National College of Ireland

An evaluation of Network Intrusion Detection Systems

Name: Kaweesa James

Student ID: 21226385

School of Computing

National college of Ireland

Supervisor: Micheal Pantridge

National College of Ireland

Project Submission Sheet – 2022/2023

Student Name:	………KAWEESA JAMES ………………………………………………………………		
Student ID:	………………21226385……………………………………………………………………………………………		
Programme	……Cybersecurity………………………………	**Year**	2022/2023………
Module:	………………………………………MSC…………………………………………………………………………		
Lecturer:	………………………………………… Michael Pantridge …………………………………………………		
Submission Due Date:	……………………………………18/09/2023……………………………………………………………………		
Project Title:	An Evaluation of Network Intrusion Detection System		
Word Count:	……		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the library. To use other

author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: …………………KAWEESA JAMES ……………………………………………………………

Date: ……………………………18/09/2023…………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

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

Abstract

In information systems and networks, detecting intrusions is a crucial yet challenging task. This increased the need for effective ways to identify intrusions to safeguard these systems. Existing intrusion detection system (IDS) models have achieved notable performance, but frequently struggle to identify multiple types of attacks due to lengthy classifier training. Comprehensive analysis and evaluation of anomaly-based and signature-based intrusion detection system was conducted**.** This study investigates two Intrusion Detection System (IDS) techniques, signature-based IDS for known attacks and anomaly-based IDS using unsupervised machine learning algorithms, specifically the Isolation Forest. The goal is to develop and implement an efficient IDS, considering performance metrics like precision, recall, and F1-score. The focus is on effectively detecting and responding to both known and unknown attacks to enhance network security and cyber threat prevention.

Keywords: IDS, Anomaly-based, Signature-based, Machine learning algorithms, Cyber-attacks

1. Introduction

Network intrusion detection systems (NIDS) are important tools for enhancing the security posture of organizations by detecting and alerting suspicious or malicious activities within their network environments. IDS can help to prevent cyber-attacks by identifying and responding to security incidents in real-time, providing valuable insights into the activities of potential attackers and allowing organizations to take proactive measures to effectively protect their systems and data. IDS can be highly effective in preventing cyber-attacks and enhancing an organization's security posture, but their effectiveness can vary depending on several factors. Some of the key factors that can impact the effectiveness of IDS include, the accuracy rate of IDS, the quality of the system's configuration and tuning, the type and complexity of the network environment being monitored, and the skill and expertise of the security analysts tasked with monitoring and responding to alerts.

IDS are **an essential component of network security**. They are designed to monitor network traffic for signs of unauthorized access or malicious activity, alerting security personnel when potential threats are detected (Sulaiman *et al.*, 2021). However, not all IDS are created equal, and different systems can have varying levels of accuracy, reliability, and effectiveness in detecting and preventing intrusions. This is why it is important to research the evaluation of IDS to determine how well they perform under different conditions and in different environments. By evaluating different IDS, we can identify their strengths and weaknesses, allowing us to make informed decisions about which system and deployment is best suited for a particular organization or use case**.** Analysis of the effectiveness and performance of host-based IDS running on various platforms and protocols demonstrated that the infrastructure employed may affect how quickly an IDS operates (Bul'ajoul, James and Pannu, 2015). Through research, we can identify areas where IDS performance can be improved, such as through the development of more accurate or efficient algorithms or the use of more advanced machine learning and evaluation techniques. Analysis of how parallel technology and QoS setup are used to improve NIDS performance, the study focused on the potential advantages of these strategies and offered a case study that illustrated their efficiency (Bul'ajoul *et al.*, 2015). As IDS technology continues to evolve, it is important to develop better evaluation methodologies to accurately measure and compare the performance of different systems (Bul'ajoul *et al.*, 2015). This can help us to better understand the capabilities and limitations of IDS and make more informed decisions about their implementation, due to the limitations of both signature-based and anomaly-based IDS techniques in terms of identifying both known and unknown attacks (Hussein, 2016)**.**

Evaluation and analysis of IDS will help researchers and organizations understand the challenges and security risks that come with choosing and deployment of an IDS to minimize cyberattacks. During this research IDS techniques such as anomaly and signature based were identified and evaluated and the deployment strategy discussed. Advancing knowledge in this area will help organization to confidently adopt network security using IDS, in addition to the above, research work in this field gives an insightful information on the performance of IDS to researchers.

The research organization is as follows: section two introduces related work on IDS evolution, types and techniques, strength, and limitation. Section three outlines methodology used when evaluating IDS and tool specifications. Section four presents the experimentation carried out for this research. Section five evaluation of IDS techniques deployed and lastly, section six discussion and conclusion as well as a Gant chart that outlines steps that were followed during this project implementation.

2. Related work

2.1 Evolution of Intrusion detection system techniques.

James Anderson's influential paper, commissioned by a government organization, presented the groundbreaking idea that audit trails hold crucial information that can be instrumental in identifying misuse and gaining insights into user behavior (Deepa and Kavitha, 2012). This paper introduced the concept of detecting misuse and specific user events, marking a significant advancement in the field. Furthermore, his conjecture laid the groundwork for the future design and development of intrusion detection systems (IDS), particularly in host-based intrusion detection (Deepa and Kavitha, 2012). Anderson's work stands as a pivotal milestone in the evolution of IDS.

A government project undertaken by SRI International in 1983, led by Dr. Dorothy Denning, initiated a fresh endeavor in the field of intrusion detection. The project aimed to examine audit trails generated by government mainframe computers and establish user profiles based on their activities (Denning, 1987). Within a year, Dr. Denning played a key role in creating the initial framework for intrusion detection, known as the Intrusion Detection Expert System (IDES). This pioneering model laid the groundwork for subsequent advancements in IDS technology.

Around 1990, Heberlein played a pivotal role as the primary author and developer of the Network Security Monitor (NSM), which was the pioneering network intrusion detection system (Heberlein et al., 1990). NSM was deployed at major government installations, leveraging network traffic analysis to gather vast amounts of information. This new awareness sparked heightened interest in intrusion detection, leading to significant investments in the market. Heberlein's contributions extended beyond NSM to the DIDS project, where he collaborated with the Haystack team to introduce the concept of hybrid intrusion detection (Heberlein et al., 1990). The groundbreaking work of the Haystack project, combined with the introduction of the Network Security Monitor, revolutionized the field of IDS, and propelled it into the commercial world.

The collaboration between Anderson, Heberlein, and Denning pioneered the development of intrusion detection systems (IDS). Supported by government funding and corporate interest, their initial concept transformed into a tangible technology that became crucial for network security. Through continuous advancements, IDS has evolved from a theoretical idea to practical implementation, resulting in the creation of commercially viable tools. In the present day, intrusion detection has become a standard and essential practice for monitoring, detecting, and responding to security threats. It plays a vital role in overall security measures, widely adopted to prevent misuse and safeguard network integrity.

