technologies, their importance, and the synergistic benefits of combining them to bolster network security. A firewall is a critical component of network security that acts as a barrier between an internal network (such as a corporate network or home network) and an external network (such as the Internet). IDS is to detect and respond to potential security breaches. Unlike firewalls, IDS analyzes the behavior of network traffic and system activities to identify signs of malicious activity.

While firewalls and IDS are powerful tools individually, their combination offers a robust security solution that addresses the limitations of each. Firewalls block known threats and control access, while IDS detect sophisticated and internal threats.

The second part of this chapter delves into the theory behind Machine Learning (ML), focusing on classifiers and feature selection methods. Machine learning has become a critical tool in enhancing security measures by improving the detection of anomalies and sophisticated threats. Feature selection methods are the selection of a subset of related features for building robust machine learning models. Effective feature selection improves model performance, reduces overfitting, and decreases training time.

3.1 Introduction of Firewall

A firewall security policy is a set of rules or criteria that dictate how a firewall handles incoming and outgoing network traffic. Each rule consists of several filtering fields, also called network fields, and an action field. These rules are organized into an order, specifying the actions on packets that match certain criteria [12]. Firewalls are used for network security, providing essential protection by filtering traffic and enforcing security policies. They can be implemented as hardware or software solutions, each offering specific advantages depending on the network environment. When defining and managing firewall security policies, organizations can control access, manage traffic, and enhance their overall security posture. The integration of advanced features further strengthens the capability of firewalls to protect against sophisticated cyber threats, making them a crucial element in any comprehensive cybersecurity strategy.

A firewall security policy is a list of ordered filtering rules that can define the actions performed on matching packets. Each rule consists of several filtering fields called network fields, and an action field. A firewall policy is included of a sequence of rules, where each rule specifies certain criteria for matching packets and the corresponding action (e.g., allow or deny). These rules are evaluated in the order they are listed in the policy. A packet is matched a rule if all these fields in the packet's header satisfy the conditions specified in the rule [18].

Firewalls typically use a first-match semantic to evaluate packets against the rules. A packet matches a rule, the associated decision (action) for that rule applies, and no further rules are evaluated for that packet. It ensures that the first applicable rule dictates the action taken. Firewalls are important in network security, but using a large number of firewall rules can make more complex and error prone, especially in enterprise networks. When writing the firewall rules, the administrator needs to be careful when modifying or adding rules. The firewall anomalies depend on the order of the currently written rule and the other rules. The firewall rules misconfiguration and admin typing errors that can cause network vulnerabilities. So, the previous work presents techniques and algorithms that discover firewall policy anomalies to reduce vulnerability [8], [34].

The firewall runs the rule matching algorithm based-on rule service to resolve the conflict segment [58]. When a firewall receives a packet, it checks whether it matches the predetermined rule and sequence. The more network traffic, the longer the firewall filter time, the more it effects on network performance. When a packet comes in, the firewall takes time to check whether it matches multiple rules, which reduces the performance of the firewall. Although the large networks use various algorithm to reduce filtering time and number of anomaly rules, they actually affect firewall performance. Because there are many limitations in the firewall, it is not possible to check the same packets in all incoming packets. Since it is not possible to prevent same packets or internal packet with malicious code and cannot fix administrative mistakes or poorly designed security policies.

The weakness of the firewall is: 1) not being able to protect against attacks from the intranet, 2) its access control policies cannot be changed depending on the type of the attacks coming in from the outside, and 3) the firewall also cannot protect against virus [59] combined with IDS in the local system. The instantly growing number of security threats on both Internet and intranet networks hugely inquire reliable security solutions. Therefore, IDS use to defend network infrastructure by detecting attacks and malicious activities. The system is evaluated by using benchmark datasets to performance of a detection rate.

3.2 IPCop Firewall

IPCop is an open-source known for its secure and stable performance. As a GNU/GPL project, IPCop offers a feature-rich standalone firewall solution to the internet community. It stands out with its comprehensive web interface, well-documented administration guides, and active user and administrative mailing lists, making it accessible to users of all technical capacities [53]. By going beyond basic firewall functionalities and offering features that rival commercial solutions, IPCop provides a secure, stable, and cost-effective option for protecting networks in various environments.

IPCop's base features provide a solid foundation for network security and management. However, what truly sets IPCop apart is its extensive range of add-ons and optional plugins, which allow users to expand and customize the functionality of their firewall according to their specific needs. These add-ons cover a spectrum of capabilities, ranging from web filtering to antivirus scanning [15].

IPCop leverages the Linux Netfilter or IPTables firewall facility to implement a stateful firewall. Stateful firewalls, such as those built by Netfilter/IPTables, maintain a comprehensive record of connections to and from all network interfaces, including GREEN, BLUE, and ORANGE network IP addresses. These firewalls keep track of connection states by monitoring various attributes [52]. Indeed, IPCop offers an impressive array of base install features making it a comprehensive and versatile firewall solution for various network environments.

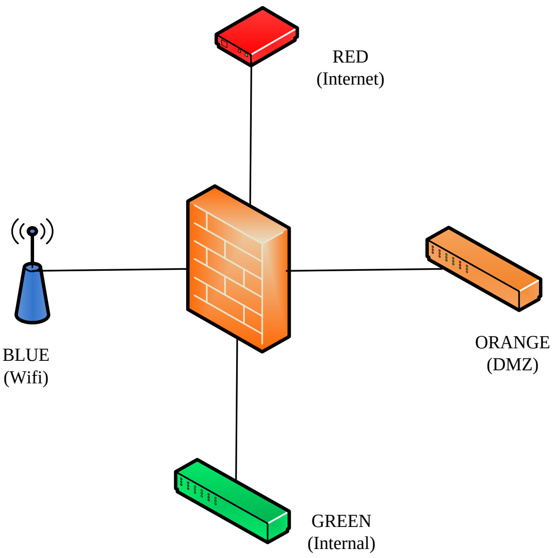


Figure 3.1 IPCop Firewall Interfaces

In Figure 3.1, the decision to implement a software-based firewall like IPCop instead of a traditional hardware-based firewall brings several advantages, particularly in terms of flexibility, cost-effectiveness, and adaptability to the network environment. The four types of network interfaces commonly used in IPCop and similar firewall solutions are Green, Red, Blue, and Orange. Each network interface is associated with a different trust level, with the firewall enforcing appropriate security policies and access controls based on these trust levels. The Green interface typically represents the trusted internal network, while the Red interface connects to the untrusted external network (e.g., the internet). The Blue and Orange interfaces may represent additional internal networks or segmented network zones with varying levels of trust.

The IPCop web interface provides a comprehensive platform for configuring and managing network security and services, with dedicated interfaces for controlling outgoing traffic, managing firewall access, segmenting internal network traffic, configuring port forwarding, and controlling external access to the firewall [42]. By leveraging the features and capabilities of the IPCop web interface, administrators can effectively secure and manage their network infrastructure while providing necessary access to authorized users and services.

3.3 Intrusion Detection System (IDS)

It is deployed on networks to monitor traffic and detect suspicious behavior [55]. There are two main types of IDS: Signature-Based Intrusion Detection System (Signature-Based IDS) and Anomaly-Based Intrusion Detection System (Anomaly-Based IDS). A signature-based intrusion detection system is a powerful tool for identifying and mitigating known threats based on specific attack patterns. By continuously monitoring network traffic and comparing it to a comprehensive one, it can quickly and accurately alert administrators to potential security breaches. However, its effectiveness is limited to the scope of its signature database, making it essential to keep the database updated with the latest threat information to maintain robust network security.

An anomaly-based IDS is a type of IDS that relies on statistical monitoring to detect unusual activity within a network. An anomaly-based IDS continuously collects and analyzes data on network traffic to establish a baseline of activity. This baseline includes various parameters such as packet size, frequency, protocol usage, and typical user behavior patterns. The system monitors ongoing network traffic in real time. It uses statistical methods to compare current traffic against the historical baseline. Any significant deviation from the baseline is a potential anomaly.

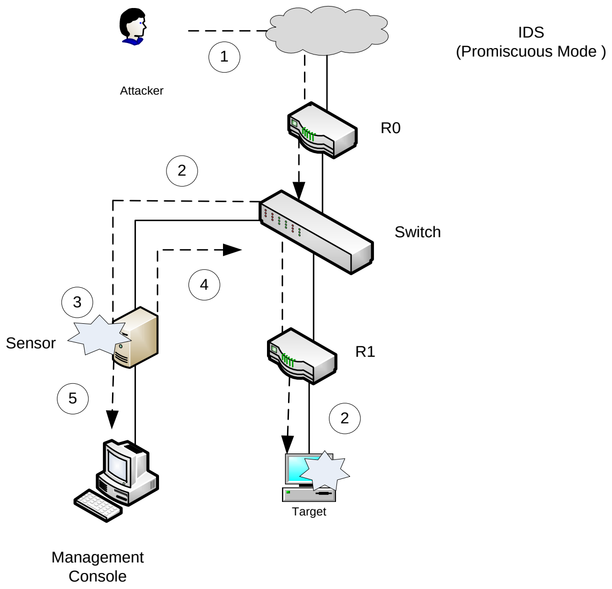


Figure 3.2 Operation of IDS

Figure 3.2 shows an attack on a network. An IDS sensor is in the network in IDS mode, it is to monitor and analyze traffic for signs of malicious activity. The IDS sensor inspects the copied packets in real-time as they are from the switch. The sensor uses a database of known attack signatures to compare and identify potentially malicious traffic. When the sensor detects traffic that matches a known attack signature, it identifies this traffic as malicious. Signature-based detection relies on pre-defined patterns of known threats, making it for recognizing well-documented attacks. The IDS sensor generates an alarm and sends it to a central management console [28].

IDS continuously monitor network traffic, examining the data packets that pass through the network. It can be monitored in the network, such as at the gateway, within the internal network, or at the host level (specific to individual devices). When network traffic matches one of these predefined rules, the IDS identifies it as a potential security threat. The matching criteria can vary from simple pattern matching to more complex behavioral analysis. The generated alert is reported to the relevant parties, such as network administrators, security personnel, or users, depending on the configuration of the IDS. This reporting allows for immediate investigation and response to potential threats.

IDS can roughly divide into three steps: (1) monitoring the network and log files, (2) comparing with the signature and with statistical data, (3) saving event and session logs and notification by console, mail, etc. An IDS receives raw inputs from sensors. It saves those inputs, analyzes them, and takes some controlling action [55].

3.3.1 Network-Based Intrusion Detection System

A Network-Based Intrusion Detection System (NIDS) is a crucial component for comprehensive network security. To monitor and analyze of network traffic, especially at the application protocol level, NIDS provides an essential layer of defense against threats that may evade firewalls. Properly configured NIDS can enhance security by flagging potentially dangerous packets, verifying firewall rules, and providing additional protection for application servers, ensuring a more robust and secure network environment.

In Figure 3.3, A firewall protects from external attacks by controlling inbound and outbound network traffic. However, once an attacker gains access to the local network. This is because firewalls primarily monitor and filter traffic that passes through them from the external network, leaving them blind to internal LAN activity. To address this gap, a NIDS complements firewalls to monitoring and analyzing network traffic within the local network.

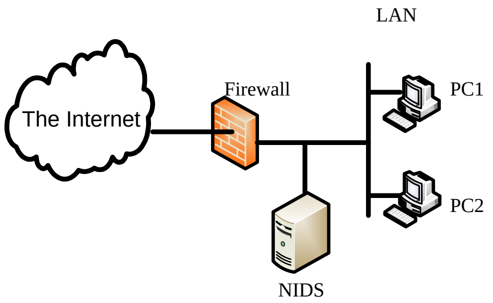


Figure 3.3 NIDS and Firewall Protection

3.3.2 Host-based Intrusion Detection System

Host-based intrusion detection systems provide a critical layer of security by focusing on specific behaviors and activities within individual hosts, complementing network-based IDS to offer a more comprehensive defense against intrusions. It includes keeping track of system calls, file system modifications, application activity, and changes to system binaries. Their ability to detect unauthorized changes and process activities makes them crucial components of a robust security strategy, enabling organizations to swiftly respond to potential threats and maintain system security. The characteristics of HIDS are international monitoring, network traffic analysis, security policy enforcement, attack outcome awareness.

International Monitoring: Monitors the internal state of a host system, including files, processes, and system calls. Detecting changes to the system’s state that may indicate a security breach, such as unauthorized modifications to critical system files or suspicious process activity.

Network Traffic Analysis: Analyzes network packets on the host’s network interfaces. Traffic to and from the specific host where the HIDS is installed.

Security Policy Enforcement: Monitors compliance with the system's security policy. Ensures that system operations adhere to defined security policies, identifying and reporting deviations.

Attack Outcome Awareness: Observe the results of attempted attacks directly. Since HIDSs have direct access to system files and processes, they can determine the effectiveness of an attack and its impact on the system.

3.4 Snort IDS

Snort is a widely used free and open-source network intrusion detection and prevention system (IDS/IPS) created by Martin Roesch in 1998. Developed by Sourcefire, which was later acquired by Cisco, Snort is known for its robust capabilities in protocol analysis, content searching, and matching to identify and mitigate potential threats in network traffic [56]. Snort's versatility in operating in different modes—sniffer, packet logger, and network intrusion detection—allows it to be used effectively for network security tasks.

Sniffer mode: Sniffer mode enables Snort to read network packets and display their contents on the console in real-time. It is primarily used for monitoring network traffic in a detailed and human-readable format.

Packet Logger Mode: Snort logs the packets to disk. This mode is useful for recording network traffic for later analysis.

Network Intrusion Detection System Mode: Snort monitors network traffic and analyzes it against a set of user-defined rules. Based on the analysis, Snort can generate alerts or take predefined actions.

Whether it is for real-time traffic observation, logging traffic for later review, or actively monitoring and defending against network threats, Snort provides robust tools to enhance network security and visibility.

In addition, Snort's powerful signature-based rule engine and extensibility through plug-ins and preprocessors make it an effective and flexible tool for network intrusion detection [54], [41].

3.4.1 Basics IDS Rule

IDS likes Snort use user-defined rules to monitor and analyze, especially malicious network traffic. It is an open-source IDS and utilizes rules in two fundamental parts: the first one is the rule header, and the second one is rule options. The general form of a Snort rule is structured to specify the action to take, the protocol to analyze, and the source and destination network parameters, followed by options that provide additional conditions and descriptions.

This is the detailed structure:

```
action proto src_ip src_port direction dst_ip dst_post (option)
```

Actions: This field provides a few built-in actions that it can use when crafting rules. These actions determine how Snort responds when a packet matches a rule.

Protocols: In Snort rules, this is to define which network protocol the rule applies and ensures is only applied to packets using the specified protocol.

IP addresses: It follows the action and protocol are to specify not only the source address but also destination IP addresses and ports. These fields help the specific traffic that the rule will match.

Ports: In Snort rules, specifying ports is essential for targeting specific network traffic. Just like IP addresses, Snort allows the use of single ports or ranges.

Options: It allows to specify additional attributes to check against when a rule is triggered

The sample of Snort rule is

```
alert udp 192.168.56.99 any -> $HOME_NET 53 (msg: “Traffic from 192.168.56.99”;
sid = 1000002; rev = 1;)
```

In Ensure that $HOME_NET is in their Snort configuration files to represent the range of IP addresses that belong to their internal network. Additionally, adjust the **sid** and **rev** values as needed to maintain uniqueness and track revisions of their rule.

3.5 Introducing of Machine Learning

Machine learning is studied to develop computer algorithms that enhance their performance through experience and the employment of data [10]. Machine Learning is a subset of artificial intelligence (AI) that concentrates on developing algorithms and statistical models that allow computers to perform specific tasks by learning from data, recognizing patterns, and making decisions with minimal human intervention [48]. As data availability and computational power grow, machine learning will play an increasingly critical role in solving complex problems and advancing technology.

Machine learning algorithms construct models from sample data, referred to as "training data," to make predictions or decisions without being explicitly programmed to perform those specific tasks. It allows an algorithm to acquire from the data and refine its performance over time. Machine learning algorithms are valuable in applications where developing conventional algorithms to perform the necessary tasks is difficult or unfeasible. Here are some key application domains where machine learning is making an impact: medicine, email filtering, speech recognition, computer vision, and additional applications [59]. Machine learning algorithms leverage historical data as input to predict new output values, making it a powerful tool for businesses to gain competitive advantages. When learning from past data, these algorithms can be identified patterns, making predictions, and automated decision-making processes, thus driving efficiency and innovation [60].

3.5.1. Types of Machine Learning Algorithms

Classical machine learning approaches are fundamentally categorized based on how an algorithm learns from data to improve its predictions. The four basic approaches are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Data scientists choose an approach based on the data and the forecast of the task [60].

Supervised learning: Supervised learning is a type of machine learning where a model uses a labeled dataset. Each training example in the dataset consists of an input-output pair, where the input is the data and the output is the correct result or label. Training Data consists of pairs of input data and corresponding correct outputs. An algorithm or mathematical construct that makes predictions based on input data. Loss Function measures how well the model's predictions match the actual outputs in the training data. The optimization Algorithm adjusts the model parameters to minimize the loss function.

Unsupervised learning: Unsupervised learning is a powerful help in discovering patterns and structures in data without labeled responses. Algorithms identified that patterns or structures within the data. Unlike supervised learning, evaluation in unsupervised learning can be more challenging. It is for clustering, dimensionality reduction, and association rule mining, among other applications.

