HARNESSING EVOLVING MACHINE LEARNING TECHNIQUES FOR ENHANCED INTRUSION DETECTION SYSTEM

MSc Research Project
MSc Cyber-security

Viral Sonavadia
Student ID: X23103116

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland

Project Submission Sheet

Student Name:	Viral Sonavadia		
Student ID:	X23103116		
Programme:	MSc in Cybersecurity	**Year:**	2023-2024
Module:	MSc Research Project		
Lecturer:	Mr. Niall Heffernan		
Submission Due Date:	12th August 2024		
Project Title:	**HARNESSING EVOLVING MACHINE LEARNING TECHNIQUES FOR ENHANCED INTRUSION DETECTION SYSTEM**		
Word Count:	6771		

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: Viral Sonavadia

Date: 12th August 2024

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):	

AI Acknowledgement Supplement

[Insert Module Name]
[Insert Title of your assignment]

Your Name/Student Number	Course	Date
X23103116	Msc in Cybersecurity	12th August 2024

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
NA	NA	NA

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

[Insert Tool Name]	
[Insert Description of use]	**NA**
[Insert Sample prompt]	[Insert Sample response]

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

[Place evidence here]

Additional Evidence:

[Place evidence here]

HARNESSING EVOLVING MACHINE LEARNING TECHNIQUES FOR ENHANCED INTRUSION DETECTION SYSTEM

Abstract

The overall design specification section determines the functional portion of the data execution proportion. This determines the functional evaluation of all the constructional approaches. The designing approach determines the involvement of Python coding. The designing approach introduces the functional section of the configuration of IDS. The implementation defines the introduction of the evaluation process by using data reading functionality. The supportable execution also provides the designing of the executional parameters. The evaluation defines the construction of various machine learning models such as KNN, SVM, Random Forest, Decision Tree, and ANN. The evaluation of those models supports the finding of the most suitable model for the detection of intrusion in the network. Future development supports the upgradation of this research process by the implementation of AI techniques and advanced methods.

Chapter 1: Introduction

1.1 Background

"Intrusion Detection System" (IDS) is a network shield technology that observes network traffic for dubious movement and comprehends threats. It dispatches attention to IT and security units when it notices a security hazard. IDSs are essential because they can assist in "Identifying security incidents, Analysing the type and quantity of attacks, identifying problems with device configurations, supporting regulatory compliance, improving network performance, providing insights into network traffic". It is also identified that IDSs can be software applications or devices. They can be established on a client computer as a host-based IDS or reside on the network as a network security device, it is also notable that, Cloud-based IDSs are also available.

IDSs work by catching and investigating network containers. They resemble the gathered data with comprehended "hazard signatures or abnormal" movement patterns. IDSs are listen-only instruments, nevertheless, and they cannot automatically make an effort to contain a witnessed exploit. Some IDSs can be incorporated with "intrusion prevention systems (IPSs)", which can automatically act to contain safety dangers. IDS has been a staple of company security since the 1980s. Their development has been characterized by improvements in technology and modifications in cyber security strategies. Some milestones in the history of IDS are "1986, late 1980s, and early 2000s". In recent years, the challenges of cloud computing and the dominance of IDS have put a new spotlight on it, and many organizations are investing in proactive security measures and other preventative procedures while still realizing the significance of glimpsing aggression that may appear thereafter.

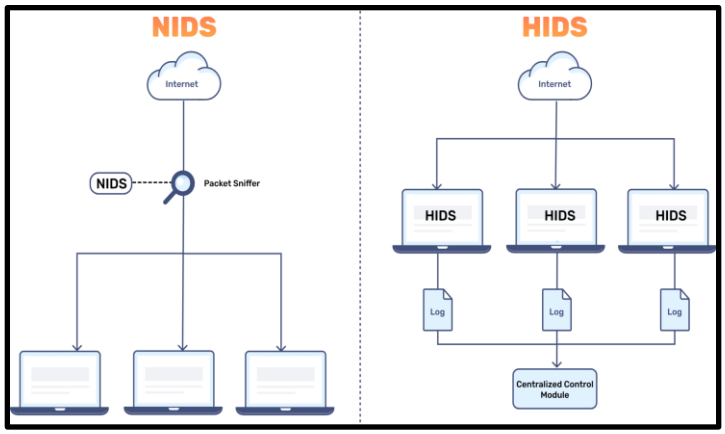


Figure 1.1: Two key types of IDS

(Source: Bunny.net 2020)