2.2 Intrusion detection system.

There are two main types of IDS, network-based IDS (NIDS) and host-based IDS (HIDS). NIDS monitors network traffic for suspicious activity, while HIDS monitors activity on individual hosts or endpoints. Both types of IDS can be further classified as signature-based or anomaly-based. Signature-based IDS use pre-defined signatures or patterns of known attacks to identify and prevent intrusions, while anomaly-based IDS use machine learning and statistical techniques to detect abnormal behavior (Samrin and Vasumathi, 2017). IDS use various techniques to detect and prevent intrusions, including rule-based, statistical, and machine learning techniques. Rule-based techniques use pre-defined rules or signatures to detect known attacks. Statistical techniques use statistical models to identify abnormal behavior based on data patterns. Machine learning techniques use algorithms and models to learn from historical data and detect anomalies and unknown attacks (Sulaiman *et al.*, 2021; Samrin and Vasumathi, 2017).

2.3 Dataset evaluation

Several studies have proposed methodologies for evaluating intrusion detection systems. One of the most common methodologies is the use of benchmark datasets. Benchmark datasets are essential for evaluating IDS as they provide a standard set of data for testing and comparing different IDS. Network-based datasets are essential for training and evaluating IDS detection methods. Analysis on a well know number of dataset (UNSW-NB 15, CICIDS2017, CSE-CIC-IDS2018, KDD99, NSL-KDD, KYOTO 2006+, ISCX2012, and CIDDS-001) was conducted giving a detailed classes, instance, and features (Ghurab *et al.*, 2021). They furthermore indicated that it is always good to use recent datasets when evaluating IDs since they contain a wide range of network attacks (Ghurab *et al.*, 2021). Similarly, while examining the features of KDD99 and UNSW-NB15 dataset using an association rule mining algorithms to evaluate the complexity based on accuracy and FAR, (Moustafa and Slay, 2015a) research indicated the accuracy of KDD99 was far better than UNSW-NB15 whereas KDD99 far was lower than UNSW-NB15 dataset. Furthermore, they indicated that the UNSW-NB15 data set's evaluation criteria reveal that due to similarities in the values of these records, the decision engine's current algorithms are unable to identify certain record categories (Moustafa and Slay, 2015b).

2.4 Performance evaluation of IDS

Researchers are developing solutions to speed up pattern matching of signature-IDS. However, it is difficult to compare the performance of those solutions because of the difficulties in datasets, metric, and simulation environments. Research carried out characterizing realistic benchmarks for signature-based IDS, by creating a pattern matching engine that can accept any new algorithm, the new engine focused on producing statistics as well as performance pattern (Aldwairi, Alshoul and Seyam, 2018). The comparison methods were based on performing the measurements in the same traffic traces, attack fingerprints, and simulation environment (Aldwairi, Alshoul and Seyam, 2018). Networks and systems are developing quickly in tandem, with the ongoing progress of intrusion detection technology. The main difficulties that IDS encounters are that the intrusion detection system's detection speed needs to be increased to keep up with network communication demands. To increase safety and accuracy, it is essential to lower false negatives and false positives in the intrusion detection system. To increase the safety performance of the entire system, the intrusion detection system's interactive performance must be upgraded (Fang, Tan and Wilbur, 2020). The system's problem is that it is not accurate and occasionally reports an error when none has occurred.

2.5 IDS Evaluation metrics.

When evaluating the effectiveness of an IDS, there are several metrics that can be used to assess its performance as indicated by (Valero León, 2017), they include:

Detection rate this measures the percentage of attacks that are detected by the IDS. A high detection rate is desirable, as it means that the system can catch most attacks.

False positive rate (TP)-these measures the percentage of alerts generated by the IDS that are not actually related to an attack. A high false positive rate can be problematic, as it can lead to many false alarms that must be investigated.

False negative rate (FN) measures the percentage of attacks not detected by the IDS. A high false negative rate is a fundamental problem, as it means that attacks are going undetected and may cause damage to the system.

True positive (TP) refers to a situation where the model correctly predicts a positive instance as positive. In other words, the model accurately identifies an instance as belonging to the positive class when it does belong to that class. It is the correct detection of a positive case.

True negative (TN) occurs when the model correctly predicts a negative instance as negative. In simpler terms, the model accurately identifies an instance as not belonging to the positive class when it does not belong to that class. It is a correct rejection of a negative case.

Accuracy-this is a measure of how well the IDS can distinguish between normal traffic and malicious traffic. A high accuracy rate means that the IDS can correctly identify attacks and avoid false positives.

Response time-these measures the time it takes for the IDS to detect attack and alert security personnel. A fast response time is important, as it can help prevent or mitigate damage caused by an attack.

Resource usage-these measures the amount of system resources, such as CPU and memory, that the IDS requires to operate. Low resource usage is desirable, as it means that the IDS is not putting undue strain on the system.

Although there are different metrics for evaluating IDS (Intrusion Detection Systems), the research focused on true positive, true negative, false negative, false negative, false negative, false positive.

2.6 A Comparative study about IDS

Comparison on performance assessment of IDS utilizing algorithms based on anomaly and signature-based to reduce false alarm rate and identify unknown attacks was made (Hussein, 2016). The findings indicated both anomaly and signature-based IDS techniques approaches have limitations when it comes to detecting known and new attacks (Hussein, 2016). A proposed solution was hybrid system which was a combination anomaly and signature based but the detection rate and false rate for each system differed (Hussein, 2016). In a different approach performance evaluation on hybrid intrusion system models by looking at the IDS ability to classify the network connection with threshold values of accurate results, that depend on classifier with a classification accuracy is considered high. Secondly speed is considered as a crucial factor by noting that the time taken by IDS to train classifies is significance in evaluating IDS performance (Bello, Ravulakolu and Amrita, 2015).

2.7 Effectiveness of IDS

There is a weakness in NIDS which is in processing high speed network traffic with a tendency to drop packets without analyzing them. In research involving the use of a parallelized Snort NIDS with QoS-enabled traffic shaping, the results of the study show that the parallelized and QoS-enabled NIDS outperforms a non-parallelized and non-QoS-enabled NIDS in terms of detection accuracy, false alarm rate, and scalability (Bul'ajoul *et al.*, 2015. The paper provides a comprehensive review of the use of QoS configuration and parallel technology in enhancing NIDS performance. It highlights the potential benefits of these approaches and presents a case study that demonstrates their effectiveness (Bul'ajoul *et al.*, 2015). Due to low detection rate and high false alarm rate associated with anomalies-based IDS, researchers proposed using hybrid data optimization by introducing Do_IDS system that was based on data sampling and feature selection, evaluated by UNSW-NB15 dataset (Ren *et al.*, 2019). However, the proposed IDS did not indicate how effective it can process malicious traffic.