Semi-supervised learning: Semi-supervised learning is an approach in machine learning that involves training a model using both a small amount of labeled data and a large amount of unlabeled data. This method leverages the unlabeled data to improve the model's performance, especially when acquiring labeled data is expensive or time-consuming. A small subset of the data has the correct output labels. The large subset of the data lacks annotations. A machine learning algorithm leverages labeled and unlabeled data to improve learning accuracy. It is beneficial when labeling data is expensive or time-consuming, yet a small labeled dataset is available to guide the learning process. It is used in image and text classification. Examples: Methods that combine supervised and unsupervised techniques, such as semi-supervised SVMs and certain types of neural networks. Semi-supervised learning strikes a balance when labeled data is scarce but abundant unlabeled data is available.

Reinforcement learning: Reinforcement learning (RL) proceeds towards machine learning where an agent learns to determine to perform actions in an environment to maximize some notion of cumulative reward. The agent interacts with the environment, receives feedback rewards or penalties, and uses this feedback to improve its future actions. This approach is used a sequence of actions, such as game playing (e.g., chess, Go), robotic control, and certain types of recommendation systems.

Each approach has its strengths and is tailored to specific types of predictive modeling tasks, guiding data scientists in selecting the most appropriate method for their particular application.

3.5.2 Machine Learning Classifiers

Machine learning classifiers are algorithms designed to categorize data into predefined classes. In security, these classifiers are applied to identify various types of network traffic, distinguishing between normal and malicious activities. Machine learning algorithms are the mechanism of machine learning datasets into models that can make predictions or decisions. The effectiveness of an algorithm depends on the problem there solving, the computing resources available, and the nature of the data [13]. Supervised learning algorithms like logistic regression and decision trees work well with structured data (e.g., tabular data with clear feature columns). Algorithms capable of handling unstructured data, such as convolutional neural networks (CNNs) for images and recurrent neural networks (RNNs) for text, are more appropriate.

Machine learning algorithms can be complex and nuanced, often going beyond the constraints of fitting data to specific mathematical functions, such as polynomials, as in traditional nonlinear regression. This flexibility allows machine learning to handle a wider variety of problems. Two major categories frequently addressed by machine learning are regression and classification.

Regression problems involve predicting continuous numeric values. The goal is to model the relationship between input features and a continuous output variable. Examples include: Predicting House Prices, Estimating Income. Classification problems involve predicting categorical, non-numeric outcomes. The examples of Regression include: Email Spam Detection, Image Recognition. Regression algorithms are best for predicting continuous numeric values, while classification algorithms are suited for categorizing data into discrete classes. Each type of problem requires different approaches and considerations to build effective models.

3.5.2.1 Logistic Regression

Logistic regression is used for binary classification tasks due to its simplicity, interpretability, and efficiency. It is a baseline model for classification tasks and a building block for more complex models. Logistic regression can help identify significant features or predictors in the dataset during the exploratory data analysis (EDA) phase. The features can then be used for further analysis or modeling [26]. Logistic regression is an interpretable algorithm for binary classification tasks. Its foundation lies in the logistic function, which effectively models the probability of an instance belonging to a particular class. With extensions for multinomial and ordinal outcomes, logistic regression remains a versatile machine learning toolkit. Its implementations in libraries like Scikit-learn and Statsmodels in Python, and the function in R, make it accessible for practical use in a wide range of applications.

3.5.2.2 Support Vector Machine

SVMs operate by representing the training data as points in a high-dimensional space, where each point belongs to one of two categories (binary classification) [53]. One limitation of SVMs is that they do not provide direct probability estimates for classification outcomes. Instead, SVMs provide binary decisions based on the side of the hyperplane on which a point falls. However, techniques like Platt scaling to approximate probabilities from SVM outputs. Overall, SVMs are powerful and versatile classifiers known for high-dimensional data and effectively separating complex classes. While they may not directly provide probability estimates, their robustness and efficiency make them popular in various machine learning applications. The common algorithms are Linear Regression, Logistic Regression, Decision Tree, SVMs, Neural Networks.

3.5.2.3 Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to improve the overall performance and robustness of the model. Instead of relying on a single decision tree, Random Forest constructs a decision tree during the training phase [61].

Random Forest employs two types of random sampling: Bootstrapped Sampling, and Feature Randomization. **Bootstrapped Sampling:** Each decision tree is applied to a bootstrap sample of the original training data, that involves sampling with replacement. It means that some instances may be sampled multiple times. Feature **Randomization:** At each split in the decision tree, only a random subset of features is for splitting. The analogy of a random forest "decorating the trees" and ensuring that each tree focuses on different aspects of the data is a creative way to describe how the random forest algorithm works. Random Forests are highly versatile and widely used across various domains due to their robustness, ease of use, and effectiveness in handling high-dimensional datasets. They are useful when dealing with complex datasets with noise, missing values, or outliers. Random Forests remain a popular and widely used machine learning algorithm due to their excellent performance across a wide range of applications and datasets. Advances in parallel processing and optimization techniques have also helped mitigate some of the computational challenges associated with Random Forests.

3.5.2.4 Decision Tree

Sequential Covering is particularly useful to induce a set of interpretable rules from data, as each rule corresponds to a specific condition on the input features that leads to a particular outcome [63]. It often trades off interpretability for predictive accuracy. It models determination and their probable consequences, including chance event outcomes, resource costs, and utility. The tree structure includes nodes representing decisions, chance events, or end states, and branches representing the outcomes of those decisions or events. Constructing a decision tree involves selecting the best feature to split the data at each node and continuing this process recursively for each subset of data. To address some of these disadvantages, techniques like pruning, which involves removing parts of the tree that provide little predictive power, can help prevent overfitting and simplify the tree structure. Additionally, decision trees are inherently interpretable, making them useful for explaining the reasoning behind classification decisions to non-experts.

3.5.2.5 Naïve Bayes

Naive Bayes is a family of simple and effective probabilistic classifiers based on applying Bayes' theorem with a strong (naive) assumption of independence between the features. Despite the simplicity of the assumption, Naive Bayes classifiers often perform surprisingly well and are widely used in various applications, particularly in text classification [51]. Naive Bayes is a foundational and practical algorithm in machine learning, particularly useful for classification tasks like spam detection, sentiment analysis, and document categorization. Despite its simplicity and the naive independence assumption, it often yields good performance and is a strong baseline for many problems. However, despite its ability to perform well with small datasets, Naive Bayes may not achieve the same level of accuracy as more complex models when large amounts of training data are available. In such cases, more sophisticated algorithms can capture more intricate patterns.

3.5.2.6 k-NN

K-nearest neighbor (k-NN) is a straightforward yet effective algorithm in classification [44]. Its simplicity lies in its concept: rather than learning a model from the training data, k-NN directly memorizes the training dataset. Selecting an appropriate value for k involves experimentation and validation to discover the optimal balance bias and variance in the model. Additionally, in some cases, distance-weighted voting, where closer neighbors have more influence on the classification decision, can further enhance the performance of k-NN.

3.5.2.7 Bytes Net

Bayesian Network Classifiers are powerful tools for modeling and reasoning about complex probabilistic relationships among variables [50]. They provide a robust framework for classification tasks when dealing with uncertainty and dependencies. During training, the parameters of the Bayesian network, such as conditional probability tables (CPTs), are learned from the training data. It involves estimating the probabilities of each variable given its parent variables in the network. Once trained, the Bayesian network is for classification. Given a new instance with values for some variables, the network computes the posterior probability distribution over the class variable(s) given the observed values. Once the posterior probabilities for each class, a classifier can decide tree, the class with the highest probability, or use a threshold to make decision based on the probabilities.

3.5.2.8 Random Tree

A Random Tree is a decision tree where the feature to split on at each node is done randomly, rather than using a deterministic algorithm like choosing the feature that provides the best split according to some criterion. This randomness introduces variability among the trees in an ensemble, which helps improve the overall model's robustness and generalization performance. When deciding which feature to split on at each node, a Random Tree selects from a random subset of the available features rather than evaluating all possible features. This subset is typically much smaller than the total number of features. Sometimes, random thresholds within the chosen feature's range may be considered for the split, adding another layer of randomness. Similar to standard decision trees, Random Trees construct recursively. They are popular in practice due to their performance across datasets and their resistance to overfitting. Additionally, they require relatively few hyperparameters to tune, making them easy to use out of the box.

3.6 Effective Feature Selection Methods

Feature selection applied a crucial step in machine learning. It involves choosing a subset of relevant features from the original set of features in the dataset. There are several techniques for feature selection, including: Filter Methods, Wrapper Methods, and Embedded Methods.

Filter methods are fast, scalable, and independent of the model, but they may ignore the interactions and dependencies among the features. They can use filter methods as a preprocessing step to reduce the dimensionality of their data and select the most relevant features. Filter methods rank features based on statistical scores or other metrics and select the top-ranked features. Common techniques include correlation analysis, mutual information, and chi-square tests [9].

Wrapper methods involve evaluating feature subsets using a specific machine learning algorithm, typically in combination with a performance metric. These methods assess different feature subsets on various compounds of features. Wrapper methods are powerful because they directly evaluate the model using the chosen evaluation criterion, the interactions between features, and their impact on model performance [31]. As such, careful consideration must be given to the computational resources available and the specific characteristics of the dataset when choosing and implementing wrapper methods for feature selection.

Embedded methods combine aspects of both filter and wrapper methods. They perform feature selection as part of the model training process, leveraging regularization techniques to penalize the complexity of the model and automatically shrink the coefficients of fewer features. Despite these challenges, embedded methods remain widely used and valuable in practice due to their efficiency, robustness, and ability to adapt to the model.

3.7 Attack Types

This section provides an overview of common cyber-attack types on network security. Understanding the various types of cyber-attacks is crucial for developing effective defense strategies. Each attack type exploits different vulnerabilities and requires specific countermeasures. By comprehensively studying these attacks, organizations can enhance their security posture, improve threat detection, and implement robust response mechanisms to safeguard their assets. In this chapter, two types of attacks are described. The author [37] provides the definitions and importance of security, including confidentiality, integrity, and availability. It knows the various types of threats such as malware, phishing, and DoS attacks.

3.7.1 Effort of DoS Attack

A DoS attack is a malicious attempt to disrupt the normal functioning of a targeted server, service, or network by overwhelming it with a flood of illegitimate traffic or requests. The objective of a DoS attack is to make the target system unavailable to its intended users, thereby denying access to legitimate users.

Attackers flood the target system with an excessive volume of traffic, such as UDP, ICMP, or SYN packets, overwhelming its capacity to handle legitimate requests. Attackers exploit vulnerabilities in the target system's resources, such as CPU, memory, or bandwidth, by sending specially crafted requests that consume resources and prevent legitimate users from accessing them. Some DoS attacks exploit weaknesses in network protocols or application-layer protocols to exhaust system resources or cause the target system to crash. The types of DoS attacks are:

SYN Flood: In a SYN flood attack, attackers send several TCP connection requests with fake source IP addresses, exhausting the target system's resources and preventing it from accepting legitimate connections.

UDP Flood: UDP flood attacks flood the target system with a high volume of User Datagram Protocol (UDP) packets, causing it to consume resources processing these packets without establishing a connection.

ICMP Flood: ICMP flood attacks overwhelm the target system with Internet Control Message Protocol (ICMP) packets, such as ping requests, disrupting its operation.

HTTP Flood: In an HTTP flood attack, attackers send HTTP requests to a web server, consuming its resources and making it unavailable to legitimate users.

Slowloris: Slowloris is a type of DoS attack that exploits vulnerabilities in web servers by sending partial HTTP requests and keeping connections open for as long as possible, consuming server resources and preventing new connections.

3.7.2 Behave of PortScan Attack

A PortScan attack is a reconnaissance technique to discover open ports and services running on a target system. It involves systematically scanning a range of IP addresses and port numbers to identify vulnerable systems or services that further attacks. By sending packets to specific ports and analyzing the responses (if any), attackers can identify potential vulnerabilities, assess the target system, and gather valuable intelligence for launching subsequent attacks [62].

Attackers identify a range of IP addresses or a specific target network to scan for potential vulnerabilities. Attackers use specialized tools or scripts to scan the target network for open ports. They typically scan ports associated with commonly used services (e.g., HTTP on port 80, SSH on port 22) and other ports to identify less common services. Once, attackers attempt to identify the services running on those ports by sending probes or requests to the target system. It helps attackers determine potential vulnerabilities and attack vectors. After identifying open ports and services, attackers analyze the results to identify potential vulnerabilities that further attack. It may involve researching known vulnerabilities associated with specific services or conducting additional to gather more information about the target system.

3.8 Summary

The integration of firewalls and IDS creates a multi-layered defense strategy that significantly enhances network security. Firewalls act as the initial gatekeepers, while IDS provide deeper inspection and anomaly detection, together forming a formidable barrier against cyber threats. This chapter underscores the importance of using these technologies in tandem to protect sensitive information and maintain robust network security. Machine learning classifiers and feature selection methods form the backbone of advanced security solutions. Classifiers help in accurately identifying and categorizing different types of cyber threats, while feature selection ensures that the models are both efficient and effective. Together, these technologies enhance the ability to detect, respond to, and mitigate security threats, providing a stronger and more adaptive defense mechanism.

CHAPTER 4
FEATURE SELECTION APPROACH FOR SFBIDS

In this chapter, features selection is described in the first part, and feature selection method are used in the second part, and the machine learning tool operates to prove that the features and especially the false positive rate are good. In the system, feature selection and whether it is correct to select the features will present in detail with two feature selection methods for attributes. These two methods are Correlation Based Feature Subset (CBFS) and Gain Ratio Feature Selection (GRFS). These feature selection techniques take part a role in optimizing the model's effectiveness by choosing relevant attributes for network traffic classification. The comparative study illustrates the significance of utilizing GRFS and CBFS for feature subset selection in the classification process. The reduced of false positive rate not only showcases the superior of its features but also increased accuracy, cost savings, and overall effectiveness.

4.1 Feature Evaluation

Reducing the false positive rate is crucial in demonstrating the superiority of features in a proposed system for several reasons:

Enhanced Accuracy: A lower false positive rate signifies that the system is better at correctly identifying positive instances, leading to higher accuracy and reliability in feature predictions; **Cost Implications:** In applications such as intrusion detection or fraud prediction, minimizing false positives is essential to avoid unnecessary costs associated with false alarms or incorrect classifications; **Time Efficiency:** The minimize the false positive improves the precision and saves time by preventing unnecessary investigations or actions triggered by false alarms; **System Effectiveness:** A system with a lower false positive rate demonstrates effectiveness in distinguishing between relevant and irrelevant instances, making it more valuable in real-world applications; and **Resource Optimization:** Minimizing false positives optimizes the allocation of resources by focusing attention on true positive instances, improving overall efficiency.

4.2 Selected Features of SFBIDS

The proposed dataset now included sixteen keys features in Table 4.1. The dataset derived by extracting some features as destination port, minimum packet length and maximum packet length [24], [43] from CICIDS-2017 and added other features to reduce false positive rate. These features are considered depending on the destination according to the packet range, such as destination ports, destination inbound/outbound packets, etc. Features are not specifically designed for the flag feature. Adding TCP flag features do not significantly improve the performance and overhead of the system resources. Therefore, instead of applying those features, synchronization (syn), synchronization and acknowledgement (syn_ack), retransmission, reset (rst) are categorized into package. Firstly, the system considers with respective features based on time as 3s, 5s, 10s in normal and attack traffics [84].