The above figure shows that the two leading types of IDS are "network-based IDS (NIDS) and host-based IDS (HIDS)". In terms of NIDS, monitors network traffic for questionable movement in real-time. Apart from these, NIDS examines "incoming and outgoing" gridlock from instruments on the network, looking at packet scopes and metadata to determine hazards. NIDS can be deployed for complete problems on the network. In terms of HIDS, monitors movements on personal appliances or hosts, such as system logs, file innocence, and user behaviour. HIDS operates on a distinct device to detect interior transformations by insiders, such as undesirable rewrites to "log and config" files.

It is notable that, both NIDS and HIDS can catch episodes that target "anomalies, signatures", or both. They can also use signature-based detection, which especially squeezes packets traversing through the network against a database of vulnerabilities of the known method.

1.2 Research Problem

Cyber threats have evolved more cultivated and difficult due to several factors such as new technologies such as AI and block chain can create new exposures that fraudsters can manipulate. Also recognized is that secret work has developed more distinctive insider hazards that have become more painful. Regulatory sophistication is another challenge as data security laws such as GDPR in the "European Union impose strict procedures" on how data must be ministered. Colonial engineering is another challenge because assertive competitors may psychologically affect individuals into perpetrating activities that authorize the aggressor to achieve unauthorized entry to the organization's computer techniques and digital data. Phishing is one of the most typical necessities of cyber threats, phishing techniques users into exposing susceptible data. Another distinction is supply chain risks, organizations' agents, supporters, subcontractors, suppliers, and other third groups with entry to their help, may be exposed to "supply chain attacks". It is notable that, to keep up with the increasing complexity of cyber threats, communities can think of using more refined endpoint explanations of security, "multi-factor authentication (MFA)", and creative resolutions such as AI.

1.3 Research aim & Objective

Aim

The aim of the evaluation define the key determination of the evaluated ML techniques which assist in the intrusion detection process. The implementation of the ML technique defines the construction of the latest ID system which helps to protect the network and detect network-based intrusions.

Objective

- *To determine the real-world drawbacks, and benefits of the application of ML-based IDS in the context of the comparison with the traditional IDS (signature-based).*
- *To evaluate the evaluation of the case studies regarding the ML-IDS installation which provides the guidelines and also the relative errors.*
- *To evaluate data privacy in the case of intrusion detection systems using machine learning, and also the flaws that need to be anticipated.*

1.4 Research Questions

1. *What are the real-world benefits and drawbacks of using ML-based IDS (ML-IDS) in comparison to traditional signature-based IDS?*
2. *Can case studies of ML-IDS installations provide insights into guidelines and possible errors?*
3. *How can we ensure data privacy while effectively utilizing evolving machine learning techniques in intrusion detection systems, and what potential flaws should we anticipate?*

1.5 Research Rational

The harnessing of evolving machine learning techniques for enhanced detection systems has been focused on the integration of advanced and adaptive methods of machine learning which can be helpful for the purpose of intrusion detection systems for the purses of improving the effectiveness. However, the intrusion detection system is designed for the purpose of monitoring the network traffic and the system activities for the purpose of signing off the malicious activities. The traditional introduction destruction system can be helpful for the purpose of challenging the high fall positive rates which can provide slow adaptation to the new threads and scalability issues. The machine learning algorithms can be helpful in analyzing a large volume of data which can be helpful to identify patterns and creative potential threats which can provide a more accurate rule-based system. The evolvement of machine learning technique is based upon using several techniques such as adaptive learning where techniques are allowed to learn from the new data as well as adapt them to immerse the thread whereas the normal detection method is based on identifying the derivation from the normal behaviour which can help the bridge of security.