2.8 Strength and Limitations of IDS techniques

2.8.1 Signature based techniques.

By comparing ongoing activities with known attack signatures, intrusions can be identified. This method involves generating signatures from audit records and matching them with current activities to detect intrusions. It is a straightforward approach to implement, offering effectiveness in detecting basic misuse while minimizing resource consumption (Radharaman Institute of Technology and Science, Bhopal, INDIA and Agrawal, 2017)

The limitation of pattern matching is its inability to identify new activities or unknown attacks. It can only recognize known attack patterns and requires constant updating of signature records to detect new attack patterns. The pattern matching technique is primarily suitable for detecting misuse and may not be effective in recognizing novel or emerging threats (Radharaman Institute of Technology and Science, Bhopal, INDIA and Agrawal, 2017).

2.8. 2 Anomaly based techniques.

Anomaly-based intrusion detection systems that employ statistical methods utilize interval timers, event counters, decision stress and resource measures to analyze the order, inter-arrival times, and values of observations. By evaluating the probability of occurrence at specific times, these systems can identify abnormal instances of observed traffic (García-Teodoro et al., 2009). Statistical approaches in A-NIDS have notable benefits. Firstly, they do not depend on prior knowledge of the target system's normal activity; instead, they learn the expected behaviors from observed data. Secondly, statistical methods demonstrate a prominent level of accuracy in detecting malicious activities that unfold over long durations (García-Teodoro et al., 2009). There are notable limitations associated with anomaly-based intrusion detection systems that should be highlighted. Firstly, these systems can be susceptible to manipulation by attackers who can train them to classify network traffic generated during an attack as normal, thus evading detection (García-Teodoro et al., 2009). Secondly, determining the optimal values for different parameters and metrics poses a challenge as it involves striking a balance between false positives and false negatives. Additionally, A-NIDS assumes a statistical distribution for each variable, but nonspecific methods may not accurately model all types of behaviors. Moreover, many of these systems rely on the assumption of a quasi-stationary process, which may not always reflect the dynamic nature of real-world networks (García-Teodoro et al., 2009).

2.8.3 Machine learning technique

Machine learning principles often overlap with the applicability of statistical techniques, as both aim to improve performance based on past results. Machine learning techniques, however, focus on building models that can adapt and change their execution strategy as new information becomes available. While this adaptability is desirable, one major drawback is the resource-intensive nature of machine learning (García-Teodoro et al., 2009). Additionally, machine learning techniques can only label instances as anomalies without providing meaningful anomaly scores, limiting their ability to provide detailed insights.

To enhance IDS effectiveness, a combination of different techniques may be necessary. Integrating pattern matching with anomaly-based approaches and leveraging the strengths of machine learning can lead to more robust and accurate intrusion detection systems. Regular updates, continuous monitoring, and the inclusion of dynamic behavior modelling can also contribute to improving IDS capabilities in detecting both known and unknown threats.

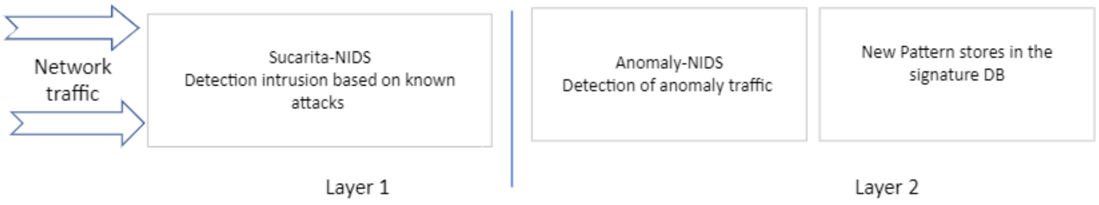
3. Methodology

This segment spotlights the suggested Intrusion Detection system Model designed to identify both unknown and known attacks to effectively defend an organization against malicious attacks.

The suggested approach relies on a combination of signature-based and anomaly-based detection methods. The signature-based approach is utilized to identify intrusions with known signatures, while the anomaly-based technique focuses on detecting deviations from the network's normal operation using machine learning Isolation Forest algorithm.

Figure 1 illustrates the key elements of the proposed system. The initial layer captures network packets and verifies if they match against a database of known intrusion signatures. The subsequent layer, dedicated to anomaly detection, employs machine learning algorithms to model normal network behaviour. The anomaly component of the model serves to identify zero-day attacks, for which signature data is not yet available to the security community. It accomplishes this by recognizing normal behaviour patterns and treating any deviation from the established baseline as an anomaly.

Figure. 1. The method to detect both known and unknown cyber-attacks.


Network traffic
Sucarita-NIDS
Detection intrusion based on known attacks
Anomaly-NIDS
Detection of anomaly traffic
New Pattern stores in the signature DB
Layer 1
Layer 2

3.1.1 Explanation of operation.

At layer 1, network traffic is checked by signature based system, if there is matching characteristics with signature database, alerts are sent or traffic is flagged, if no known packets, packets are moved to layer two where it is compared to the baseline algorithm , if the packets deviate from baseline, alerts are sent and a copy is saved in the signature database to be detected as an intrusion (known signature).

3.2 Use case scenario

This experimental scenario is designed to narrow the focus of this research towards a particular type of threat, rather than considering all the available threats indiscriminately.

Before implementing additional security measures to enhance the overall security of the renowned microfinance computer networks system, the system administrator needs to know how effective Intrusion Detection Systems (IDS) are? The current system includes firewalls at the network boundary, antivirus software on local machines, and an IDS router to analyze incoming and outgoing network traffic, server hosting FTP services and other connected nodes and services that support end users.

4. Experimentation

This chapter puts into action the experimental plan outlined in chapter three, detailing the tools used for each experiment and providing instructions for their installation and configuration.

4. 1 Dataset

4.1.1 CICIDS2017

The CICIDS2017 dataset is a trustworthy and publicly accessible collection of network flows, consisting of benign traffic and seven commonly encountered attack patterns. This dataset is designed to align with real-world conditions and offers an extensive range of network traffic features, as well as various machine learning algorithms analysis files making it closer to realistic situation (Sharafaldin et al., 2018).

CICIDS2017 encompasses all the crucial eleven criteria, comprising prevalent and up-to-date attack types like DoS (Denial of service), DDoS (Distributed Denial of Service), Brute Force, XSS, SQL Injection, Infiltration, Port Scan, and Botnet figure 2 and table 1 (Kurniabudi et al., 2020). It is a fully labelled dataset, ensuring that each data instance is correctly classified, and it offers an extensive collection of over 80 network traffic features carefully extracted and computed for all benign traffic instances (Sharafaldin et al., 2018).