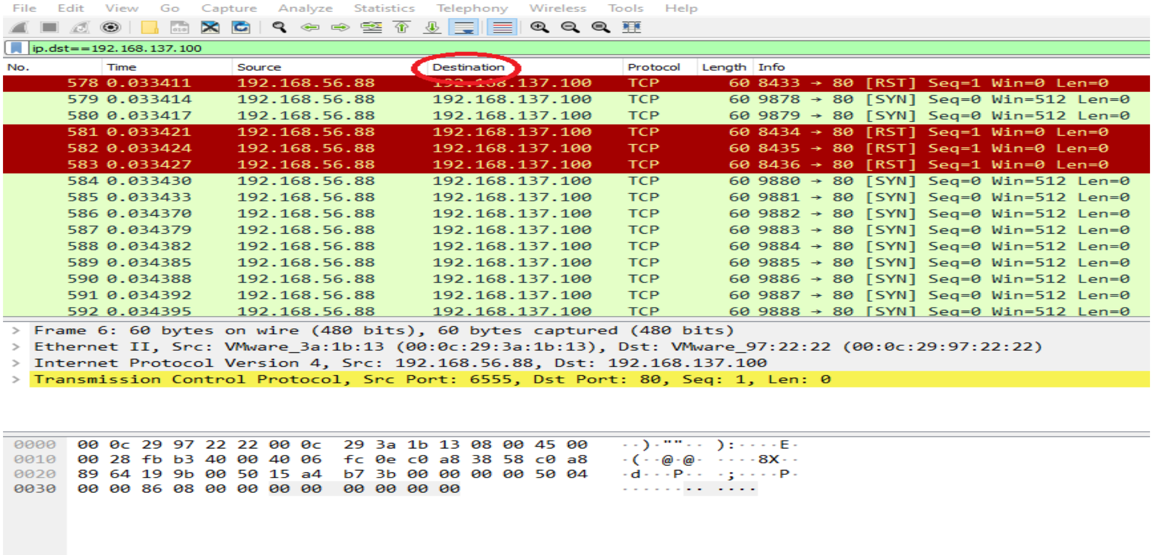
Table 4.1 Selected Features

No	Features	Description
1	Dst_port	Destination Port
2	Dst_IP	Target IP Address
3	Total_Inpkt	Total Inbound packages to destination host
4	Total_Outpkt	Total Outbound packages from destination host
5	Inpkt_bytes	Inbound packages bytes to destination
6	Outpkt_bytes	Outbound package bytes from destination
7	Total_InOut_pkt	Total packages to/from destination host
8	Inpkt_bits/s	Inbound packet bits/s to destination
9	Outpkt_bits/s	Outbound packet bits/s from destination
10	Protocol	Protocol as TCP or UDP
11	Service	Service type as http, ftp
12	Min_pktlen	Minimum packet length in the packet range
13	Max_pktlen	Maximum packet length in the packet range
14	Avg_pktlen	Average packet length that fall in the packet range

15	InOut_count	Number of packet count with source and destination IP in this range
16	Class	Describe normal or attack

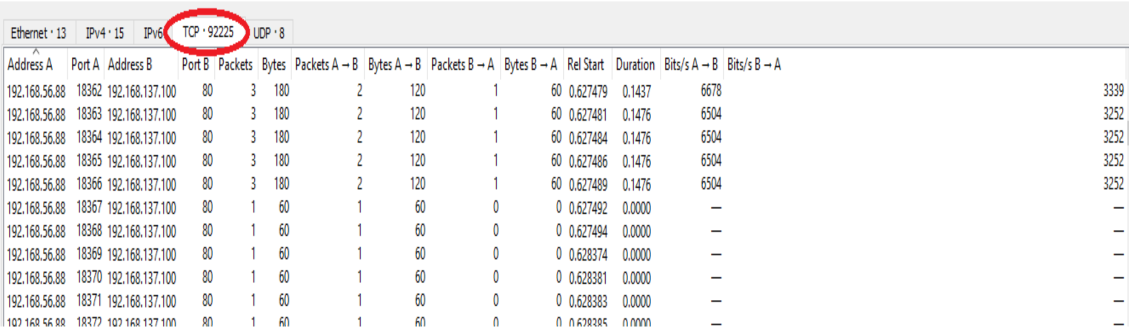
4.2.1 Definition of Features Values

The difference of features values is calculated manually the group of package range based on destination IP address in the selected network traffic interval. For example, when accessing attack traffic, the traffic is filtered that accesses the web server at the destination from more than 400 thousand of traffic in Figure 4.1. The main one of the four filters is calculated based on the destination host's TCP packet in Figure 4.2. In Normal traffic, depending on the filtered traffic of each different destination host, the bytes data of each inbound packet and the total bytes data are calculated in detail and set as a feature value in Figure 4.3.



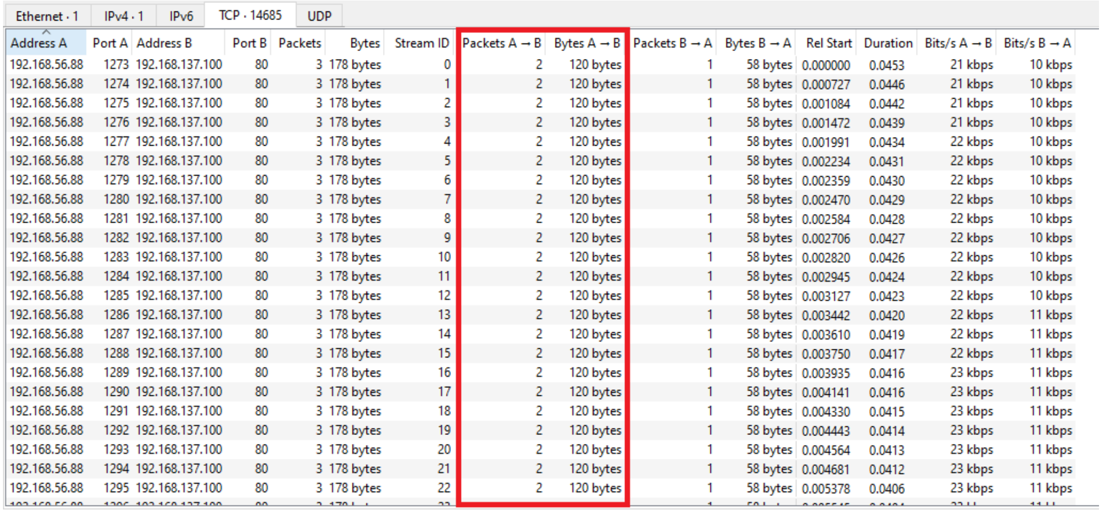
```
ip.dst==192.168.137.100
No.    Time        Source          Destination       Protocol Length Info
   578 0.033411    192.168.56.88   192.168.137.100   TCP      60 8433 → 80 [RST] Seq=1 Win=0 Len=0
   579 0.033414    192.168.56.88   192.168.137.100   TCP      60 9878 → 80 [SYN] Seq=0 Win=512 Len=0
   580 0.033417    192.168.56.88   192.168.137.100   TCP      60 9879 → 80 [SYN] Seq=0 Win=512 Len=0
   581 0.033421    192.168.56.88   192.168.137.100   TCP      60 8434 → 80 [RST] Seq=1 Win=0 Len=0
   582 0.033424    192.168.56.88   192.168.137.100   TCP      60 8435 → 80 [RST] Seq=1 Win=0 Len=0
   583 0.033427    192.168.56.88   192.168.137.100   TCP      60 8436 → 80 [RST] Seq=1 Win=0 Len=0
   584 0.033430    192.168.56.88   192.168.137.100   TCP      60 9880 → 80 [SYN] Seq=0 Win=512 Len=0
   585 0.033433    192.168.56.88   192.168.137.100   TCP      60 9881 → 80 [SYN] Seq=0 Win=512 Len=0
   586 0.034370    192.168.56.88   192.168.137.100   TCP      60 9882 → 80 [SYN] Seq=0 Win=512 Len=0
   587 0.034379    192.168.56.88   192.168.137.100   TCP      60 9883 → 80 [SYN] Seq=0 Win=512 Len=0
   588 0.034382    192.168.56.88   192.168.137.100   TCP      60 9884 → 80 [SYN] Seq=0 Win=512 Len=0
   589 0.034385    192.168.56.88   192.168.137.100   TCP      60 9885 → 80 [SYN] Seq=0 Win=512 Len=0
   590 0.034388    192.168.56.88   192.168.137.100   TCP      60 9886 → 80 [SYN] Seq=0 Win=512 Len=0
   591 0.034392    192.168.56.88   192.168.137.100   TCP      60 9887 → 80 [SYN] Seq=0 Win=512 Len=0
   592 0.034395    192.168.56.88   192.168.137.100   TCP      60 9888 → 80 [SYN] Seq=0 Win=512 Len=0
> Frame 6: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
> Ethernet II, Src: VMware_3a:1b:13 (00:0c:29:3a:1b:13), Dst: VMware_97:22:22 (00:0c:29:97:22:22)
> Internet Protocol Version 4, Src: 192.168.56.88, Dst: 192.168.137.100
> Transmission Control Protocol, Src Port: 6555, Dst Port: 80, Seq: 1, Len: 0

0000  00 0c 29 97 22 22 00 0c  29 3a 1b 13 08 00 45 00   ··)·""··  ):····E·
0010  00 28 fb b3 40 00 40 06  fc 0e c0 a8 38 58 c0 a8   ·(··@·@·  ····8X··
0020  89 64 19 9b 00 50 15 a4  b7 3b 00 00 00 00 50 04   ·d···P··  ·;····P·
0030  00 00 86 08 00 00 00 00  00 00 00 00               ········  ····
```

Figure 4.1 Selected Values on Destination Host



Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.56.88	18362	192.168.137.100	80	3	180	2	120	1	60	0.627479	0.1437	6678	3339
192.168.56.88	18363	192.168.137.100	80	3	180	2	120	1	60	0.627481	0.1476	6504	3252
192.168.56.88	18364	192.168.137.100	80	3	180	2	120	1	60	0.627484	0.1476	6504	3252
192.168.56.88	18365	192.168.137.100	80	3	180	2	120	1	60	0.627486	0.1476	6504	3252
192.168.56.88	18366	192.168.137.100	80	3	180	2	120	1	60	0.627489	0.1476	6504	3252
192.168.56.88	18367	192.168.137.100	80	1	60	1	60	0	0	0.627492	0.0000	—	—
192.168.56.88	18368	192.168.137.100	80	1	60	1	60	0	0	0.627494	0.0000	—	—
192.168.56.88	18369	192.168.137.100	80	1	60	1	60	0	0	0.628374	0.0000	—	—
192.168.56.88	18370	192.168.137.100	80	1	60	1	60	0	0	0.628381	0.0000	—	—
192.168.56.88	18371	192.168.137.100	80	1	60	1	60	0	0	0.628383	0.0000	—	—
192.168.56.88	18372	192.168.137.100	80	1	60	1	60	0	0	0.628385	0.0000	—	—

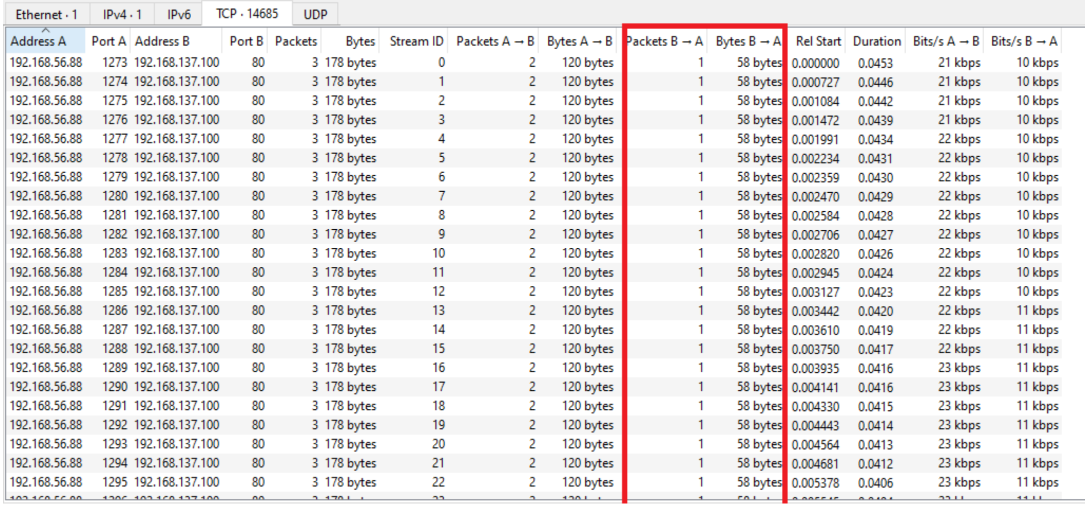
Figure 4.2 Selected Values on TCP Protocol



Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.56.88	1273	192.168.137.100	80	3	178 bytes	0	2	120 bytes	1	58 bytes	0.000000	0.0453	21 kbps	10 kbps
192.168.56.88	1274	192.168.137.100	80	3	178 bytes	1	2	120 bytes	1	58 bytes	0.000727	0.0446	21 kbps	10 kbps
192.168.56.88	1275	192.168.137.100	80	3	178 bytes	2	2	120 bytes	1	58 bytes	0.001084	0.0442	21 kbps	10 kbps
192.168.56.88	1276	192.168.137.100	80	3	178 bytes	3	2	120 bytes	1	58 bytes	0.001472	0.0439	21 kbps	10 kbps
192.168.56.88	1277	192.168.137.100	80	3	178 bytes	4	2	120 bytes	1	58 bytes	0.001991	0.0434	22 kbps	10 kbps
192.168.56.88	1278	192.168.137.100	80	3	178 bytes	5	2	120 bytes	1	58 bytes	0.002234	0.0431	22 kbps	10 kbps
192.168.56.88	1279	192.168.137.100	80	3	178 bytes	6	2	120 bytes	1	58 bytes	0.002359	0.0430	22 kbps	10 kbps
192.168.56.88	1280	192.168.137.100	80	3	178 bytes	7	2	120 bytes	1	58 bytes	0.002470	0.0429	22 kbps	10 kbps
192.168.56.88	1281	192.168.137.100	80	3	178 bytes	8	2	120 bytes	1	58 bytes	0.002584	0.0428	22 kbps	10 kbps
192.168.56.88	1282	192.168.137.100	80	3	178 bytes	9	2	120 bytes	1	58 bytes	0.002706	0.0427	22 kbps	10 kbps
192.168.56.88	1283	192.168.137.100	80	3	178 bytes	10	2	120 bytes	1	58 bytes	0.002820	0.0426	22 kbps	10 kbps
192.168.56.88	1284	192.168.137.100	80	3	178 bytes	11	2	120 bytes	1	58 bytes	0.002945	0.0424	22 kbps	10 kbps
192.168.56.88	1285	192.168.137.100	80	3	178 bytes	12	2	120 bytes	1	58 bytes	0.003127	0.0423	22 kbps	10 kbps
192.168.56.88	1286	192.168.137.100	80	3	178 bytes	13	2	120 bytes	1	58 bytes	0.003442	0.0420	22 kbps	11 kbps
192.168.56.88	1287	192.168.137.100	80	3	178 bytes	14	2	120 bytes	1	58 bytes	0.003610	0.0419	22 kbps	11 kbps
192.168.56.88	1288	192.168.137.100	80	3	178 bytes	15	2	120 bytes	1	58 bytes	0.003750	0.0417	22 kbps	11 kbps
192.168.56.88	1289	192.168.137.100	80	3	178 bytes	16	2	120 bytes	1	58 bytes	0.003935	0.0416	23 kbps	11 kbps
192.168.56.88	1290	192.168.137.100	80	3	178 bytes	17	2	120 bytes	1	58 bytes	0.004141	0.0416	23 kbps	11 kbps
192.168.56.88	1291	192.168.137.100	80	3	178 bytes	18	2	120 bytes	1	58 bytes	0.004330	0.0415	23 kbps	11 kbps
192.168.56.88	1292	192.168.137.100	80	3	178 bytes	19	2	120 bytes	1	58 bytes	0.004443	0.0414	23 kbps	11 kbps
192.168.56.88	1293	192.168.137.100	80	3	178 bytes	20	2	120 bytes	1	58 bytes	0.004564	0.0413	23 kbps	11 kbps
192.168.56.88	1294	192.168.137.100	80	3	178 bytes	21	2	120 bytes	1	58 bytes	0.004681	0.0412	23 kbps	11 kbps
192.168.56.88	1295	192.168.137.100	80	3	178 bytes	22	2	120 bytes	1	58 bytes	0.005378	0.0406	23 kbps	11 kbps

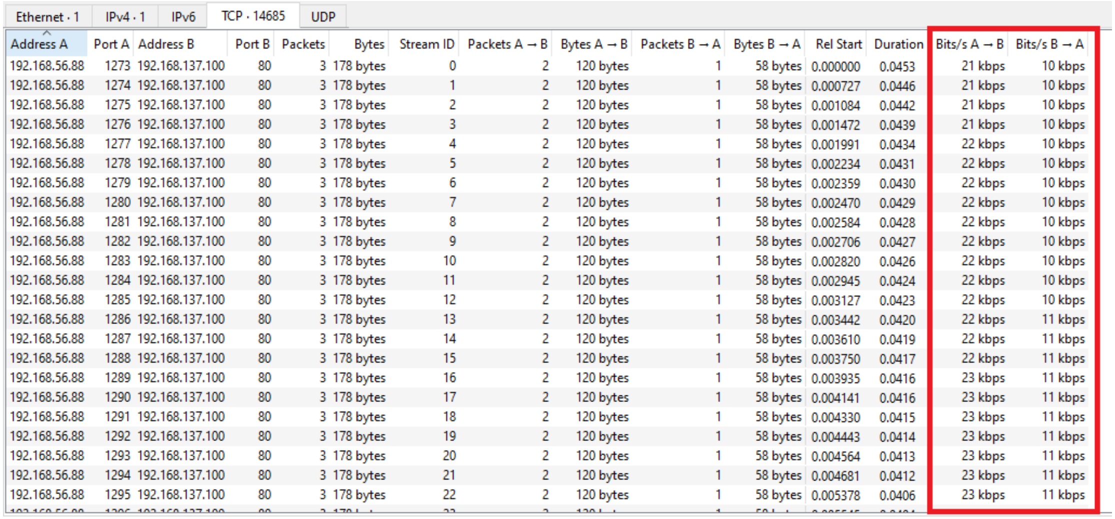
Figure 4.3 Preferred Data on Inbound Packets with Bytes/s

Figure 4.4 shows the outbound packets from the destination host to the source. The total number of packets and the total number of bytes of those packets are in the feature values. The total number of packets are sum up of the count of all packets in the dataset. The total number of bytes are sum up the sizes of all packets. Packet Count Analysis: Determine the frequency and pattern of packet transmissions. It can help potential trends, anomalies, or specific periods with high activity. Byte Count Analysis: Assess the volume of data. High byte counts could indicate large data transfers, potentially signaling file transfers, streaming, or other data-intensive activities. Figure 4.5 elects the inbound/outbound packets, and the total number of bits in all packets is a feature value. The total number of bits is the cumulative size of all packets in both inbound and outbound, measured in bits.



Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.56.88	1273	192.168.137.100	80	3	178 bytes	0	2	120 bytes	1	58 bytes	0.000000	0.0453	21 kbps	10 kbps
192.168.56.88	1274	192.168.137.100	80	3	178 bytes	1	2	120 bytes	1	58 bytes	0.000727	0.0446	21 kbps	10 kbps
192.168.56.88	1275	192.168.137.100	80	3	178 bytes	2	2	120 bytes	1	58 bytes	0.001084	0.0442	21 kbps	10 kbps
192.168.56.88	1276	192.168.137.100	80	3	178 bytes	3	2	120 bytes	1	58 bytes	0.001472	0.0439	21 kbps	10 kbps
192.168.56.88	1277	192.168.137.100	80	3	178 bytes	4	2	120 bytes	1	58 bytes	0.001991	0.0434	22 kbps	10 kbps
192.168.56.88	1278	192.168.137.100	80	3	178 bytes	5	2	120 bytes	1	58 bytes	0.002234	0.0431	22 kbps	10 kbps
192.168.56.88	1279	192.168.137.100	80	3	178 bytes	6	2	120 bytes	1	58 bytes	0.002359	0.0430	22 kbps	10 kbps
192.168.56.88	1280	192.168.137.100	80	3	178 bytes	7	2	120 bytes	1	58 bytes	0.002470	0.0429	22 kbps	10 kbps
192.168.56.88	1281	192.168.137.100	80	3	178 bytes	8	2	120 bytes	1	58 bytes	0.002584	0.0428	22 kbps	10 kbps
192.168.56.88	1282	192.168.137.100	80	3	178 bytes	9	2	120 bytes	1	58 bytes	0.002706	0.0427	22 kbps	10 kbps
192.168.56.88	1283	192.168.137.100	80	3	178 bytes	10	2	120 bytes	1	58 bytes	0.002820	0.0426	22 kbps	10 kbps
192.168.56.88	1284	192.168.137.100	80	3	178 bytes	11	2	120 bytes	1	58 bytes	0.002945	0.0424	22 kbps	10 kbps
192.168.56.88	1285	192.168.137.100	80	3	178 bytes	12	2	120 bytes	1	58 bytes	0.003127	0.0423	22 kbps	10 kbps
192.168.56.88	1286	192.168.137.100	80	3	178 bytes	13	2	120 bytes	1	58 bytes	0.003442	0.0420	22 kbps	11 kbps
192.168.56.88	1287	192.168.137.100	80	3	178 bytes	14	2	120 bytes	1	58 bytes	0.003610	0.0419	22 kbps	11 kbps
192.168.56.88	1288	192.168.137.100	80	3	178 bytes	15	2	120 bytes	1	58 bytes	0.003750	0.0417	22 kbps	11 kbps
192.168.56.88	1289	192.168.137.100	80	3	178 bytes	16	2	120 bytes	1	58 bytes	0.003935	0.0416	23 kbps	11 kbps
192.168.56.88	1290	192.168.137.100	80	3	178 bytes	17	2	120 bytes	1	58 bytes	0.004141	0.0416	23 kbps	11 kbps
192.168.56.88	1291	192.168.137.100	80	3	178 bytes	18	2	120 bytes	1	58 bytes	0.004330	0.0415	23 kbps	11 kbps
192.168.56.88	1292	192.168.137.100	80	3	178 bytes	19	2	120 bytes	1	58 bytes	0.004443	0.0414	23 kbps	11 kbps
192.168.56.88	1293	192.168.137.100	80	3	178 bytes	20	2	120 bytes	1	58 bytes	0.004564	0.0413	23 kbps	11 kbps
192.168.56.88	1294	192.168.137.100	80	3	178 bytes	21	2	120 bytes	1	58 bytes	0.004681	0.0412	23 kbps	11 kbps
192.168.56.88	1295	192.168.137.100	80	3	178 bytes	22	2	120 bytes	1	58 bytes	0.005378	0.0406	23 kbps	11 kbps

Figure 4.4 Preferred Data on Outbound Packets with Bytes/s



Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.56.88	1273	192.168.137.100	80	3	178 bytes	0	2	120 bytes	1	58 bytes	0.000000	0.0453	21 kbps	10 kbps
192.168.56.88	1274	192.168.137.100	80	3	178 bytes	1	2	120 bytes	1	58 bytes	0.000727	0.0446	21 kbps	10 kbps
192.168.56.88	1275	192.168.137.100	80	3	178 bytes	2	2	120 bytes	1	58 bytes	0.001084	0.0442	21 kbps	10 kbps
192.168.56.88	1276	192.168.137.100	80	3	178 bytes	3	2	120 bytes	1	58 bytes	0.001472	0.0439	21 kbps	10 kbps
192.168.56.88	1277	192.168.137.100	80	3	178 bytes	4	2	120 bytes	1	58 bytes	0.001991	0.0434	22 kbps	10 kbps
192.168.56.88	1278	192.168.137.100	80	3	178 bytes	5	2	120 bytes	1	58 bytes	0.002234	0.0431	22 kbps	10 kbps
192.168.56.88	1279	192.168.137.100	80	3	178 bytes	6	2	120 bytes	1	58 bytes	0.002359	0.0430	22 kbps	10 kbps
192.168.56.88	1280	192.168.137.100	80	3	178 bytes	7	2	120 bytes	1	58 bytes	0.002470	0.0429	22 kbps	10 kbps
192.168.56.88	1281	192.168.137.100	80	3	178 bytes	8	2	120 bytes	1	58 bytes	0.002584	0.0428	22 kbps	10 kbps
192.168.56.88	1282	192.168.137.100	80	3	178 bytes	9	2	120 bytes	1	58 bytes	0.002706	0.0427	22 kbps	10 kbps
192.168.56.88	1283	192.168.137.100	80	3	178 bytes	10	2	120 bytes	1	58 bytes	0.002820	0.0426	22 kbps	10 kbps
192.168.56.88	1284	192.168.137.100	80	3	178 bytes	11	2	120 bytes	1	58 bytes	0.002945	0.0424	22 kbps	10 kbps
192.168.56.88	1285	192.168.137.100	80	3	178 bytes	12	2	120 bytes	1	58 bytes	0.003127	0.0423	22 kbps	10 kbps
192.168.56.88	1286	192.168.137.100	80	3	178 bytes	13	2	120 bytes	1	58 bytes	0.003442	0.0420	22 kbps	11 kbps
192.168.56.88	1287	192.168.137.100	80	3	178 bytes	14	2	120 bytes	1	58 bytes	0.003610	0.0419	22 kbps	11 kbps
192.168.56.88	1288	192.168.137.100	80	3	178 bytes	15	2	120 bytes	1	58 bytes	0.003750	0.0417	22 kbps	11 kbps
192.168.56.88	1289	192.168.137.100	80	3	178 bytes	16	2	120 bytes	1	58 bytes	0.003935	0.0416	23 kbps	11 kbps
192.168.56.88	1290	192.168.137.100	80	3	178 bytes	17	2	120 bytes	1	58 bytes	0.004141	0.0416	23 kbps	11 kbps
192.168.56.88	1291	192.168.137.100	80	3	178 bytes	18	2	120 bytes	1	58 bytes	0.004330	0.0415	23 kbps	11 kbps
192.168.56.88	1292	192.168.137.100	80	3	178 bytes	19	2	120 bytes	1	58 bytes	0.004443	0.0414	23 kbps	11 kbps
192.168.56.88	1293	192.168.137.100	80	3	178 bytes	20	2	120 bytes	1	58 bytes	0.004564	0.0413	23 kbps	11 kbps
192.168.56.88	1294	192.168.137.100	80	3	178 bytes	21	2	120 bytes	1	58 bytes	0.004681	0.0412	23 kbps	11 kbps
192.168.56.88	1295	192.168.137.100	80	3	178 bytes	22	2	120 bytes	1	58 bytes	0.005378	0.0406	23 kbps	11 kbps

Figure 4.5 Elected Data on Inbound/Outbound Packets with Bits/s

4.3 Feature Selectin Methods

Feature selection methods support to eradicate the redundant and irrelevant data to effective and improve accuracy for classification in the intrusion detection system. It is crucial in machine learning for several reasons: (1) Enhanced model performance, (2) Faster training, (3) Reduced complexity, and (4) Improved exploration. The feature selection method reduces the complexity time, system resources as CPU and memory, and calculation time of data.

4.3.1 The Best Features of Correlation-based Feature Subset (CBFS)

A Correlation-based feature subset is a wrapper method. A wrapper method is a method that uses different subsets of features to judge the performance of a machine learning model. This approach involves selecting features based on their correlation with the target variable. It evaluates intrinsic value of each attribute to allowing for the identification of features with strong associations with the classification task. Table 4.2 and Table 4.3 show the performance of SFBIDS dataset on DoS and PortScan attacks using CBFS feature selection with five classifiers.

Table 4.2 FPR Result on DoS Attack using CBFS

Classifiers	CBFS Selected Features	TPR	FPR	PRE	REC	ACC
Logistic Regression	Dst_port, Inpkt_bits/s, Min_pktlen, Max_pktlen, Avg_pktlen, InOut_count	0.999	**0.001**	0.999	0.999	99.900
Naïve Bayes		0.995	**0.004**	0.995	0.995	99.502
Bayes Net		1.000	**0.000**	1.000	1.000	100
J48		0.999	**0.001**	0.999	0.999	99.900
Random Tree		1.000	**0.000**	1.000	1.000	100

Table 4.3 FPR Result on PortScan Attack using CBFS

Classifiers	CBFS Selected Features	TPR	FPR	PRE	REC	ACC
Logistic Regression	Outpkt_bits/s, Services, Min_pktlen, Max_pktlen, InOut_count	0.995	**0.023**	0.995	0.995	99.542
Naïve Bayes		0.995	**0.000**	0.996	0.995	99.542
Bayes Net		1.000	**0.000**	1.000	1.000	100
J48		1.000	**0.000**	1.000	1.000	100
Random Tree		1.000	**0.000**	1.000	1.000	100

4.3.2 Gain Ration Feature Selection (GRFS)

Gain Ratio Feature Selection is a technique used in machine learning to identify relevant attributes for model training. It considers both information gain and the number of outcomes associated with a feature. The GRFS calculates the mutual information normalized by feature entropy, aiding in the election of informative features and enhancing the performance of the model.

Unlike information gain, which may favor attributes with many outcomes, gain ratio normalizes this bias, making it particularly useful in datasets with varying feature dimensions. It helps determine the most informative attributes for classification or regression tasks, providing a balanced approach to feature selection.

4.3.2.1 Gain Ration Feature Selection with String

In feature selection, the first step is to organize a dataset to reconnoiter; In the second step, the parameters of the dataset make the best forecast model with the least number of variables. Typically, the procedure involves ranking (string) variables based on their information gain values, starting from the highest in Table 4.4. The string order helps select the most relevant features for model building, optimizing predictive accuracy in Table 4.5.

Table 4.4 Selected Features from GRFS

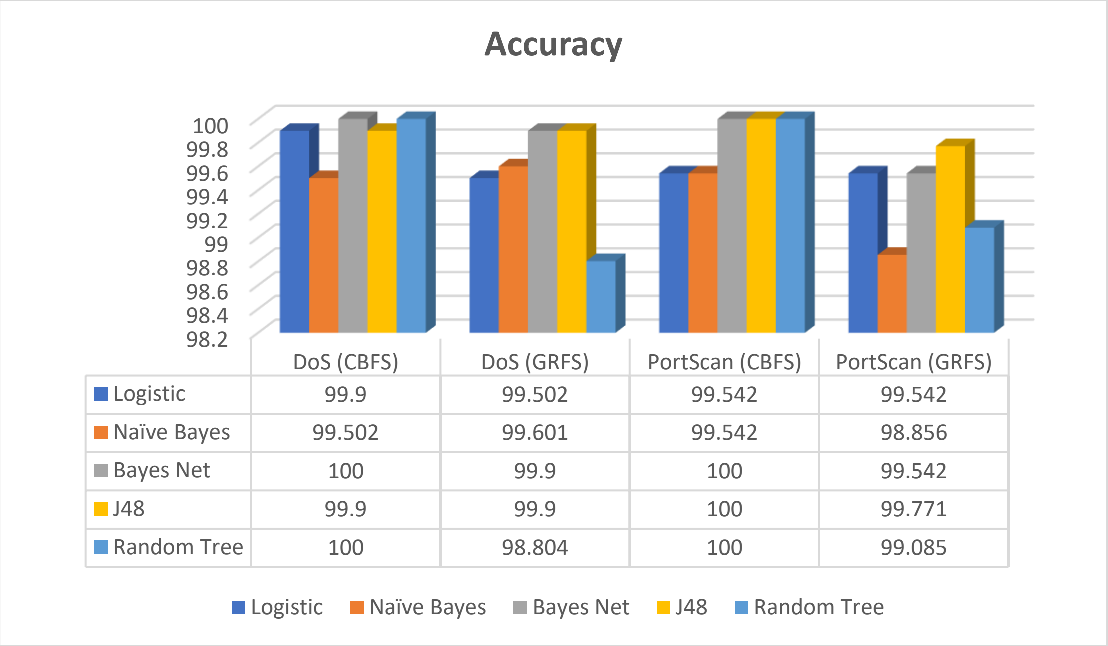
No	Selected Features	String
1	Avg_pktlen	0.99
2	Max_pktlen	0.832
3	Dst_port	0.782
4	Service	0.731
5	Inpkt_bits/s	0.415
6	Min_pktlen	0.401
7	Outpkt_bits/s	0.359
8	Dst_IP	0.279
9	Total_Inpkt	0.241
10	Total_InOutpkt	0.237
11	Inpkt_bytes/s	0.206
12	Total_Outpkt	0.194
13	InOut_count	0.187
14	Outpkt_bytes/s	0.18
15	Protocol	0.139

Table 4.5 GRFS of FPR Result on Two Attacks

Attacks	Classifiers	TPR	FPR	PRE	REC	ACC
DoS	Logistic Regression	0.995	**0.005**	0.995	0.995	99.502
	Naïve Bayes	0.996	**0.004**	0.996	0.996	99.601
	Bayes Net	0.999	**0.001**	0.999	0.999	99.900
	J48	0.999	**0.001**	0.999	0.999	99.900
	Random Tree	0.988	**0.013**	0.988	0.988	98.804
PortScan	Logistic Regression	0.995	**0.000**	0.996	0.995	99.542
	Naïve Bayes	0.989	**0.001**	0.990	0.989	98.856
	Bayes Net	0.995	**0.000**	0.996	0.995	99.542
	J48	0.998	**0.023**	0.998	0.998	99.771
	Random Tree	0.991	**0.023**	0.991	0.991	99.085

4.4 Evaluation with Feature Selection Methods

In the SFBIDS dataset, two feature selection methods of Correlation-Based Feature Selection (CBFS) and Gain Ratio Feature Selection (GRFS) were used, to prove the accuracy. These methods can help improve the model's performance by reducing overfitting, decreasing computational complexity, and enhancing interpretability. CBFS typically selects features that have the highest correlation with the target variable. It is a straightforward method that can be effective when there are linear relationships between features and the target variable. GRFS calculates the gain ratio of each feature based on its ability to split the data effectively into classes. Features with higher gain ratios are considered more informative for classification tasks. It can be seen in Figure 4.3 that the accuracy of the Random Tree classifier in the GRFS method is 88.9% in the DoS attack. It can be seen that the accuracy of the remaining classifiers is 99 and above in both methods. The SFBIDS dataset proved that the features are good in calculating the performance by using feature selection methods.



	DoS (CBFS)	DoS (GRFS)	PortScan (CBFS)	PortScan (GRFS)
Logistic	99.9	99.502	99.542	99.542
Naïve Bayes	99.502	99.601	99.542	98.856
Bayes Net	100	99.9	100	99.542
J48	99.9	99.9	100	99.771
Random Tree	100	98.804	100	99.085

Figure 4.6 Accuracy with Two Feature Selection Methods

4.5 Summary

This chapter for network traffic classification utilizes machine learning techniques and employs two keys feature selection methods: CBFS and GRFS. By reducing the unnecessary features and their values, the false positive rate and processing time that the system will be effective. In conclusion, feature selection is a vital step in the machine learning pipeline, contributing to improved model performance, efficiency, and interpretability. It is to prove that the features are superior by reducing the false positive rate and saving the time in the system.

CHAPTER 5
IMPLEMENTATION OF THE SYSTEM

The SFBIDS system accomplished the three main parts in this section. The first part is building the system setup and the policies and rules related to users in the organization regarding secure devices. In the second part, the preprocessing part is to build a dataset from the packets obtained by using the network tools and the rules depending on the service from the system setup and normal and attack traffic. In the third part, the false-positive-rate and accuracy are calculated from this dataset. Optimizing intrusion detection systems can conduct to improve false positive rates, enhancing overall performance. The performance of feature selection techniques in machine learning is influenced by dataset characteristics, emphasizing the importance of selecting relevant and high-quality features.

5.1 Implementation of System Design

Operating a software-based firewall like IPCop is a common approach for securing network traffic within an organization. In Figure 5.1, the firewall serves as a crucial barrier between different network segments, such as the WAN, LAN, and DMZ. LAN is the trusted zone where organizational users operate. Rules here ensure secure to communication within the network. WAN represents the untrusted external network. The rules set to filter and monitor incoming and outgoing traffic to safeguard against external threats. DMZ positioned between internal and external zones. It hosts public-facing servers like web and file servers. Forwarding rules are crucial for allowing public user access to these servers while maintaining security. In the DMZ, forwarding rules enable public users to access web servers and file servers securely.