1.6 Summary

The advanced method is the deep learning method which can learn complex patterns from the road followed by improving the detection capabilities. The research has demonstrated that traditional techniques are signature-based detection normally-based detection and many others. The signature-based detection is the description that uses a predefined pattern from the known thread to detect the intrusion whereas a normally based detection is based upon establishing a baseline of normal behaviour that detects the derivation from the norm. The machine learning techniques are supervised learning, unsupervised learning, and many others techniques. In which the supervised learning uses labelled data to train the models and recognize the classification of attack whereas

unsupervised learning is found to identify the pattern and anomalies in the unlabeled data. The execution defines the key finding of the evaluation approach to replace the traditional method and change the intrusion detection system.

Chapter 2: Related Work

2.1 Introduction

ML plays the greatest role in IDS by analysing data to determine and organize safety threats. ML specifically sustain IDS techniques to notice involved hazards, hostile action, and anomalies within a network. It can even help to reduce false positives and acclimate to increasing threats. This team considers different corresponding work on the IDS method, it has been determined that ML can improve IDS techniques, and ML can examine network traffic, procedure diaries, and other data to notice questionable actions and potential threats in real-time (Saranya *et al*.,2020). ML can help IDS systems achieve high detection rates with low false alarm rates. Some research papers explore the use of ML in IDS for applications such as smart cities, fog computing, big data, 5G networks, and the Internet of Things (IoT). Research has also been conducted on using deep machine learning techniques to develop intrusion detection systems.

2.2 Real-world drawbacks, and benefits of the application of ML-based IDS in the context of the comparison with the traditional signature-based IDS

"Signature-based intrusion detection system" (SIDS) observes network traffic for designs that correspond to learned attack autographs. It's also understood as knowledge-based detection or mishandling detection. As per the view of Einy, Oz, and Navaei, (2021), SIDS are good at recognizing and intercepting reported attacks. But they have boundaries. Some pros and cons, in terms of advantages, such that it is correct, therefore it delivers low false positive accelerations. It is also comfortable to operate and, therefore simple performance and composition, needing little training. It also has fast processing, it can fast recognize and block explicit episodes. Another benefit is frequent updates, signature databases are regularly revised by agents or security professionals. Useful at catching well-known episodes such as "worms, denial-of-service (DoS), and malware".

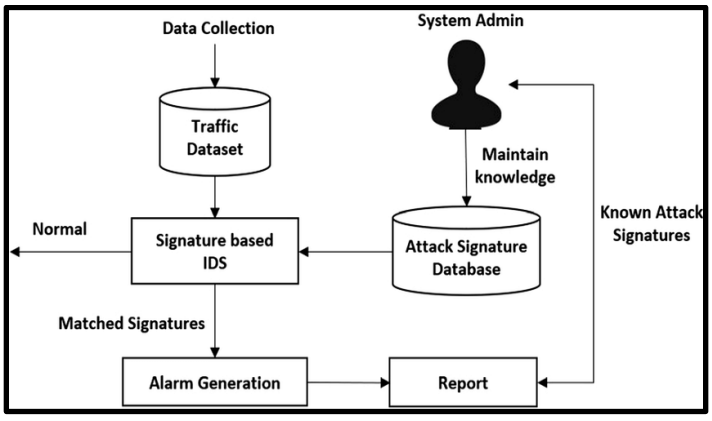


Figure 2.1: "Signature-based intrusion detection system"

(Source: Bangui *et al*., 2022)

According to RiskXchange (2023), "Signature-based intrusion detection systems" have several disadvantages such as limited detection, False positives, Resource-intensive, and Vulnerability. In

terms of limited detection, it can't catch new or unexplored episodes, including "zero-day and polymorphic" attacks. In terms of false positives, it can misinterpret legitimate gridlock for attacks, generating chaos and workload for safety teams. Resource-intensive is another penalty it especially examines every data container against different signatures and can even slow down the implementation of the network. As per the view of Kwon *et al*.,(2022), "Signature-based detection" is a standard foundational strategy for IDSs, but it's not the greatest one. Safety specialists often incorporate signature-based detection with different tools to comprehend network conduct and notice context.

"Signature-based IDS and anomaly-based IDS" are two different kinds of intrusion IDS that differ in how they determine hazards. According to Saputra *et al*.,(2022), Depends on identifying characteristic practices in network gridlock, such as byte series or understood negative education, to determine known hazards such as "malware and phishing". It's straightforward to carry and execute but can ignore new dangers. On the other hand, Anomaly-base IDS, examines network traffic or design training to select a baseline of expected conduct and determine variations. It can notice new episodes but may cause more false positives, including recognizing beneficial but irregular conduct as anomalous.