Table 1: Summary of the dataset daily label

Days	Labels
Monday	Benign
Tuesday	SSH, SFTP and Bforce
Wednesday	Slowhttptest,Dos and heartbleed attacks slowloris,GoldenEye and Hulk
Thursday	Cool disl,Sql inject, XSS, Web bforce, and wen and Infiltration attacks
Friday	DDoS LOIT, BotnetARES, portScan (sS, ST, SF, SX, SN, SP, SV, su,So,Sa,sW,SR,sl andB

4.1.2 Description

Regarding the dataset size, the CICIDS2017 dataset comprises 2,830,743 records, with 2,273,097 records corresponding to normal traffic and 557,646 records representing abnormal traffic that were recorded from Monday to Friday during working hours. The dataset was compiled over a span of five days, comprising 80.3% regular network traffic, while the remaining 19.7% consisted of fourteen distinct attacks. Table 2 shows recorded file formats in Pcap and CSV files while figure 2 illustrates the distribution of various attacks within the dataset.

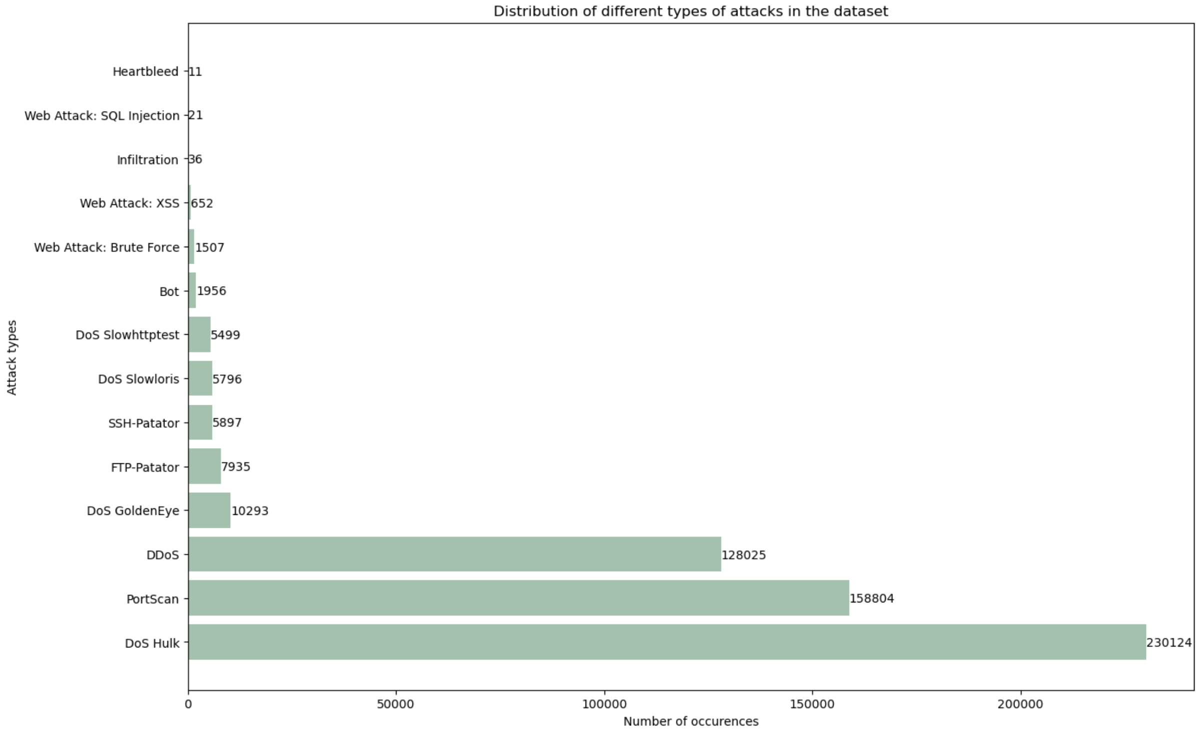
Table 2. CICIDS2017 Pcap and CSV file distribution

Flow_recording (working hours)	Pcap_file size	Duration	CSV_file size	Attack_Name	Flow_count
Monday	10GB	All day	257	Normal traffic	529918
Tuesday	10GB	All day	166MB	SSH-patator, FTP-patator	445909
Wednesday	12gb	All day	272MB	Dos Hulk, Dos Goldeneye,heartbleed, Dos slowhttptest, Dos slowloris	692702
Thursday	7.7GB	Morning	87.7MB	Web attacks-Brute force, XAA,Sql injection	170366
		Afternoon	103 mb	Infiltration	288602
Friday	8.2 gb	Morning	71.8MB	Bot	192033
		Afternoon	97.1MB	portscan	286467
		Afternoon	92.7	DDos	225745

Table 3. CICIDS2017 dataset

Network traffic classes	Frequency
Normal	2,273,097
Abnormal/Attack	557,646
Total instances	2,830,743

Figure 2. Distribution of attack types in CICIDS2017 dataset



4. 2 Anomaly-based approaches

Anomaly detection using the Isolation Forest, during this approach I categorized the sample traffic generated into broader group. Additionally, I assigned labels to facilitate the prediction of the target variable and establish the basis for the input and where -1 represents attack and 1 represents normal.

4.2.1 Isolation Forest Algorithm

Pattern matching with anomaly-based approaches, such as the Isolation Forest algorithm, are techniques used to detect anomalies or outliers in data. The Isolation Forest is an unsupervised machine learning algorithm specifically designed for anomaly detection. It works by isolating anomalies from most normal data points based on their distinctive patterns (Liu et al., 2008). Instead of predicting specific labels, unsupervised learning algorithms aim to identify underlying structures or groups in the data.

The Isolation Forest algorithm, originally introduced by (Liu et al., 2008) in their 2008 paper "Isolation Forest," is a one-class classifier designed for anomaly detection. It focuses on isolating anomalies as the rare instances in the data by constructing isolation trees, where normal data points are easier to isolate and have shorter paths in the trees compared to anomalies.

The algorithm randomly selects subsets of data points and splits them based on feature values to create isolation trees. It then measures the isolation path length from the root node to each data point, allowing it to score and identify anomalies based on shorter path lengths. This approach makes the Isolation Forest efficient and effective for detecting anomalies even in high-dimensional datasets with low memory and CPU usage (Liu et al., 2008).

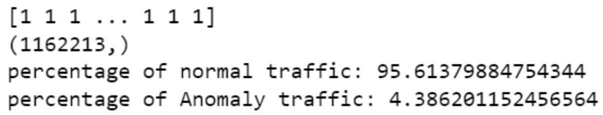
In the Isolation Forest algorithm, you do not need to split the dataset into testing and training parts, as it is an unsupervised algorithm that does not rely on labelled data. Instead, you can use various evaluation metrics to assess its performance in identifying anomalies in the data (Chabchoub et al., 2022).