A balance between robust security and optimal system performance is essential when setting firewall rules. Security measures involve defining rules that permit or deny specific types of traffic based on security policies. System performance considerations ensure that the firewall operations do not degrade network speed or responsiveness. Predefined rules for IDS are integral to firewall configurations. These rules are to identify and respond to potential security threats, enhancing the overall security posture.

When implementing these rules, the system can ensure that firewall rules are designed to provide security while minimizing their impact on system performance, ultimately optimizing the overall performance of network infrastructure. While firewalls primarily control traffic, flow based on predefined rulesets, IDS systems focus on monitoring and analyzing network traffic for signs of malicious activity or policy violations. However, both serve to enhance network security and often work in conjunction to provide comprehensive protection. In addition, effective firewall management involves a thoughtful approach to zone definition, forwarding rules, and a careful balance between security and system performance considerations. Incorporating predefined Intrusion Detection rules adds an extra layer of protection to the network.

The system uses two Linux machines to perform attackers in the external network. The network admin and internal user machines are setup with OpenSUSE Linux distribution. The DMZ network hosts a web server with essential services: DNS (Domain Name System): Resolves domain names to IP addresses. HTTP (Hypertext Transfer Protocol): Facilitates web content delivery. And SSH (Secure Shell): Ensures secure communication for remote administration. This network architecture has enhanced security by isolating public-facing services in the DMZ, minimizing the risk of unauthorized access to the internal network. The Linux machines for attackers add a layer of realism for testing and fortifying the network against potential threats.

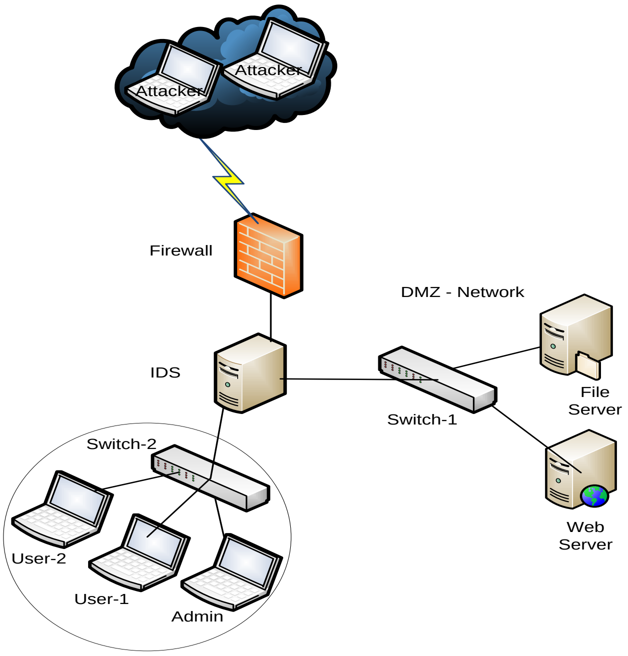


Figure 5.1 System Architecture

5.2 Firewall Policy

When configuring firewall rules, administrators must be cautious to avoid misconfigurations and ensure a secure network environment. The default state of the firewall set to deny all incoming and outgoing traffic. This ensures that no communication is allowed unless explicitly permitted. Administrators must define specific "allow" or "accept" rules for desired IP addresses and services. This explicit definition reduces the risk of misconfiguration and ensures clarity in access permissions. Regularly update and review firewall rules to accommodate changes in network requirements. Failure to update rules consistently can lead to inconsistencies and potential vulnerabilities.

The firewall separates the internal network from the DMZ (Demilitarized Zone). DMZ is a firewall configuration for securing local area networks (LANs). In a DMZ configuration, most computers on the LAN run behind a firewall connected to a public network like the Internet.

IDS is an open source and it uses the popular snort tool and it can know the “access” and “attack” that enter in from snort alert. Firewall and IDS are combined as a system that contributes to protect for unwanted attack entering.

To ensure that network performance is compromised while maintaining robust security, it is a general approach to firewall policies for both internal and external users:

- External users permit ICMP traffic (ping) to access Web server and File server that is located in the DMZ but without through remote service (port 22). These users are not available other access to the firewall.
- Internal users have access to ping the Firewall for diagnostics purposes and to check connectivity and allow ping access to the servers in the DMZ network for the troubleshooting and monitoring purpose. Remote access (e.g., SSH) and HTTPS services (port 443) are restricted to all internal users except administrators. All internal users access HTTP service (port 80) and FTP service (port 21). A rule is implemented in the firewall that blocks internet access for local users who do not adhere to the established policies.
- The system wants to restrict access to firewall management interfaces to only the administrator, allowing HTTPS for web access and SSH for remote access.

The firewall rules are configured based on default rules that align with the organization's policies. The IPCop web interfaces are assigned firewall rules individually each interface with firewall policies.

5.2.1 Create Firewall Rules on Each Interfaces

IPCop firewall is designed with distinct web interfaces and network types to manage and control various aspects of network traffic. Firewall have three web interfaces such as external IPCop Access, Port Forwarding, Internal Traffic.

External IPCop Access dedicated to accessing IPCop externally and provide a gateway for remote management and configuration. Port Forwarding handles the configuration of port forwarding, allowing specific services to be accessible from the external network. Internal Traffic manages the internal traffic within the network, governing communication between devices and services.

It has four types of network interfaces. **Green (LAN)** represents the local area network (LAN) and is associated with the internal network. It typically includes devices such as computers and servers. **Blue (Wifi)** interface designates the interface for wireless network connections, providing connectivity for devices by using Wifi. **Orange (DMZ)** stand for the demilitarized zone and dedicated to less secure, semi-public-facing services like web servers. It acts as an intermediary between the internal network and the external network. **Red (WAN)** represents the Wide Area Network (WAN) or the external network, serving as the gateway to the internet.

The default settings of IPCop firewall are as follows:

- The traffic from GREEN interface (LAN) to other interfaces that are ORANGE interface (DMZ), RED interface (WAN) and BLUE (Wifi) interfaces are not allowed. The IPCop firewall restricts communication from the LAN network to the DMZ, WAN, and Wifi networks.
- The traffic from ORANGE interface (DMZ) is only allowed to the RED interface (WAN), while communication with other interfaces, specifically GREEN (LAN) and BLUE (Wifi), is blocked by default.
- The traffic from BLUE interface (Wifi) and RED interface (WAN) are also blocked.

- The traffic from RED interface (WAN) to other interfaces, including GREEN interface (LAN), BLUE interface (Wifi), and ORANGE interface (DMZ), is also blocked by default.

Based on the default rules allow and deny, rules are set on the respective interfaces. This work uses three interfaces not use Blue interface for Wifi. Firewall has three gateways IP address for each interface as shown in the following.

Firewall LAN interface = 192.168.235.1
Firewall DMZ interface = 192.168.137.1
Firewall WAN interface = 192.168.56.201

5.2.1.1 External IPCop Access

The interface assigned rules 1 to permit Ping access to test the firewall at WAN interface from any external users. Allowing the HTTPS service in the firewall is intended to manage easily through the web interface. In rule 2, the firewall prohibited from external users for the administration access by using port 443. Because outside users can change the configuration of firewall by enabling remote access unnecessary, in rule 3, port 22 is closed to prevent remote access.

Table 5.1 Rule Assign to External IPCop Access

Rule	Protocol	Src_IP	Src_port	Dst_IP	Dst_port	Action
r1	ICMP	192.168.56.0/24	any	192.168.56.201	Ping	allow
r2	TCP	192.168.56.0/24	any	192.168.56.201	443	deny
r3	TCP	192.168.56.0/24	any	192.168.56.201	22	deny

5.2.1.2 Port Forwarding

The interface permits to access ping, http, https, ftp from WAN network to web and file servers in DMZ Network. The rule 1 and, 2 allow Ping access to Web server. The web server permits the web access of http, https and, ftp services from outsider in rule 3,4 and, 5.

Table 5.2 Rule Assign to Port Forwarding Interface

Rule	Protocol	Src_IP	Src_port	Dst_IP	Dst_port	Action
r1	ICMP	192.168.56.0/24	any	192.168.137.100	Ping	allow
r2	ICMP	192.168.56.0/24	any	192.168.137.110	Ping	allow
r3	TCP	192.168.56.0/24	any	192.168.137.100	80	allow
r4	TCP	192.168.56.0/24	any	192.168.137.100	443	allow
r5	TCP	192.168.56.0/24	any	192.168.137.110	21	allow

5.2.1.3 Internal Traffic

In rules 3 and 4, security must restrict gain to remote access port 22. Allowing unrestricted inbound traffic on TCP port 22 poses a significant risk, potentially granting administrator authority to anyone in the local network. By implementing of these rules, only the designated admin can access web and file servers, enhancing overall system security. Rules 1, 2, and the remaining rules design to permit specific services for all local users. ICMP, WEB, and FTP services are allowed, ensuring necessary communication and access while maintaining a secure environment.

Table 5.3 Rule Assign to Internal Traffic

Rule	Protocol	Src_IP	Src_port	Dst_IP	Dst_port	Action
r1	ICMP	192.168.235.0/24	any	192.168.137.100	Ping	allow
r2	ICMP	192.168.235.0/24	any	192.168.137.110	Ping	allow
r3	TCP	192.168.235.50/32	any	192.168.137.100	22	allow
r4	TCP	192.168.235.50/32	any	192.168.137.110	22	allow
r5	TCP	192.168.235.0/24	any	192.168.137.100	80	allow
r6	TCP	192.168.235.0/24	any	192.168.137.100	443	allow
r7	TCP	192.168.235.0/24	any	192.168.137.110	21	allow

5.2.1.4 External IPCoP Access

This is rule permit Ping access from any external to IPCoP firewall.

5.2.1.5 IPCoP Access

In table 5.4, the rule 2 and rule 3 are for the administrator to access and configure the web interface on the firewall and to manage command line. Similarly, the rule 1 is for admin and rule 4 for the local users to get Ping access. The rule 5 and 7 are set to prevent unauthorized users from remotely accessing the firewall from the local network and all DMZ networks. The rule 6 and 8 prohibits the firewall from accessing the web interface and updating it at will.

Table 5.4 Rule Assign to IPCoP

Rule	Protocol	Src_IP	Src_port	Dst_IP	Dst_port	Action
r1	ICMP	192.168.235.50	any	192.168.235.1	Ping	allow
r2	TCP	192.168.235.50	any	192.168.235.1	443	allow
r3	TCP	192.168.235.50	any	192.168.235.1	22	allow
r4	ICMP	192.168.235.0	any	192.168.235.1	Ping	allow
r5	TCP	192.168.235.0	any	192.168.235.1	22	deny
r6	TCP	192.168.235.0	any	192.168.235.1	443	deny
r7	TCP	192.168.137.0	any	192.168.235.1	22	deny
r8	TCP	192.168.137.0	any	192.168.235.1	443	deny

5.2.1.6 Outgoing Traffic

In table 5.5, the firewall permits all internet services from any LAN network and DMZ network to firewall’s WAN interface. But it blocks interface services of punished user in Local LAN.

Table 5.5 Rule Assign Outgoing Traffic of Firewall

Rule	Protocol	Src_IP	Src_port	Dst_IP	Dst_port	Action
r1	TCP	192.168.235.100	any	192.168.56.201	*	deny
r2	TCP	192.168.235.*	any	192.168.56.201	*	allow
r3	TCP	192.168.137.*	any	192.168.56.201	*	allow

5.3 IDS-Snort Rules

Intrusion Detection is a role to enhance system security by detecting and preventing various cyber threats such as malware, trojans, rootkits, and phishing. The IDS employs two Network Interface Cards (NICs), facilitating the monitoring and management of network security. The system configures for the external and internal users to access web and file servers in the Demilitarized Zone (DMZ). This setup enhances accessibility while maintaining security protocols. The IDS has a specific administrator IP address assigned for Secure Shell (SSH) access, providing a secure and controlled means for system administration. The only authorized individuals with the designated IP address can access and manage the IDS through SSH. The rules in Snort's local rules are listed below and are defined under ‘snort/rules’ to detect important services for alert.

```
alert tcp any any -> any 22 (msg: “SSH Traffic detected from any Network”; flow: to server, established; content “SSH –“; sid = 1000005; rev = 2;)

alert tcp 192.168.56.0/24 any -> 192.168.137.100 80 (msg: “HTTP Traffic detected from any Network”; sid = 1000009; rev = 3;)
```

5.4 Preprocessing to Implement of Dataset

All the network traffics use the TcpDump tool to capture and create a pcap file in Figure 5.2. Each of the packet ranges applies the Wireshark tool from the pcap file. The hping3 tool is actually a powerful packet crafting tool commonly employed for various network attacks. It allows users to capture attack traffic by customizing ICMP/UDP/TCP packets and displaying target replies similar to ICMP replies.

The FSBIDS dataset is composed of the main sixteen selected features and the package range from normal and attacks traffic based on times in previous chapter. When choosing the values of the features that calculate in detail depending on the inbound/outbound of the destination host according to the package range in the system.

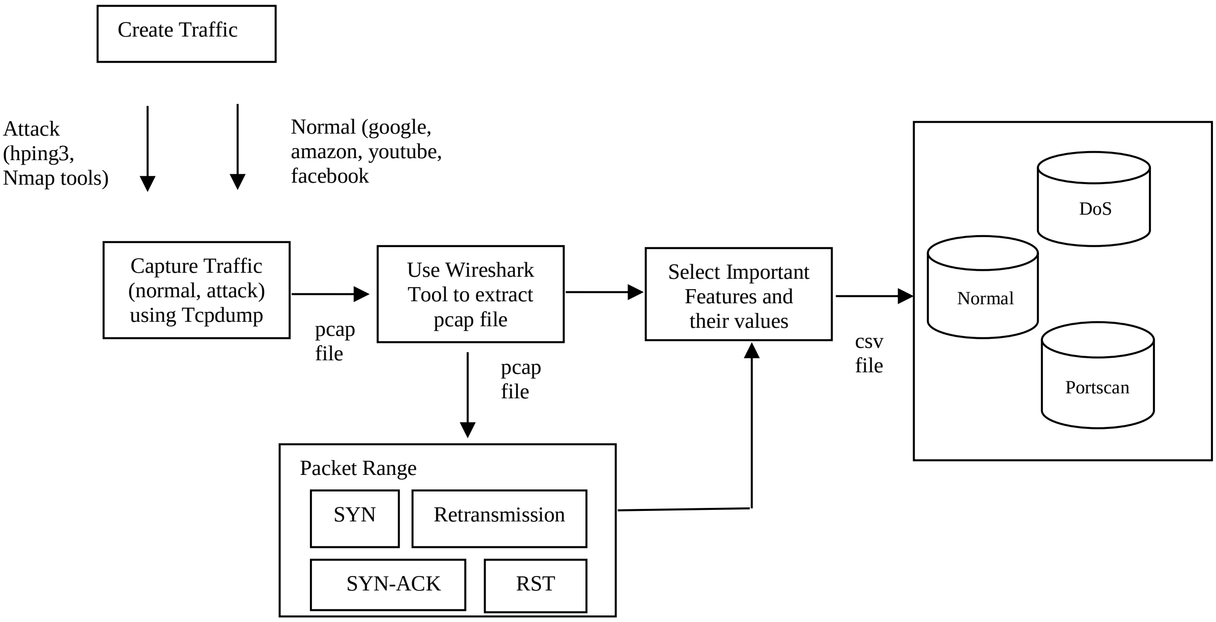

Create Traffic
Attack (hping3, Nmap tools)
Normal (google, amazon, youtube, facebook
Capture Traffic (normal, attack) using Tcpdump
pcap file
Use Wireshark Tool to extract pcap file
pcap file
Select Important Features and their values
csv file
DoS
Normal
Portscan
Packet Range
SYN
Retransmission
SYN-ACK
RST

Figure 5.2 Implementation of SFBIDS Dataset

5.5 The Process Flow of the System

In Figure 5.3 describes the process of the system. Creating rulesets from system services is a common practice in designing network security architectures. In the context of a firewall and an IDS, these rulesets are essential instructions that dictate incoming packets should be handled based on predefined criteria. The firewall checks whether the packet matches any rules in its ruleset.

A firewall inspects the network packets and cross-references them with rules specified by an IDS. Unfortunately, if the firewall misses an attack, IDS will check the packet. The network traffic is used by implementing the dataset. Machine learning classifiers verifying the validity of values and instances in a dataset is a common practice, especially in tasks that prove performance. Using feature selection methods like CBFS and GRFS can help identify the most relevant features in a dataset, which can improve the performance of machine learning classifiers. Comparing the performance of the proposed dataset with an existing dataset like CICIDS-2017 is a valuable approach to validating the effectiveness of the SFBIDS dataset with feature selection methods.