2.3 The evaluation of the case studies regarding the ML-IDS installation which provides the guidelines and also the relative errors

The preferred method of IDS monitoring depends on the different types of threats specifically detected such as anomaly-based, this method is considered more useful than the signature-based methods due to it can properly detect new threats by analyzing network data and traffic for suspicious patterns (N-able., 2024). It is determined that Anomaly-based IDS can use ML to establish baselines for regular host activity and alert administrators when there are significant deviations. This technique is particularly beneficial at uncovering zero-day and also unknown attacks, but it mostly delivers a bunch of false positives. According to N-able., 2024, the Signature-based method is most reasonable for determining known hazards by examining specific designs and successions in inbound network traffic that correspond to learned attack signatures. These signatures can possess "indicators of compromise (IOCs)" such as file "hashes, malicious domains, or unusual email subject lines". Apart from these, signature-based IDS can't catch anonymous attacks that don't contain available designs.

As per the view of Ahmad *et al*.,(2021), the best algorithm for IDS depends on the kind of data processed and the desired performance metrics. In terms of pattern-matching algorithms, these algorithms are the essence of IDPSs and exploit signatures of available episodes to notice and contain negative movement. The selection of a pattern-matching algorithm is essential to the implementation of the IDS, and implementation can be discussed in the duration of run time, network length, and numeral of conventions. Some pattern-matching algorithms include "Brute-force, Rabin-Karp, Boyer-Moore, and Knuth-Morris-Pratt". As per the view of Saranya *et al*., (2020) in terms of the Naive Bayes algorithm, this algorithm can be utilized as a classifier and to process intrusion data. One strategy uses quality embedding to market with identical features of "normal and abnormal data", and to transform "low-quality data into high-quality" data that's more comfortable to organize (Jadhav, and Pellakuri, 2021). Another strategy employs PCA-based Naive Bayes, which can deliver reasonable effects consistent when information collections have misplaced importance. Apart from these, precision declines and the procedure delays down as the information length advances.

2.4 Data privacy in the case of intrusion detection systems using machine learning, and also the flaws that need to be anticipated

Although IDS provide several advantages still it has several limitations such as false positives, also anointed false alarms, these can make IDS helpless to determine possible dangers that aren't true threats. It is important that to bypass this, communities can configure their IDS to comprehend what average looks such as, and what should be deemed a negative movement. In terms of false negatives, these can be a greater consideration, as the IDS resolution misperceives an existing protection danger for honest traffic (Ahmad *et al*.,2021). This can permit an assailant to pass into the institution's network without "IT and security" groups being conscious. Different operational procedures are another IDS limitation, it has been determined that extra handling systems have additional performance instruments, which can make it challenging to determine a suitable design library (Mebawondu *et al*.,2020). IDS also require full-time monitoring; IDS may demand full-time monitoring by a professional systems manager who can answer to regular false positives and bring movement on any hazard.

IDS and IPS both enhance network visibility, but only IPSes can bring action on hazards. Still, both have limitations. IDS can flag everyday training as questionable, or miss real dangers due to further attack imprints. Host-based IDS only observes the host it is established on, determining its capacity to detect network-wide hazards (Kumar *et al*.,2022). In terms of IPS precisely detecting and stopping network attacks, it can include some restrictions such as IPSs directing resources to analyze and block traffic, which specifically latency issues depending on the size of the network and also the importance of traffic. It is notable that if there are numerous IPSes on a network, data will have to pass through each to contact the end user, driving the implementation of the network.

2.5 Theoretical Underpinning

Information Theory

Information theory, also comprehended as the mathematical approach to contact, is a branch of the mathematical possibility approach that investigates how communication is transmitted, processed, and calculated. It employs statistical and mathematical regulations to explain the concepts, parameters, and regulations that control the communication of transmissions through transmission techniques. The communication approach also estimates the efficiency of transmission revolutions between "humans and machines", and the processing ability of transmission techniques to communicate notification (Shwartz Ziv, and LeCun, 2024). Apart from these, Information theory is used in many scientific and engineering fields, such as "Computer science, Physics, Quantum computing, Communication engineering, Molecular biology, and Social networks". In computer science, the information approach can help with Drawing assumptions, Creating industry findings, and Quantifying efficiency.