4.2.2 Creation of the anomaly detection model

The process involved combining the individual CSV data files from CICIDS2017 dataset collected on Thursday to Friday into a single consolidated dataset. This merged dataset was created to develop an Isolation Forest model for anomaly detection in network traffic.

The Isolation Forest model was set at the specified maximum samples of 10,000 value and a fixed random seed of 42 at the initial stage. It was essential to set the random seed to reproduce the same results during multiple runs and compare the model's performance consistently. Secondly the choice of maximum samples can affect the trade-off between computational efficiency and accuracy in anomaly detection, limiting the number of samples in each tree is to reduce the computational complexity and memory requirements of the algorithm. Smaller values can speed up the training process, but it might result in less accurate predictions. As indicated in figure 3, 95.5% of the dataset is normal whereas 4.3% is abnormal traffic at 10000 samples tree.

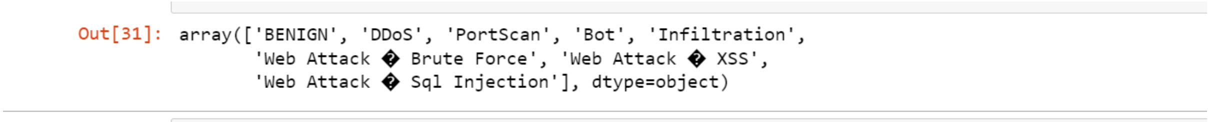
Figure 3. Proportion percentage of normal and anomaly traffic detected by the Isolation Forest



```
[1 1 1 ... 1 1 1]
(1162213,)
percentage of normal traffic: 95.61379884754344
percentage of Anomaly traffic: 4.386201152456564
```

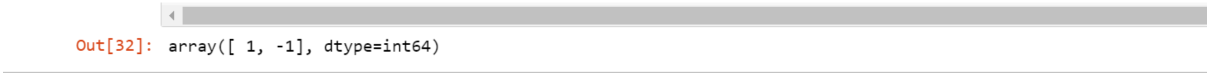
This was followed by creating true data, preprocessing the labels to prepare them for the Isolation Forest model figure 4. This assigns the value -1 to all instances of attack categories, treating them as anomalies, and assigns the value 1 to instances of the normal class, treating them as true data. This ensures that the labels are compatible with the binary classification approach of the Isolation Forest algorithm, where it predicts anomalies with -1 and true data with 1, figure 5.

Figure 4. Preprocessing of labels in the grouped dataset



```
Out[31]: array(['BENIGN', 'DDoS', 'PortScan', 'Bot', 'Infiltration',
                'Web Attack � Brute Force', 'Web Attack � XSS',
                'Web Attack � Sql Injection'], dtype=object)
```

Figure 5. True data



```
Out[32]: array([ 1, -1], dtype=int64)
```

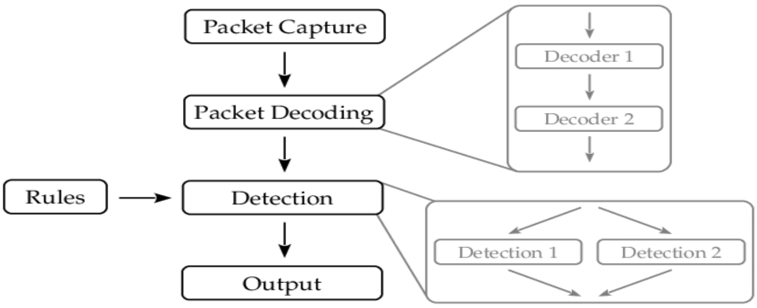
4.3 Signatured based detection

4.3.1 Signature detection with Suricata IDS

Suricata functions as an Intrusion Detection and Prevention System (IDPS) that operates using predefined rules. The accuracy of these rules determines the level of false negatives and false positives in threat detection. Suricata can include actions such as 'alert' and 'log' in IDS mode, and additional actions like 'drop,' 'sdrop,' and 'reject' in IPS mode. Suricata employs a multi-threaded detection approach, enabling efficient utilization of multi-core systems and concurrent network traffic analysis (Fekolkin, 2014). This design allows for superior scalability compared to other open-source IDS. Moreover, on single-core machines, Suricata can run in a single-threaded mode, processing packets one by one.

In Suricata, every packet undergoes a two-step processing decoding and detection. During decoding, the packet is read, and its data is converted into an internal representation. This internal representation is saved for further analysis. The decoding functions are executed sequentially, one at a time, on the packet data. After decoding, the packet is passed to the detection modules for further analysis and identification of potential threats figure 6.

Figure 6. Suricata decoding and detection mechanisms.



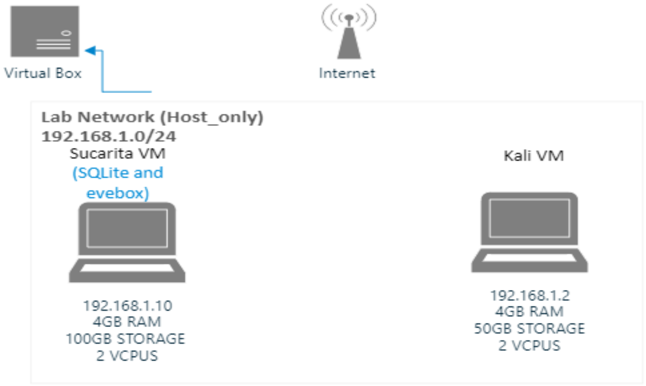
4.3.2 Experimental design for anomaly detection analysis.

Two open-source tools for the network security setup were deployed. Suricata, an IDS/IPS system, served as an intrusion detection system operating an open ruleset, while evebox and SQLite were used to visualize and monitor the Suricata logs such as timestamp, source IP, destination IP and signature.

An Intrusion Detection System (IDS) is responsible for identifying network anomalies and issuing alerts. As malicious network activity generates a high volume of logs every hour, it becomes essential to employ tools like Security Information and Event Management (SIEM). SIEM helps in indexing logs to facilitate event correlation, network analytics, and data visualization.

I utilized the virtual environment for the lab setup as shown in figure 7. Suricata IDS was installed and configured on Suricata VM (Virtual Machine) and evebox and finally Kali VM for exploit generation.

Figure 7. Virtual Lab setup



4.3.3 Exploitation generation

Kali VM represented as a threat actor, attacking Suricata VM, utilizing Pcap files of CICIDS2017 dataset of Tuesday to Friday. Using TCPreplay can be a convenient way to replay Pcap files and test the functionality of IDS systems using real-world network traffic. It allows us to reproduce network conditions and evaluate the effectiveness of security measures.