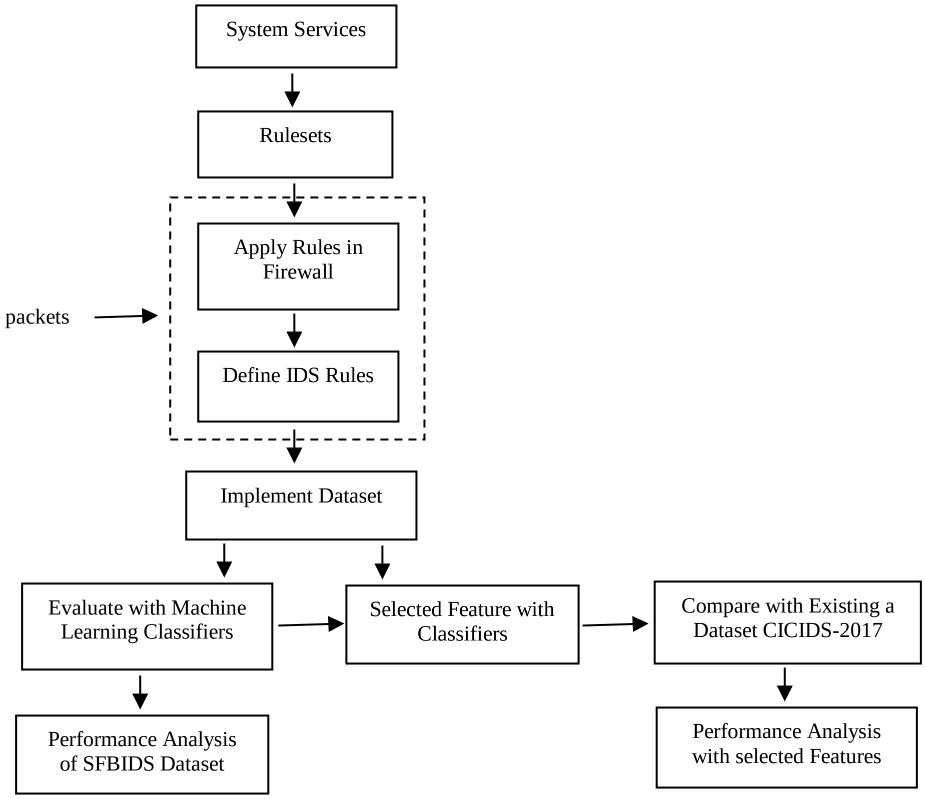

System Services
Rulesets
Apply Rules in Firewall
packets
Define IDS Rules
Implement Dataset
Evaluate with Machine Learning Classifiers
Selected Feature with Classifiers
Compare with Existing a Dataset CICIDS-2017
Performance Analysis of SFBIDS Dataset
Performance Analysis with selected Features

Figure 5.3 Process Flow of the System

5.6 Performance of the SFBIDS System

In implementation, six classifiers are used for the dataset with sixteen features. A high false positive rate indicates that the system may report non-existent attacks, leading to decreased accuracy in attack detection. False positives rate can erode trust in security scanners, affecting overall reliability of the system. Addressing and reducing false positive rates is crucial for improving the performance of Dos/DDoS attack detection methods. Achieving low false alarm rates while maintaining high attack recognition is a significant challenge in intrusion detection systems.

The performance metrics described in Tables 5.6 and 5.7 are used to evaluate the effectiveness of machine learning classifiers in intrusion detection system. The following equations are calculating the false positive rate and accuracy of a system.

$$False\ Positive\ Rate\ (FPR)\ = \frac{Incorrectly\ Normal\ Classified\ Instances}{Total\ Normal\ Instances} \tag{1}$$

$$Accuracy = \frac{Correctly\ Classified\ Instances}{Total\ number\ of\ Instances} \tag{2}$$

In Table 5.6, it appears that a maximum false positive rate of 2.3% specifically in detecting attacks. In Table 5.7, the false positive rate has been further improved, with Random Tree achieving a maximum false positive rate of 1.3% and other classifiers showing reductions in false positive rates.

Table 5.6 FPR with Classifiers in PortScan Attack

Classifiers	**PortScan Attack**			
	TPR	**FPR**	**RPC**	**REC**
Logistic	0995	**0.000**	0.996	0.995
SVM	0.998	**0.000**	0.998	0.998
Naïve Bayes	0.989	**0.001**	0.990	0.989
Bayes Net	0.995	**0.000**	0.996	0.995
J48	0.998	**0.023**	0.998	0.998
Random Tree	0.991	**0.023**	0.991	0.991

In Table 5.8, the accuracy is the lowest for Dos about 97%, and the PortScan is 98%. It demonstrated the superior performance of the SFBIDS dataset.

Table 5.7 FPR with Classifiers in DoS Attack

Classifiers	**DoS Attack**			
	TPR	**FPR**	**RPC**	**REC**
Logistic	0.995	**0.005**	0.995	0.995
SVM	0.993	**0.997**	0.993	0.993
Naïve Bayes	0.996	**0.004**	0.996	0.996
Bayes Net	0.999	**0.001**	0.999	0.999
J48	0.999	**0.001**	0.999	0.999
Random Tree	0.988	**0.013**	0.988	0.988

Table 5.8 Accuracy in DoS and PortScan Attacks

Classifiers	Accuracy (%)	
	DoS	PortScan
Logistic	99.502	99.542
SVM	99.302	99.771
Naïve Bayes	99.601	98.856
Bayes Net	99.900	99.542
J48	99.900	99.771
Random Tree	98.804	99.085

5.6.1 Reduce three Features from SFBIDS Dataset

In performance, the False Positive rate, True Positive rate, Precision, and Recall calculate without considering three general features as Destination IP Address, Protocol, and Services out of sixteen main features. Only SVM has the highest false positive rate of 2.2% in DoS Attack. Only SVM has the highest false positive rate of 2.2%. 7% in Naive Bytes, seen in Table 5.9 that Random Tree and Logistic have 2%, and the remaining Bayes Net and J48 have only 1%.

Table 5.9 Performance of DoS Attack in SFBIDS Dataset

Classifiers	DoS Attack			
	TPR	FPR	RPC	REC
Logistic	0.998	0.002	0.998	0.998
SVM	0.982	0.022	0.983	0.982
Naïve Bayes	0.992	0.007	0.992	0.992
Bayes Net	0.999	0.001	0.999	0.999
J48	0.999	0.001	0.999	0.999
Random Tree	0.998	0.002	0.998	0.998

In PortScan Attack at Table 5.10, the Bayes Net and Naïve Bayes classifiers have 1% and 2%, respectively. The SVM classifier has the highest false positive rate of 81.8%, the second classifier is Logistic 4.6%, and the third classifier is J48 and Random Tree 2.3%, respectively.

Table 5.10 Performance of PortScan Attack in SFBIDS Dataset

Classifiers	**PortScan Attack**			
	TPR	**FPR**	**RPC**	**REC**
Logistic	0.993	0.046	0.993	0.993
SVM	0.918	0.818	0.924	0.918
Naïve Bayes	0.979	0.002	0.983	0.979
Bayes Net	0.993	0.001	0.994	0.993
J48	0.998	0.023	0.998	0.998
Random Tree	0.998	0.023	0.998	0.998

Table 5.11 shows the solution that can calculate the accuracy based on five classifiers from two attacks by removing the three features. The lowest accuracy found in the SVM classifier is 97.81% in DoS and only 52.88% in PortScan. The lowest accuracy in the DoS attack is 99.3% in Naive Bayes and 99.8, and 99.9 in the rest of the classifiers, respectively. Also, in the DoS attack, the least accuracy found in Logistic is 95.57%, and the other classifiers are 99.9, 99.8, and 97.75 obtained, respectively. After removing the general three features, except for the SVM classifier, the rest of the classifiers were found to be good at calculating accuracy.

Table 5.11 Accuracy with Both Attacks

Classifiers	**Accuracy (%)**	
	DoS	**PortScan**
Logistic	99.801	99.314
SVM	98.205	91.762
Naïve Bayes	99.202	97.941

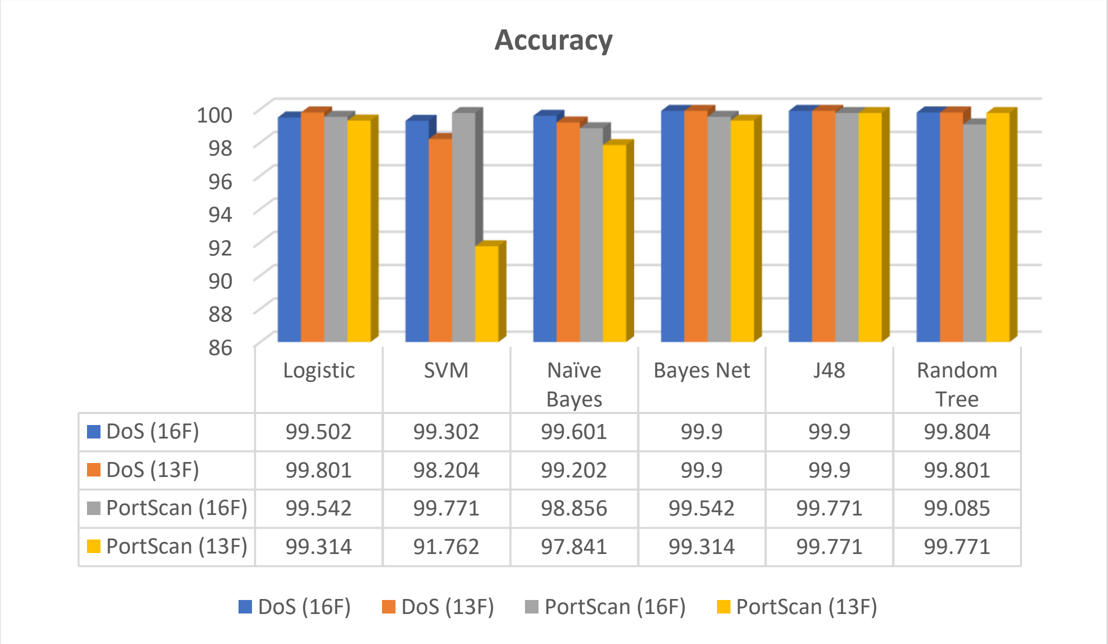
Bayes Net	99.9	99.314
J48	99.9	99.771
Random Tree	99.801	99.771

A Correctly Classified Instance (CCI) refers to an observation or data point is accurately assigned to its truly class or category by a model. The Incorrectly Classified Instance (ICI) is the wrong class by the model. The instances of CCI and CIC calculated using five classifiers are shown in Table 5.12 after randomly removing three features in the proposed dataset.

Table 5.12 Three Remove Features with Instances Result on Two Attacks

Attacks	**Classifiers**	**Correctly Classified Instances (%)**	**Incorrectly Classified Instances (%)**
DoS	Logistic Regression	99.801	0.199
	SVM	99.205	1.795
	Naïve Bayes	99.202	0.798
	Bayes Net	99.9	0.1
	J48	99.9	0.1
	Random Tree	99.801	0.199
Portscan	Logistic Regression	99.314	0.687
	SVM	91.762	8.238
	Naïve Bayes	97.941	2.059
	Bayes Net	99.314	0.687
	J48	99.771	0.229
	Random Tree	99.771	0.229

The Figure 5.4 shows the comparison of the accuracy obtained by calculating full features (16F) and thirteen features (13F) with six classifiers. If the user checks the accuracy of each classifier, it is not significant in DoS attack in Logistic, but 3.9% less accuracy in thirteen features in PortScan. In the SVM classifier, the accuracy of thirteen features drops to 2.2% in DoS and 46% in PortScan compared with full features. Naive Bayes found a 0.5% reduction in DoS and the 0.2% reduction in PortScan from thirteen features. In the Bayes Net classifier, there is no difference at all. It was found that J48 classifier has 2.15% less accuracy from full features in DoS and 2.15% less accuracy from thirteen features in PortScan. In the Random Tree Classifier, the 2% of the full features in the DoS attack is less than the thirteen features and the same accuracy in the PortScan attack.



	Logistic	SVM	Naïve Bayes	Bayes Net	J48	Random Tree
■ DoS (16F)	99.502	99.302	99.601	99.9	99.9	99.804
■ DoS (13F)	99.801	98.204	99.202	99.9	99.9	99.801
■ PortScan (16F)	99.542	99.771	98.856	99.542	99.771	99.085
■ PortScan (13F)	99.314	91.762	97.841	99.314	99.771	99.771

Figure 5.4 Comparison of Full Features and Three Removed Features

5.7 Summary

When leveraging a combination of the above machine learning classifiers within the firewall system, organizations can enhance their capabilities for intrusion detection, and classification. In the proposed dataset, the accuracy is good for the Full features. It seems that the accuracy of the SVM classifier and Logistic among the thirteen features has significantly decreased, and the remaining classifiers have an average accuracy of only 2%. It has the benefit of improving system performance by reducing irrelevant and unessential features. The firewall rules are based on the organization's context to enhance security by reducing false positives, improving accuracy, and maintaining optimal network performance.

CHAPTER 6
EVALUATION OF SFBIDS AND CICIDS2017

The chapter is compared on performance as minimum false positive rate and maximum false positive rate of the proposed dataset and CICIDS2017 dataset. This chapter will demonstrate using machine learning classifiers how the inclusion and exclusion of the flag features can affect the performance. Both the SFBIDS dataset and CICIDS2017 dataset comparison section do not consider the Flag features with CBFS that limited features to analyze. Using CBFS in the CICIDS2017 dataset, features are selected and compared in this chapter focusing on correctly classified instance and false positive rate.

The network traffics are classified as normal or attacks in the existing testbed environment based on six machine learning classification methods applied in the system. It is required to be tested to get datasets and applied for DoS and PortScan. This system tested random extracted 26 features from the CICIDS2017 dataset. In [47], the both datasets compare with only DoS attack by five machine learning classifiers but not use feature selection method.

The system is reducing the complexity times and, system resources by using feature-selection method as CBFS and CA of Filter-Method. The SFBIDS dataset compares with CICIDS2017 that the performance improves without considering the Flag features. The performance will calculate with the CBFS method and compare with and without considering Flag features in CICIDS2017. When using the CA Method, the minimum boundary value is determined by taking the average value of the two datasets based on the trains of the features. It finds the good features that extract based on the destination host of the desired traffic. The system is to reduce false positive rates and to improve accuracy in the implemented testbed design. The system also proves good performance by selecting important features and comparing existing a dataset.

6.1 Random Selected Features from CICIDS2017

The existing dataset [43] extracted the 26 random features in Table 6.1. In [47], the comparison of both datasets proved the accuracy with Postscan attack only. The table 6.2 proved the performance of True Positive (TPR), False Positive (FPR), Precision (PRC) and, Recall (REC) for DoS attack. And also, Portscan attack result with machine learning classifiers in Table 6.3. The random selected features do not consider the twelve flag features in Table 6.4.

Table 6.1 Extracted 26 Features from CICIDS2017

No	Random Features Extracted from CICIDS2017
1	Destination Port
2	Flow Duration
3	Total Fwd Packets
4	Total Backward Packets
5	Total Length of Fwd Packets
6	Total Length of Bwd Packets
7	Fwd Packet Length Max
8	Fwd Packet Length Min
9	Fwd Packet Length Mean
10	Fwd Packet Length Std
11	Bwd Packet Length Max
12	Bwd Packet Length Min
13	Bwd Packet Length Mean
14	Bwd Packet Length Std
15	Bwd Header Length
16	Fwd Packets
17	Bwd Packets
18	Min Packet Length
19	Max Packet Length
20	Packet Length Mean
21	Packet Length Std
22	Packet Length Variance
23	Average Packet Size
24	Avg Fwd Segment Size
25	Avg Bwd Segment Size
26	Fwd Header Length

The system performance is calculated in Table 6.2 with 26 randomly selected features without using the feature selection method. The random 26 features result proved the performance as false positive. The maximum false positive rate is 1.8% for

Logistic Regression classifier and the minimum false positive rate is 0.1% for J48 classifier in DoS attack.

Table 6.2 DoS Random Extracted 26 Features Result

Classifiers	**TPR**	**FPR**	**RPC**	**REC**	**Correctly Classify (%)**	**Incorrectly Classify (%)**
Logistic	0.978	0.018	0.997	0.978	97.778	2.221
Naïve Bayes	0.669	0.007	0.952	0.669	66.933	33.067
Bayes Net	0.9212	0.010	0.986	0.912	91.153	8.847
J48	0.998	0.001	0.998	0.998	99.825	0.175
Random Tree	0.998	0.002	0.998	0.998	99.794	0.206

6.2 Correlation Based Feature Selection (CBFS) on CICIDS2017

The Twelve Flag features from 78 features of CICIDS2017 described in the following Table 6.3.

Table 6.3 Twelve Flag Features from CICIDS2017

No.	**CICIDS2017 Flag Features**
1	FIN Flag Count
2	SYN Flag Count
3	RST Flag Count
4	PSH Flag Count
5	ACK Flag Count
6	URG Flag Count
7	CWE Flag Count
8	ECE Flag Count
9	Fwd PSH Flags
10	Bwd PSH Flags
11	Fwd URG Flags
12	Bwd URG Flags

6.2.1 Full Features with CBFS on DoS Attack

In CICIDS2017, The DoS attacks have four selected features by correlation based feature selection method in Table 6.4. In all features, the Fwd Header Length is the duplicate feature that cause the complexity of calculation. Therefore, it features removed in the existing dataset.