The TCPreplay tool sent the packets from the Pcap files back into the network. On the destination computer (192.168.1.10) running Suricata IDS, monitor the network traffic as TCPreplay replays the Pcap files. Suricata to analyze incoming traffic for intrusion detection and other security purposes.

The above process did not yield the best results since the lab dint have enough resources mainly the CPUs and memory to run traffic generation and capturing the same time, the alternative option was to run Suricata in offline mode analyzing each Pcap file separately, creating a temporary eve.json and using Evebox at the same time to extract timestamp, source and destination IP, signature . Each time Suricata was run, it first empties the eve.json to delete the old logs before the new alerts are stored. Alerts for each Pcap file were recorded.

5. Evaluation

5.1 Anomaly detection evaluation

I assessed the effectiveness of the isolation forest algorithm using various tree structures by applying the evaluation metrics explained in Section 2.5. I used mayplotlib to generate a confusion matrix that illustrated the visualization and evaluation of isolation forest model performance figure 8, using predicated value and actual values. This aids in threshold selection and fine-tuning the model to achieve optimal detection results. In binary classification, the confusion matrix displays the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) results, as presented in Table 4 and 5 with 5000 tree sample and table 7 at 10000 tree samples. The columns represent the correct classification of the data, while the rows represent the available classifications.

Figure 8: Confusion matrix

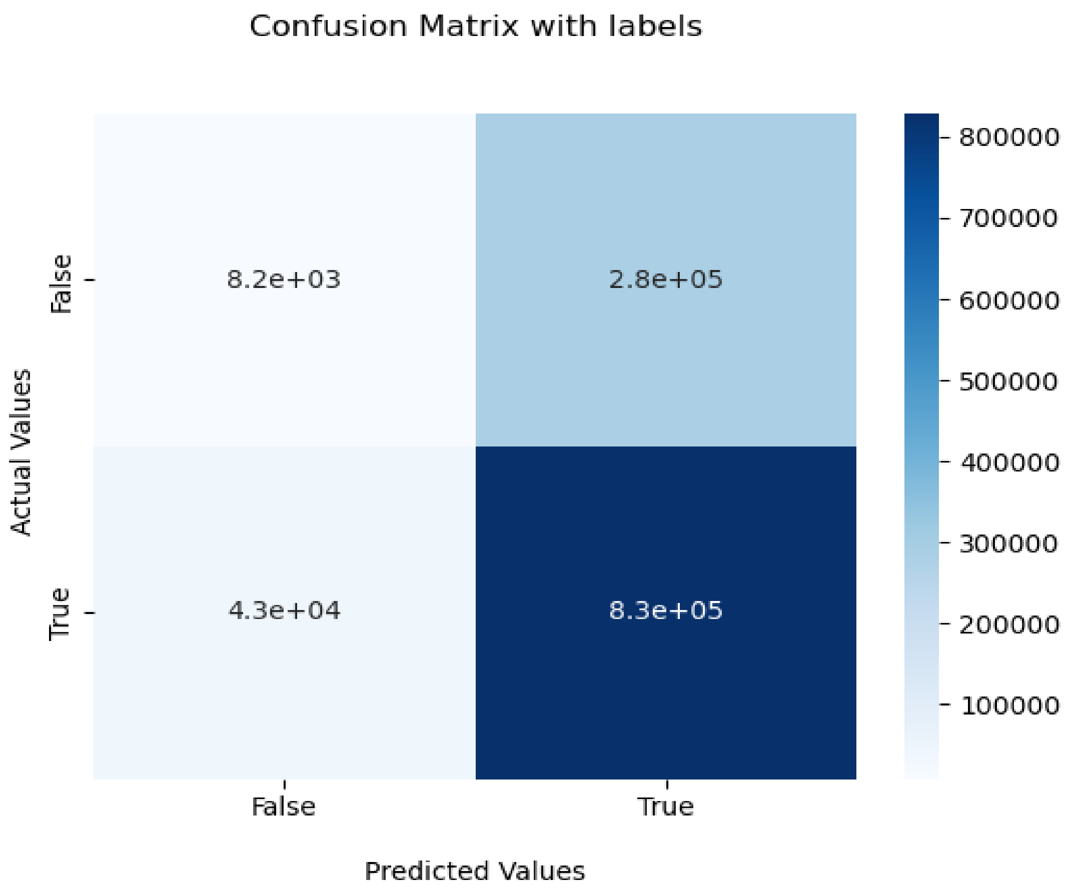


Table 4. Confusion matrix binary classification.

Actual label		
Predicated label	No attack	Attack
No attack	True negative	False negative
Attack	False Positive	True Positive

Table 5. Sampling the model with 5000 trees

Classify	Counts
True Negative (tn)	11053
True Positive (tp)	823327
False negative (fn)	47747
False positive (fp)	280086

recall = tp/(tp+fn)

precision = tp/(tp+fp)

f1 score = 2 * (precision*recall)/(precision+recall)

Table 6: classification summary

Precision	Recall	F1-score
0.74	0.94	0.83

Table 7. Fine-tuning the model with 10000 sample trees

Classify	Counts
True Negative (tn)	8240
True Positive (tp)	828337
False negative (fn)	42737
False positive (fp)	282899

Recall = tp/(tp+fn)

Precision = tp/(tp+fp)

f1 score = 2 * (precision*recall)/(precision+recall)

Table 8. classification summary

Precision	**Recall**	**F1-score**
0.74	0.95	0.85

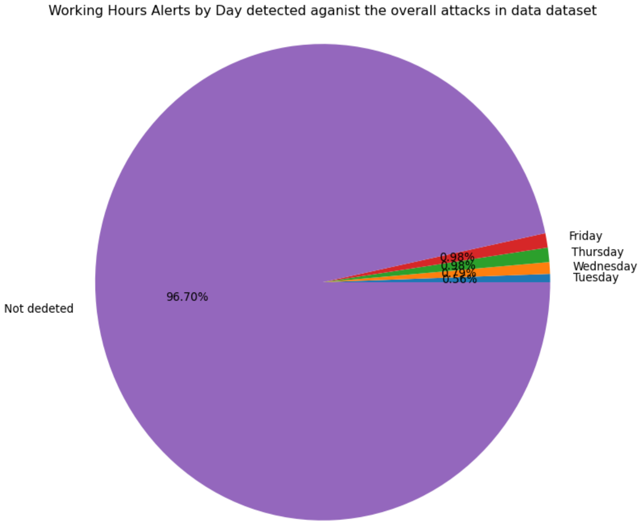
5.2 Signature based.