Table 6.4 DoS Features of CICIDS2017 by CBFS

CICIDS2017	Selected Features
All Features	Destination Port Total Length of Bwd Packets Init_Win_bytes_forward Idle Max
Removed Flags	Destination Port Total Length of Bwd Packets Init_Win_bytes_forward Idle Max

CICIDS2017's DoS attack uses full features including the flag features and the result is the same as not using the flag features in Correlation based feature selection in table 6.5. The first maximum false positive rate is 27.3% at Logistic Registration and the second is 2.6% at Naïve Bayes. The minimum false positive rate is 8%, which can be found in J48, Random Tree and, Random Forest classifiers respectively. Similarly, the participation percentages of correctly and incorrectly classified instances can be clearly seen in these three classifiers.

Table 6.5 DoS Extracted Features Result by CBFS

Classifiers	TPR	FPR	RPC	REC	Correctly Classify (%)	Incorrectly Classify (%)
Logistic	0.825	0.273	N/A	0.825	82.485	17.515
Naïve Bayes	0.886	0.026	0.930	0.886	88.580	11.421
Bayes Net	0.983	0.009	0.983	0.983	98.259	1.741
J48	0.992	0.008	0.992	0.992	99.214	0.786
Random Tree	0.992	0.008	0.992	0.992	99.223	0.777

6.2.2 Full Features of CBFS on PortScan Attack

Using CFS in the Portscan attack and selecting the effective features, it was found that five features were selected in all features. In addition, without considering the Flag features, when using CBFS to select the features, PSH Flag Count is not available, and the remaining four features are selected, as can be seen in Table 6.6.

Table 6.6 PortScan Features of CICIDS2017 by CBFS

CICIDS2017	Selected Features
All Features	Bwd Packet Length Mean PSH Flag Count Init_Win_bytes_backward act_data_pkt_fwd min_seg_size_forward
Removed Flags	Bwd Packet Length Max Init_Win_bytes_forward act_data_pkt_fwd min_seg_size_forward

In Portscan attack, 97% is the minimum correctly classify instance and the maximum is 97% that for CICIDS2017. The false positive rate is maximum 3.4% and min 2% in all features of Portscan attack in Table 6.7.

Table 6.7 PortScan Extracted Features Result with all Features by CBFS

Classifiers	TPR	FPR	RPC	REC	Correctly Classify (%)	Incorrectly Classify (%)
Logistic	0.992	0.008	0.992	0.992	99.245	0.755
Naïve Bayes	0.971	0.034	0.972	0.971	97.097	2.903
Bayes Net	0.993	0.007	0.993	0.993	99.317	0.684
J48	0.998	0.002	0.998	0.998	99.827	0.174
Random Tree	0.998	0.002	0.998	0.998	99.838	0.162

6.2.3 Remove Flag Features with CBFS

When Portscan attack does not include the Flag features, the false positive rate results are the same except for Logistic Regression and Naïve Bayes classifiers. In Naïve Bayes, the false positive is about 43%, and as a correctly classified instance, it can see significantly less in table 6.8.

Table 6.8 PortScan Extracted Features Remove Flags by CBFS

Classifiers	TPR	FPR	RPC	REC	Correctly Classify (%)	Incorrectly Classify (%)
Logistic	0.912	0.109	0.923	0.912	91.207	8.793
Naïve Bayes	0.656	0.429	0.786	0.656	65.601	34.399
Bayes Net	0.995	0.004	0.995	0.995	99.506	0.494
J48	0.998	0.002	0.998	0.998	99.769	0.231
Random Tree	0.998	0.002	0.998	0.998	99.773	0.227

6.2.4 Existing Dataset with and without Flag Features

The table 6.9 shows the same result of false positive rate of DoS attack with and without considering the **Flag features** in CICIDS2017 by using correlation-based feature selection method. In the Portscan attack, except for Logistic Regression and Naïve Bayes, the remaining classifiers have similar false positive rate reduction. It can be seen that not including the flag fields reduces the false positive in Portscan's Bayes Net. Therefore, the flag fields are not considered to reduce the calculation time and computer resources consumption when comparing the performance of proposed dataset and CICIDS2017 dataset.

Table 6.9 CICIDS2017 With Flags and Without Flags Field by CBFS

Classifiers	False Positive Rate (With Flag)		False Positive Rate (Without Flag)	
	DoS	PortScan	DoS	PortScan
Logistic	0.273	0.008	0.273	0.109

Naïve Bayes	0.026	0.034	0.026	0.429
Bayes Net	0.009	0.007	0.009	0.004
J48	0.008	0.002	0.008	0.002
Random Tree	0.008	0.002	0.008	0.002

6.3 Comparison of SFBIDS and CICIDS2017 Dataset by CBFS

Among the 78 features in CICIDS2017 [58], flag features are not considered, and the remaining features are considered. The CICIDS 2017 has 78 features. If we remove the flag features, there are 65 features, and depending on it, the selected features is calculated by CBFS.

In table 6.4 and 6.6, the CBFS features selection that choose four features as Destination Port, Total Length of Bwd Packets, Init_Win_bytes_forward, and Idle Max for DoS attacks and also takes four features as Bwd Packet Length Mean, Init_Win_bytes_backward, act_data_pkt_fwd, and min_seg_size_forward for Portscan attack respectively. In table 6.9, it shows the results of false positive rate with six machine learning classifiers.

The machine learning classifiers calculate the False Positive Rate (FPR) from the result of Correlation Based Feature Selection (CBFS) method. The SFBIDS dataset of the minimum false positive rate is 1% and 0% for DoS and Portscan. The minimum false positive in CICIDS2017, 8% for Dos and 2 % for Portscan.

In table 6.10, the maximum false positive rate of SFBIDS and CICIDS2017 is 4% of Naïve Bayes classifier in DoS and 2% of Logistic Regression classifier in Postscan. It can be seen that the maximum FP rate is only 4%. In CICIDS2017, it can be seen that only Naïve Bayes classifier decrease false positive rate and the rest of the classifiers that the false positive rate increases significantly.

Table 6.10 Comparison of FPR from Two Datasets

Classifiers	**FPR of Proposed**		**FPR of CICIDS2017**	
	DoS	**PScan**	**DoS**	**PScan**
Logistic	0.001	0.023	0.273	0.109

Naïve Bayes	0.004	0.000	0.026	0.429
Bayes Net	0.000	0.000	0.009	0.004
J48	0.001	0.000	0.008	0.002
Random Tree	0.000	0.000	0.008	0.002

6.3.1 Accuracy of SFBIDS and CICIDS2017 Dataset with CBFS

The accuracy of SFBIDS and CICIDS2017 dataset showed in Table 6.11. The accuracy of the SFBIDS dataset is significantly higher than the existing CICIDS2017 dataset applied in Machine Learning. Comparing the SFBIDS to existing dataset, in CICIDS2017 it is not obvious in the rest of classifiers, it can see a significant decrease in Logistic Regression and Naïve Bayes classifiers.

Table 6.11 Comparison of Accuracy from Two Datasets

Classifiers	Accuracy of Proposed (%)		Accuracy of CICIDS2017 (%)	
	DoS	PScan	DoS	PScan
Logistic	99.9	99.542	82.485	91.207
Naïve Bayes	99.502	99.508	88.580	65.601
Bayes Net	100	99.542	98.259	99.506
J48	99.9	100	99.223	99.769
Random Tree	100	100	99.234	99.773

6.4 Comparison of SFBIDS and CICIDS2017 Datasets by CA

Comparison of SFBIDS and CICIDS2017 Dataset by correlation Attribute (CA) method from the 78 full features without redundant feature show in Table 6.12, and 28 features are listed. From the 16 features in the proposed dataset, the eleven features acquired by setting and calculating the average boundary value of the solution obtained using the CA method. In determining the boundary value in the Proposed and CICIDS2017 Dataset, the average value calculates by adding the train of the features of the two datasets. The boundary value is set to 0.152955 for DoS and 0.07995 for PortScan attack.

Table 6.13 shows the performance of the false-positive rate and accuracy of the proposed and CICIDS2017 with percentages. If there were to express the FPR for each classifier for both datasets, it can see that J48 is 0.1% the same and the accuracy is almost the same. It can be seen that the NB classifier has an FPR of 9.8% in CICIDS2017, while the proposed one has only 0.4%. The RT classifier has an FPR of 0.2% in CICIDS2017 and 2.6% in the SFBIDS. The LG and NB classifiers have FPR 1.1% and 1.5% in CICIDS2017, while the proposed has only 0.5% and 0.2%, respectively. In the SFBIDS dataset, the accuracy of the RT classifier is close to 98%, and the other classifiers are above 99%. In the CICIDS2017 dataset, the accuracy is above 99% in J48 and RT; BN and LG have 98%, while the NB classifier has more than 79%, which shows significantly less accuracy.

Table 6.12 CA Method Result of CICIDS2017 Dataset in DoS

No	Feature Code	Features	No	Feature Code	Features
1	13	Bwd Packet Length Mean	15	41	Average Packet Size
2	43	Avg Bwd Segment Size	16	18	Flow IAT Std
3	14	Bwd Packet Length Std	17	21	Fwd IAT Total
4	11	Bwd Packet Length Max	18	2	Flow Duration
5	23	Fwd IAT Std	19	12	Bwd Packet Length Min
6	38	Packet Length Std	20	22	Fwd IAT Mean
7	64	Idle Max	21	35	Min Packet Length
8	62	Idle Mean	22	17	Flow IAT Mean
9	24	Fwd IAT Max	23	28	Bwd IAT Std
10	19	Flow IAT Max	24	1	Destination Port
11	65	Idle Min	25	29	Bwd IAT Max
12	36	Max Packet Length	26	40	Down/Up Ration
13	37	Packet Length Mean	27	8	Fwd Packet Length Min
14	39	Packet Length Variance	28	33	Fwd Packets/s

Table 6.13 Performance Comparison of DoS Attack with CA Method in DoS

Detection Classifiers	CICIDS2017 CA Method			Proposed CA Method		
	Features - Code	FPR	Acc (%)	Features No:	FPR	Acc (%)
LG	1,2,8,11,12, 13,14,17,18,19, 21,22,23,24,28, 29,33,35,36,37, 38,39,40,41,43, 62,64,65	0.011	98.825	1,2,3, 4,7,8,10, 11,13,14,15	0.005	99.502
NB		0.098	79.546		0.004	99.502
BN		0.015	98.184		0.002	99.801
J48		0.001	99.827		0.001	99.9
RT		0.002	99.984		0.026	97.906

6.5 Evaluation

The SFBIDS dataset creates sixteen features, and the goodness of these features proves the performance as a false-positive rate (FPR) and accuracy with machine learning classifiers. A high false positive rate is not a real attack, but an alert, so the security of the organization may be affected by not being aware of the intruder's attack. Therefore, it can see the reduction of the false-positive rate in this system.

In CICIDS2017, considering and not considering the 13 flag features achieves the same performance in a DoS attack. Table 4 (a and b) shows that there is only a slight change in the PortScan attack and no impact on performance. If the attributes related to the 13 flag features and the values are not considered, the calculation time and complexity time for performance is significantly reduced. The CICIDS2017 dataset using 28 features has an accuracy of 99.83%, and the accuracy of the proposed dataset using 11 features is 99.9%. It can be found in the J48 classifier of Table 7.

In [24], by setting the values of feature weight with CICIDS2017 dataset and using 4, 15, 22, 35, 52, 77 features and calculating performance with five classifiers, the highest accuracy is 99.87% in J48 classifier with 52 features, 99.86% in Random Forest classifier with 22 features, and 99.79% in Random Tree classifier with 15 features. When using many features, the execution time is significantly increased, but the accuracy is not seriously improved.

When the SFBIDS dataset implement that the quality of a feature selected rather than the instance involved implementing a dataset depends on the value of that feature. Data quality directly impacts the effectiveness and efficiency of a dataset, highlighting the significance of ensuring high-quality data for optimal machine learning outcomes. The performance of the system is determined by the value different and validity of the selected features on normal, DoS and Portscan attack traffics. The SFBIDS dataset can further reduce the false positive rate compare with the existing CICIDS2017 dataset with machine learning classifiers.

Table 6.14 Evaluation with CICIDS2017 in DoS

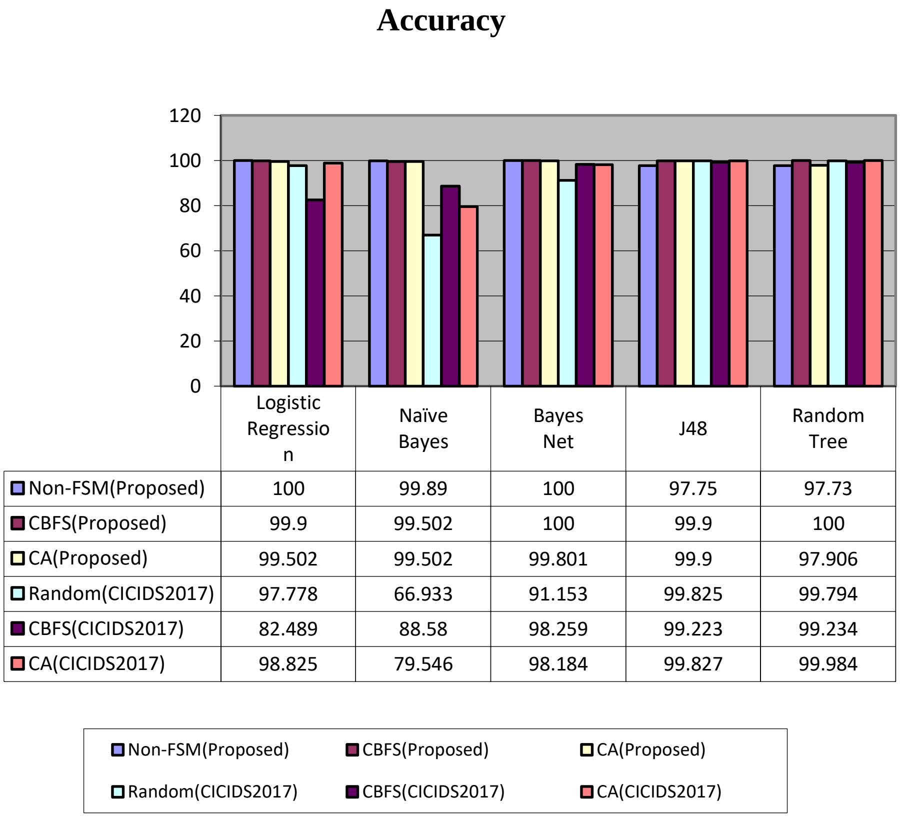
Classifiers	**False Positive Rate**			**Accuracy**		
	Random	**CBFS**	**CA**	**Random**	**CBFS**	**CA**
Logistic	0.018	0.273	0.011	97.778	82.485	98.825
Naïve Bayes	0.007	0.026	0.098	66.933	88.580	79.546
Bayes Net	0.010	0.009	0.015	91.153	98.259	98.184
J48	0.001	0.008	0.001	99.825	99.223	99.827
Random Tree	0.002	0.008	0.002	99.794	99.234	99.984

Table 6.14 shows the performance of CICIDS2017 using four features in CBFS, 28 features in CA, and 26 features in random features. Except for the Naive Bayes Classifier, the rest of the classifiers have good performance and are found to be above 98%. Table 6.15 shows the performance of the SFBIDS dataset using sixteen features in Non-FSM (Not used Feature Selection Methods), five features in CBFS, and eleven features in CA. In the SFBIDS dataset, eleven features are stable and Accuracy are found to be above 99% except for Random Tree classifiers. In evaluating Table 6.14 and Table 6.15, it is observed that the performance of the SFBIDS dataset using only eleven features is superior. Figure 6.1 compares the accuracy of the SFBIDS dataset and CICIDS2017 on the three conditions.

In addition, comparing the SFBIDS dataset and the CICIDS2017 dataset, it wants to focus on the goodness of the features of the SFBIDS dataset rather than CICIDS2017.

Table 6.15 Evaluation with FSBIDS in DoS

Classifiers	**False Positive Rate**			**Accuracy**		
	Non-FSM	**CBFS**	**CA**	**Non-FSM**	**CBFS**	**CA**
Logistic	0.005	0.001	0.005	100	99.9	99.502
Naïve Bayes	0.997	0.004	0.004	99.89	99.502	99.502
Bayes Net	0.004	0.000	0.002	100	100	99.801
J48	0.001	0.001	0.001	97.75	99.9	99.9
Random Tree	0.001	0.000	0.026	97.73	100	97.906



	Logistic Regressio n	Naïve Bayes	Bayes Net	J48	Random Tree
Non-FSM(Proposed)	100	99.89	100	97.75	97.73
CBFS(Proposed)	99.9	99.502	100	99.9	100
CA(Proposed)	99.502	99.502	99.801	99.9	97.906
Random(CICIDS2017)	97.778	66.933	91.153	99.825	99.794
CBFS(CICIDS2017)	82.489	88.58	98.259	99.223	99.234
CA(CICIDS2017)	98.825	79.546	98.184	99.827	99.984

Figure 6.1 Comparative Accuracy of Both Datasets

6.6 Summary

Comparing of the proposed dataset and existing dataset CICIDS2017 is only to measure the good features of the system. Unnecessary features can significantly impact performance, consuming valuable CPU resources during calculations. In CICIDS2017, the performance of the dataset is disturbed due to the redundant feature in 78 features. Machine learning classifiers can be employed to effectively reduce false positive rate, enhancing the efficiency of systems. By reducing the unnecessary features and their values, the false positive rate and processing time will be reduced and the system will be effective.