While conducting the evaluation of signature-based ids, realistic and representative test data from pcap file of CICIDS2017 dataset recorded on Tuesday, Wednesday, Thursday, and Friday that mimic real-world scenarios were used, and following best set up and configuring of Suricata. Each file was analyzed separately using ruleset, et/open with more than 2850 signature count that generated alerts as indicated in table 9.

Table 9. Alerts detected by emerging threat ruleset.

Working hours	Alerts generated	% detected
Tuesday	3108	0.79
Wednesday	4416	0.56
Thursday	5446	0.98
Friday	5444	0.98

Chart 1: **Percentage of generated alerts using emerging threat ruleset.**

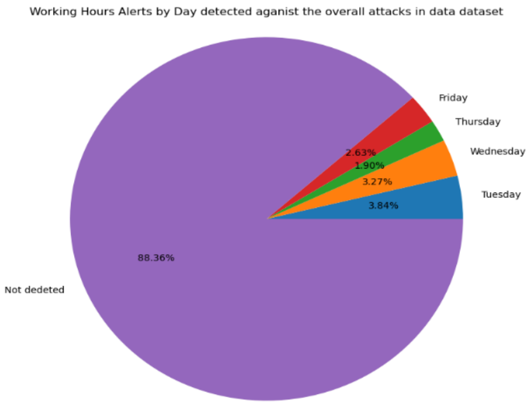


To evaluate the effectiveness of signature-based detection using Suricata, I developed a customized ruleset specifically targeting attacks related to port scans, cross-site scripting (XSS), brute force, SQL injection, bots, DDoS, and DoS. These attack types were the primary labels in the CICIDS2017 dataset. The second run using customized generated alerts is shown in table 10 and their distribution against the total attacks in the dataset in chart 2.

Table 10. Alerts Detected by customized ruleset.

Working hours	Alerts Detected	% detected
Tuesday	21413	3.84
Wednesday	18221	3.27
Thursday	10622	1.9
Friday	14674	2.63

Chart 2. **Percentage of generated alerts using customized ruleset**



6. Discussion

This section presents the study's findings derived from the experiments detailed in the previous section. The evaluation of the isolation forest algorithm revealed its ability to effectively identify anomalies in the dataset when sampled with a bigger tree size. The confusion matrix allowed us to observe true positives, false positives, true negatives, and false negatives, aiding in understanding the model's strengths and weaknesses.

The achieved precision of 0.74 indicted that out of all the instances predicted as anomalies by the Isolation Forest model, 74% were correctly classified as true anomalies (true positives), while the remaining 26% were incorrectly classified as anomalies (false positives) whereas the recall of 0.95 indicates that out of all the actual anomalies present in the dataset, the Isolation Forest model correctly identified 95% of them as anomalies (true positives), and 5% of the actual anomalies were misclassified as normal instances (false negatives). Finally, the F1-score of 0.85 is a balanced measure that combines both precision and recall. It indicated that the model achieved a good trade-off between precision and recall, providing a relatively accurate and balanced detection of anomalies in the dataset.

In summary, the high recall (0.95) suggests that the Isolation Forest model is effective in capturing most of the actual anomalies present in the dataset. The precision of 0.74 indicates that there is a moderate number of false positives, meaning the model may sometimes flag normal instances as anomalies and the F1-score of 0.85 demonstrates the overall effectiveness of the Isolation Forest model in identifying anomalies while maintaining a reasonable balance between precision and recall.

The customized ruleset for Suricata targeted a range of attacks, including port scans, cross-site scripting, brute force, SQL injection, bots, DDoS, and DoS. The evaluation during working hours on different days provided valuable insights into the ruleset's effectiveness. The detection rates varied across different attack types, indicating the ruleset's adaptability to real-world scenarios. The signature-based IDS showed promising results in detecting and alerting people to suspicious activities during the evaluation period. The percentage of detected alerts varied depending on the attack type and its prevalence in the dataset.

Conclusion and future work

The primary objective of this study was to address the research question, "What is the effectiveness of Network Intrusion Detection Systems (NIDS) in improving organizational security and preventing cyber-attacks?" To investigate this question, two experiments were conducted, involving anomaly detection using the Isolation Forest machine learning technique and signature detection using Suricata.

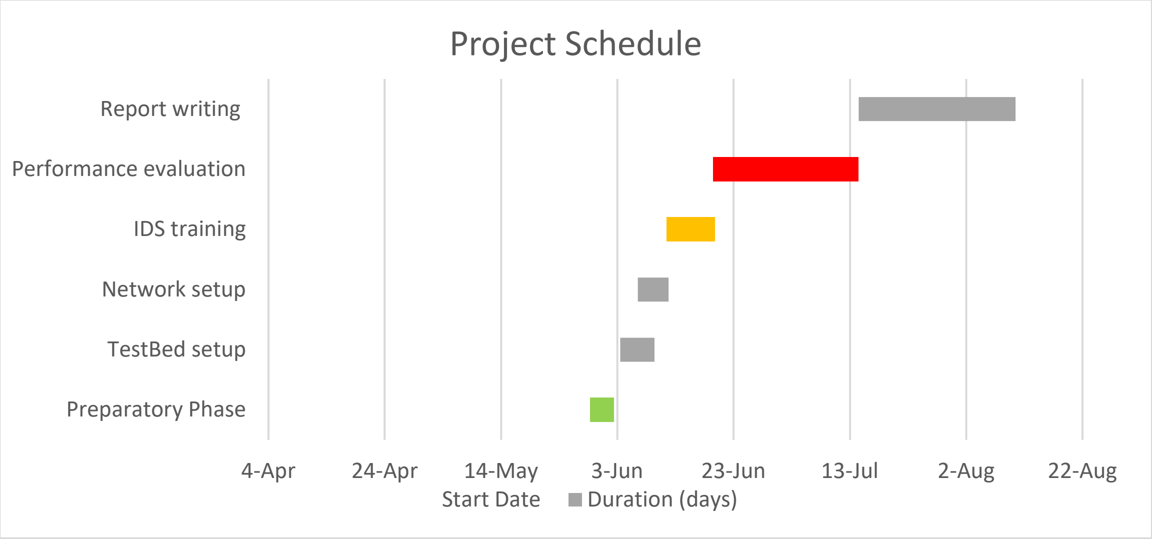
The Isolation Forest algorithm demonstrated promising results in detecting anomalies within the dataset when using a big sample size. By visualizing the model's performance through a confusion matrix, we could fine-tune the model's thresholds and achieve optimal detection results. The precision, recall, and F1-score were calculated for both the "Attack" and "Normal" classes, offering insights into the model's performance. The achieved metrics were competitive and showed the algorithm's potential in identifying unusual network behaviors and potential threats. However, it is essential to recognize the limitations of each approach. The anomaly detection model might face challenges in identifying previously unseen attack patterns, while the signature-based IDS heavily relies on known signatures, making it susceptible to zero-day attacks.

Based on the evaluation results, I recommend adopting a multi-layered approach to intrusion detection. Integrating both anomaly detection and signature-based IDS can provide complementary benefits, enhancing overall network security monitoring capabilities.

In conclusion, the evaluation of the anomaly detection model and the signature-based IDS highlighted their effectiveness in detecting various network attacks. By combining these approaches, organizations can bolster their security posture and better defend against both known and unknown threats. The findings provide valuable insights into these detection mechanisms' capabilities, offering a foundation for further research and improvements in network security monitoring. For future work, incorporating threat intelligence feeds and machine learning techniques to continuously update and refine the ruleset and the anomaly detection model is advised. Additionally, evaluating the effectiveness of the detection mechanisms on larger and more diverse datasets will further validate their applicability in real-world scenarios.

Project schedule

The Gant chart below in figure 1 is generated from section 3.4 that explains the steps and methods that were taken duration IDS evaluation.



Bibliography

Aldwairi, M., Alshboul, M.A., Seyam, A., 2018. Characterizing Realistic Signature-based Intrusion Detection Benchmarks, in: Proceedings of the 6th International Conference on Information Technology: IoT and Smart City. Presented at the ICIT 2018: IoT and Smart City, ACM, Hong Kong Hong Kong, pp. 97–103. doi:10.1145/3301551.3301591

Bello, F.L., Ravulakollu, K., Amrita, 2015. Analysis and evaluation of hybrid intrusion detection system models, in: 2015 International Conference on Computers, Communications, and Systems (ICCCS). Presented at the 2015 International Conference on Computers, Communications, and Systems (ICCCS), IEEE, Kanyakumari, India, pp. 93–97. doi:10.1109/CCOMS.2015.7562879

Bul'ajoul, W., James, A., Pannu, M., 2015a. Improving network intrusion detection system performance through quality of service configuration and parallel technology. Journal of Computer and System Sciences 81, 981–999. doi:10.1016/j.jcss.2014.12.012

Bul'ajoul, W., James, A., Pannu, M., 2015b. Improving network intrusion detection system performance through quality of service configuration and parallel technology. Journal of Computer and System Sciences 81, 981–999. doi:10.1016/j.jcss.2014.12.012

Chabchoub, Y., Togbe, M.U., Boly, A., Chiky, R., 2022. An In-Depth Study and Improvement of Isolation Forest. IEEE Access 10, 10219–10237. doi:10.1109/ACCESS.2022.3144425

Deepa, A.J., Kavitha, V., 2012. A Comprehensive Survey on Approaches to Intrusion Detection System. Procedia Engineering 38, 2063–2069. doi:10.1016/j.proeng.2012.06.248

Denning, D.E., 1987. An Intrusion-Detection Model. IIEEE Trans. Software Eng. SE-13, 222–232. doi:10.1109/TSE.1987.232894

Fang, W., Tan, X., Wilbur, D., 2020. Application of intrusion detection technology in network safety based on machine learning. Safety Science 124, 104604. doi:10.1016/j.ssci.2020.104604

Fekolkin, R., 2014. Intrusion Detection and Prevention Systems: Overview of Snort and Suricata.

García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E., 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security 28, 18–28. doi:10.1016/j.cose.2008.08.003

Ghurab, M., Gaphari, G., Alshami, F., Alshamy, R., Othman, S., 2021. A Detailed Analysis of Benchmark Datasets for Network Intrusion Detection System. AJRCoS 14–33. doi:10.9734/ajrcos/2021/v7i430185

Heberlein, L.T., Dias, G.V., Levitt, K.N., Mukherjee, B., Wood, J., Wolber, D., 1990. A network security monitor, in: Proceedings. 1990 IEEE Computer Society Symposium on Research in Security and Privacy. Presented at the Proceedings. 1990 IEEE Computer Society Symposium on Research in Security and Privacy, IEEE, Oakland, CA, USA, pp. 296–304. doi:10.1109/RISP.1990.63859

Hussein, S.M., 2016. Performance Evaluation of Intrusion Detection System Using Anomaly and Signature Based Algorithms to Reduction False Alarm Rate and Detect Unknown Attacks, in: 2016 International Conference on Computational Science and Computational Intelligence (CSCI). Presented at the 2016 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE, Las Vegas, NV, USA, pp. 1064–1069. doi:10.1109/CSCI.2016.0203

Kurniabudi, Stiawan, D., Darmawijoyo, Bin Idris, M.Y., Bamhdi, A.M., Budiarto, R., 2020. CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection. IEEE Access 8, 132911–132921. doi:10.1109/ACCESS.2020.3009843

León, V., n.d. INsIDES: A new Machine Learning-based Intrusion Detection System.

Liu, F.T., Ting, K.M., Zhou, Z.-H., 2008. Isolation Forest, in: 2008 Eighth IEEE International Conference on Data Mining. Presented at the 2008 Eighth IEEE International Conference on Data Mining (ICDM), IEEE, Pisa, Italy, pp. 413–422. doi:10.1109/ICDM.2008.17

Moustafa, N., Slay, J., 2015. The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems, in: 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS). IEEE, pp. 25–31.

Radharaman Institute of Technology and Science, Bhopal, INDIA, Agrawal, G., 2017. A SURVEY ON ATTACKS AND APPROACHES OF INTRUSION DETECTION SYSTEMS. ijarcs 8, 499–504., doi:10.26483

Ren, J., Guo, J., Qian, W., Yuan, H., Hao, X., Jingjing, H., 2019. Building an Effective Intrusion Detection System by Using Hybrid Data Optimization Based on Machine Learning Algorithms. Security and Communication Networks 2019, 1–11. doi:10.1155/2019/7130868

Samrin, R., Vasumathi, D., 2017. Review on anomaly based network intrusion detection system, in: 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT). Presented at the 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), IEEE, Mysuru, pp. 141–147. doi:10.1109/ICEECCOT.2017.8284655

Sharafaldin, I., Habibi Lashkari, A., Ghorbani, A.A., 2018. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization:, in: Proceedings of the 4th International Conference on Information Systems Security and Privacy. Presented at the 4th International Conference on Information Systems Security and Privacy, SCITEPRESS - Science and Technology Publications, Funchal, Madeira, Portugal, pp. 108–116. doi:10.5220/0006639801080116

Sulaiman, N.S., Nasir, A., Othman, W.R.W., Wahab, S.F.A., Aziz, N.S., Yacob, A. and Samsudin, N. (2021) 'Intrusion detection system techniques: A review', *Journal of Physics: Conference Series*, 1874, pp. 1-10. doi: 10.1088/1742-6596/1874/1/012042.