CHAPTER 7
CONCLUSION AND FUTURE WORKS

Network security is a critical in safeguarding information and systems across various sectors, including education, government, and business. In educational institutions, sensitive student data and research information store digitally, making them potential targets for cyber threats. Government agencies handle vast amounts of sensitive data, including citizen records and national security information. Ensuring robust network security in government prevents unauthorized access, data breaches, and potential cyber-attacks that may compromise national security. So, Network security is essential in protecting businesses from data breaches, financial fraud, and disruptions to operations, preserving the integrity and trust of stakeholders. The SFBIDS employs machine learning techniques to classify network traffic, considering temporal aspects and selecting importance features. This approach ensures a comprehensive evaluation of the capabilities of the system in distinguishing between normal and malicious network activities by a particular emphasis on DoS and PortScan attacks.

The SFBIDS dataset for intrusion detection employs advanced feature selection techniques to identify implicit features. Notably, GR is utilized for feature selection, enhancing the ability of the system to discern relevant information. Additionally, CBFS and CA are employed to maximize the relevance between input features and the output, further improving the detection of intrusion systems. This approach aims to optimize the dataset by selecting features that contribute significantly to the accurate discovery of potential intrusions.

The system focuses on feature superiority and to achieve the goal of comparing False Positive Rates with the CICIDS2017 dataset using machine learning classifiers. The system classifies network traffic into normal patterns and potential attacks using the selected machine learning methods. The dataset used for testing and training the machine learning models is based on CICIDS2017. This dataset is widely recognized in the cybersecurity and provides a foundation for evaluating the system's performance.

Implementing a software-based open source firewall offers several advantages, including complexity reduction, time efficiency, adaptability in configuration, and cost-effectiveness. However, it's essential to exercise caution, as the filtering rules established by the firewall can inadvertently create network vulnerabilities. Misconfigurations or improper order of firewall rules may expose the network to potential security risks. Therefore, meticulous attention to rule configuration and order is paramount to ensure the firewall effectively safeguards the network without introducing vulnerabilities. The misconfiguration of firewall rules can inadvertently expose the network to vulnerabilities or disrupt essential services [5].

A network testbed established to simulate real-world network conditions, comprising a firewall and an IDS. In the existing testbed environment, network traffic subject to classification as usual or attacks with the implementation of six machine learning classification methods. The system focused to testing and applying these methods specifically for DoS and PortScan attacks. The dataset utilized for this evaluation from CICIDS2017, with additional features incorporated to enhance the classification process.

The IDS all the time monitors network traffic for suspicious activities, providing the security against potential threats. A dataset is created using both DoS traffic and normal traffic within the testbed environment, enabling the evaluation of the response of the system to various scenarios.

By combining these elements, the system aims to understand and address the challenges posed by firewall rule misconfigurations, enhance network security, and effectively mitigate the risks associated with potential attacks. In the network traffic classification, machine learning methods play a pivotal role, enabling the recognize of importance features based on the temporal dimension.

7.1 Advantages and Limitation of the SFBIDS System

A software-based firewall offers several benefits. The first one is flexibility, the second one is cost-effectiveness, and the third one is scalability. In flexibility, Software Firewalls can be installed on individual devices, providing flexibility in configuration and management. Cost-effectiveness: they are typically more affordable compare hardware firewalls, making them suitable for smaller businesses or individual users. Scalability: it is easier to scale software firewalls to accommodate in network size or configuration.

IDS can detect unauthorized access attempts and potential security breaches early, allowing for prompt mitigation before significant damage occurs. By alerting security teams to suspicious activities, IDS facilitates swift incident response, enabling organizations to contain and neutralize threats effectively. IDS provides insight into network traffic and activity, helping organizations understand their network environment better and identify potential vulnerabilities. IDS aids in meeting regulatory requirements by providing evidence of security measures and incident response capabilities. IDS offers centralized management for correlating distributed attacks, simplifying the monitoring and planning of security events.

In theory, the more of the data improves the accuracy of the outcomes. However, it is crucial to note that simply having more data does not guarantee better results; the quality and relevance of the data also play significant roles.

7.2 Summary and Future Work

In summary, network security becomes crucial in education, government, business, and other sectors with related network connections to safeguard sensitive information, ensure privacy, and maintain the integrity of operations. The system leverages machine learning methods to classify traffic based on temporal aspects, selecting importance features for more accurate and efficient traffic analysis. A meticulous process will manage to create a high-quality dataset based on features and their values. The involve selecting, refining, and optimizing features is to improve the overall dataset quality. Special attention is given to testing and applying the classification methods specifically for DoS and PortScan attacks. Various intrusion detection algorithms incorporate into the system. This approach aims to diversify the detection mechanisms, ensuring that the IDS is proficient in identifying different types of attacks.

In summary, the future work in the network system involves a multifaceted approach, including dataset extension, feature-based refinement, the inclusion of new attacks, and the adoption of diverse intrusion detection algorithms. These strategies collectively contribute to enhancing the performance and adaptability of the IDS System. The system will detect a broader range of two attacks, if the new attack types can add to the dataset to enhance the IDS's capability.

AUTHOR'S PUBLICATIONS

[P1] Security Awareness of Network Infrastructure: Real-time Intrusion Detection and Prevention System with Storage Log Server, **16th International Conference on Computer Applications** (ICCA, 2018), Yangon, **Myanmar**, February, 2018. **Page [469-474]**

[P2] A Framework for Secure Network Infrastructure: Intrusion Prevention System Based on Firewall Rules, The **12th Conference on Project Management** (ProMAC2018), **Thailand**, **ISBN: 978-4-902378-63-4**, November, 2018. **Page [125-132]**

[P3] Awareness of Policy Anomalies with Ruled-Based Firewall, **The 13th Conference on Project Management** (ProMAC2019), Yangon, **Myanmar, ISBN: 978-4-902378-69-6**, November, 2019. **Page [678-686]**

[P4] Machine Learning Based DoS Traffic Analysis on the Testbed Environment**, 19th International Conference on Computer Applications (ICCA2021)**, IEEE, Yangon, **Myanmar**, February, 2021. **Page [265-270]**

[P5] Performance Analysis of Traffic Classification with Machine Learning, **International Conference on Information Technology and Electrical Engineering** (ICITEE2021), **Australia**, **Online ISSN: 1307-6892**, February, 2021. **Page [33-38]**

[P6] Performance Analysis of Traffic Classification with Machine Learning, **Engineering and Technology International Journal of Computer and Information Engineering, Australia**, Vol:15, No:1, February, 2021.

[P7] Comparison of Effective Features Selection Method in Intrusion Detection System with Testbed, **2022 International Conference on Communication and Computer Research** (ICCR2022), **Korea**, October, 2022.

[P8] Enhancing Performance of Traffic Classification with Feature Selection Methods" **Indian Journal of Computer Science and Engineering** (IJCSE), **India**, **e-ISSN: 0976-5166, p-ISSN: 2231-3850** Vol:15, No:2, March, 2024.

BIBLIOGRAPHY

[1] Muhammad Abedin, Syeda Nessa, Latifur Khan, and Bhavani Thuraisingham, “Detection and Resolution of Anomalies in Firewall Policy Rules”, Data and Applications Security 2006, pp. 15-29, 2006.

[2] Sandhya Peddabachigari, Ajith Abraham*, and Johnson Thomas, “Intrusion Detection Systems Using Decision Trees and Support Vector Machines”, International Journal of Computer Application, June, 2010.

[3] N. Akhyari , and S.Fahmy, “Design a Network Security Tool Using Open Source Appliction”, Australian Journal of Basic a And Application Science, April, 2014.

[4] Elie Alhajjar, Paul Maxwell, and Nathaniel Bastian, “Adversarial machine learning in Network Intrusion Detection Systems”, Expert Systems with Applications (ELSEVIER), August, 2021.

[5] M. Alicea and I. Alsmadi, “Misconfiguration Firewalls and Network Access Controls: Literature Review”, Computer Information Systems Faculty Publications, October, 2021.

[6] Ehab S. Al-Shaer and Hazem H. Hamed, “Firewall Policy Advisor for Anomaly Discovery and Rule Editing”, FIP/IEEE Eighth International Symposium on Integrated Network Management, March, 2003.

[7] Ehab S. Al-Shaer and Hazem H. Hamed, “Discovery of Policy Anomalies in Distributed Firewalls”, IEEE INFOCOM, April, 2004.

[8] Ehab S. Al-Shaer and Hazem H. Hamed, “Modeling and Management of Firewall Policies,” Transaction on Network and Service Management, IEEE, 2004.

[9] Feature Selection Techniques in Machine Learning. [Online] Available at https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/ (Accessed 25.1.2024)

[10] Ed Burns, “Machine Learning”, [Online] Available at https://searchenterpriseai.techtarget.com/definition/machine-learning-ML. (Accessed: 17.7.2021)

[11] le Cessie, S. and van Houwelingen, J.C. (1992). “Journal of the Royal Statistical Society. Series C (Applied Statistics)”, Ridge Estimators in Logistic Regression. Applied Statistics, Vol. 41, pp. 191-201, 1992.

[12] D. Chapman and E. Zwicky. Building Internet Firewalls, Second Edition, Orielly & Associates Inc., 2000.

[13] Classification in Machine Learning [Online] Available at https://www.edureka.co/blog/classification-in-machine-learning. (Accessed: 13.5.2022)

[14] Hassan Gobjuka, and Kamal A. Ahmat, “Fast and Scalable Method for Resolving Anomalies in Firewall Policies”, 14th IEEE Global Internet Symposium (IEEE INFOCOM2011), 2011.

[15] Joseph Guarino, “The Perfect Linux Firewall IPCop”, [Online] Available at https://www.howtoforge.com/perfect_linux_fire8wall_ipcop. (Accessed: 11.7.2021)

[16] Filip Hock, Peter Kortis, “Commercial and open-source based Intrusion Detection System and Intrusion Prevention System (IDS/IPS) design for an IP networks”, Conference: 2015 13th International Conference on Emerging eLearning Technologies and Applications (ICETA), Nov, 2015.

[17] IDS vs IPS vs Firewall [Online] Available at

https://community.fs.com/blog/ids-vs-ips-vs-firewallhow-to-choose-.html (Accessed 23.1.2024)

[18] Hongxin Hu, Gail-Joon Ahn and Ketan Kulkarni, “FAME: A Firewall Anomaly Management Environment”, ACM, October, 2010.

[19] Alan jeffrey, and Taghrid Samak, “Model Checking Firewall Policy Configurations”, 2009.

[20] Alison Grace Johansen, “Emerging Threats” [Online] Available at https://us.norton.com/internetsecurity-emerging-threats-what-is-firewall.html. (Accessed: 14.7.2022)

[21] S. Jungsuk, T, Hiroki, and O. Yasuo, “Statistical nalysis of Honeypot Data and Building of Kyoto 2006+ Dataset for NIDS Evaluation”, 1st Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS 2011), April, 2011.

[22] S. Jungsuk, T, Hiroki, and O. Yasuo, “Statistical nalysis of Honeypot Data and Building of Kyoto 2006+ Dataset for NIDS Evaluation”, 1st Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS 2011), April, 2011.

[23] Koushal Kumar, and Jaspreet Singh Batth, “Network Intrusion Detection with Feature Selection Techniques using Machine-Learning Algorithms”, International Journal of Computer Applications, Vol 150 – No.12, September, 2016.

[24] Kurniabudi, D. Stiawan, and et al. “CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection”, IEEE, July, 2019.

[25] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and Ali A. Ghorbani ,“Characterization of Tor Traffic using Time based Features”, In Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017), pp. 253-262, 2017.

[26] George Lawton, “Logistic Regression”, [Online] Available at https://www.geeksforgeeks.org/understanding-logistic-regression. (Accessed: 17.7.2021)

[27] Y. Li, J. Xia, Silan Zhang, Jiakai Yan, and et. al “An efficient intrusion detection system based on support vector machines and gradually feature removal method”, Expert System with Applications, pp. 424-430, 2012.

[28] Andrew Lockhart, “Network Security Hacks”, O’Reilly Media, Inc., April 2004.

[29] Malicious Port Scan Attack [Online] Available at https://www.extrahop.com/resources/attacks/malicious-port-scanning. (Accessed: 17.7.2021)

[30] Network Security [Online] Available at https://resources.infosecinstitute.com/topics/network-security-101/firewalls-and-ids-ips/ (Accessed 23.1.2024)

[31] Feature Selection Method and how to choose them. [Online] Avallable at https://neptune.ai/blog/feature-selection-methods (Accessed 25.1.2024)

[32] M. Navarikuth, K. Sachan and R Kumar, "A dynamic firewall architecture based on multi-source analysis", CSIT Publication, December, 2013.

[33] Alireza Osareh, Bita Shadgar, "Intrusion Detection in Computer Networks based on Machine Learning Algorithms", IJCSNS International Journal of Computer Science and Network Security, Vol.8 No.11, November, 2008.

[34] A. Papagrigoriou, P. Petrakis, and M.D. Grammatikakis, "A Firewall Module Resolving Rules Consistency", Proceedings of the 13th Workshop on Intelligent Solutions in Embedded Systems (WISES), IEEE, 2017.

[35] K. Park, Y. Song, and Y.-G Cheong, "Classification of Attack Types for Intrusion Detection Systems using a Machine Learning Algorithm", 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications, pp. 282-286, 2018.

[36] P. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014), Dec, 2014.

[37] Charles P. Pfleeger, S. Lawrence Pfleeger, "Security in Computing", Prentice Hall, Dec2, 2002.

[38] D. Protic, “Review of KDD Cup '99, NSL-KDD and Kyoto 2006+ datasets”, Vojnotehnicki Glasnik/ Military technical Courier, Vol. 66, pp. 560-596, 2018.

[39] B. M. Serinelli, A. Collen, N. A. Nijdam, “On the analysis of open source datasets: validating IDS implementation for well-known and zero day attack detection”, The 18th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), pp. 192-199, August, 2021.

[40] Syed Ali Raza Shah, Biju Issac, “Performance comparison of intrusion detection systems and application of machine learning to Snort system”, Future Generation Computer Systems, ELSEVIER,Vol 80, Page 157-170, March, 2018.

[41] Snort [Online] Available at https://en.wikipedia.org/wiki/Snort_(software). (Accessed: 11.7.2021)

[42] Status Connection of Iptables [Online] Available at https://www.ipcop.org/2-0-0/en/admin/html/status-connections.html. (Accessed: 11.7.2021)

[43] A. Thakkar and R. Lohiya. “A Review of the Advancement in Intrusion detection Datasets”, Procedia Computer Science, Vol-167, pp. 636-645, 2020.

[44] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, Wei-Yang Lin, “Intrusion detection by machine learning: A review”, Expert Systems with Applications (Elsevier), pp. 11994–12000, 2009.

[45] M. Urvashi, and A. Jain, “A survey of IDS classification using KDD CUP 99 dataset in WEKA”, International Journal of Scientific & Engineering Research, Vol.6, Issue 11, November, 2015.

[46] H. H. Yi, Z. M. Aye, “Awareness of Policy Anomalies with Ruled-Based Firewall”, ProMAC 2019, pp. 678-686, November, 2019.

[47] H. H. Yi, Z. May. Aye, “Performance Analysis of Traffic Classification with Machine Learning”, International Conference on Information Technology and Electrical Engineering (ICITEE 2021), pp. 33-38, February, 2021.

[48] Machine Learning [Online] Available at https://en.wikipedia.org/wiki/Machine_learning. (Accessed: 16.7.2021)

[49] What is an Intrusion Detection System? [Online] Available at https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-detection-system-ids (Accessed 23.1.2024)

[50] Naytes Net Classifier https://acodez.in/bayesian-networks-classifiers/ Accessed (12.5.2024)

[51] Naïve Bayes Classifier [Online]

https://www.javatpoint.com/machine-learning-naive-bayes-classifier (Accessed: 13.5.2024)

[52] Available at https://www.ipcop.org/2-0-0/en/admin/html/status-connections.html. (Accessed: 11.7.2021)

[53] IPCop Firewall [Online], Available at https://www.howtoforge.com/perfect_linux_firewall_ipcop. (Accessed: 11.7.2021)

[54] Andrew Lockhart, "Network Security Hacks", O'Reilly Media, Inc., April 2004.

[55] Charles P. Pfleeger, S. Lawrence Pfleeger, "Security in Computing", Prentice Hall, Dec2, 2002.

[56] Snort [Online] Available at https://en.wikipedia.org/wiki/Snort_(software). (Accessed: 11.7.2021)

[57] Lin Zhang, Mengxing Huang, "A Firewall Rules Optimized Model Based On Service-Grouping", 12th Web Information System and Application Conference, IEEE, 2015.

[58] Xinyou Zhang; Chengzhong Li; Wenbin Zheng,"Intrusion Detection System Design", The Fourth International Conference on Computer and Information Technology, IEEE, 2004.

[59] Machine Learning [Online] Available at https://en.wikipedia.org/wiki/Machine_learning. (Accessed: 16.7.2021)

[60] Machine-Learning-ML [Online] Available at https://searchenterpriseai.techtarget.com/definition/machine-learning-ML. (Accessed: 17.7.2021)

[61] Logistic Regression [Online] Available at https://searchbusinessanalytics.techtarget.com/definition/logistic-regression. (Accessed 17.7.2021)

[62] Random Forest [Online] Available at https://en.wikipedia.org/wiki/Random_forest. (Accessed: 30.5.2024)

[63] PortScan [Online] Available at https://www.extrahop.com/resources/attacks/malicious-port-scanning/. (Accessed :13.5.2022)