Anomaly Detection Theory

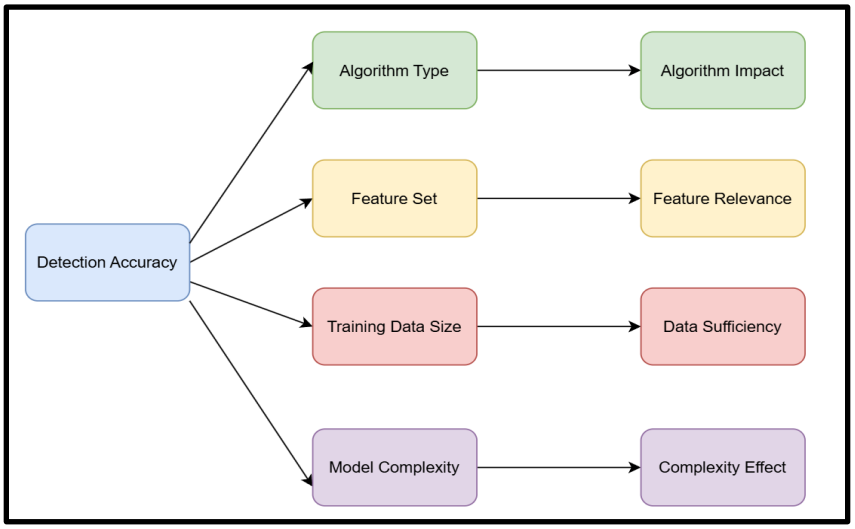
Anomaly detection theory plays a vital role in harnessing evolving ML techniques for enhanced IDS. As per the view of Rosa *et al*.,(2021), Anomaly detection is a method to catch negative movement or procedure infringements in a network or design operating an IDS. IDS utilises "statistics and thresholds" to examine network data for practices and symptoms of strange manners and analogises it to predefined practices and practices to determine possible aggression. It is determined that, if the IDS witnesses something that corresponds to a direction or pattern, it transmits an attentiveness to the procedure manager. It is determined that Anomaly-based IDS can be employed at both the network and announcer levels. Host-based anomaly-based IDS are one of the last coatings of protection and can deliver powdery security at the application level. Apart from

these, anomaly-based IDS can include an increased false-positive momentum and may be deceived by a well-delivered attack.

Machine Learning Theory

ML theory enhances IDS by allowing them to understand data and notice practices that indicate negative movement. This permits them to go further with straightforward rule-based detection and identify both available and anonymous hazards. It is noteworthy that, ML algorithms can analyze tremendous datasets of network traffic and design logs to choose patterns that deviate from normal behaviour, signaling potential attacks (Wang *et al*.,2020). This includes anomaly detection, category models, and deep learning methods. Apart from these, ML abse IDS can adapt to new threats and develop practices of attack. As the system overlooks new data, it regularly updates it is the performance of "normal" and "anomalous" behaviour, and it can improve the detection accuracy over time.

2.6 Conceptual Framework



2.7 Summary

After completing various related work, it has been concluded that ML techniques are employed in IDS to identify new types of episodes by examining traffic of the network. IDSs are classifiers that expressly suggest the class of input records associated with distinct classes of attacks. It is also identified that some ML techniques are employed in IDSs.

Chapter 3: Research Methodology

3.1 Introduction

Research methodology is a systematic plan or approach that is mostly utilised in research to acquire knowledge and understand the proper research problem. It is a vital part of this research because it explains how the research paper. After all, it specifically explains how this research was conducted, which mainly helps readers evaluate this approach's reliability and accuracy. This research aims to develop a good methodological approach that allows readers to authorise the findings. Some processes that mainly help with this research paper's pre-established methodology provide this framework for the paper, which specifically allows investigators to create a hypothesis and describe the method of this study.

3.2 Research Philosophy

Research philosophy is a foundational idea that can affect multiple elements of research, including method, approach, and experience of outcomes. It can also permit researchers to present their choices, and present thoughts clearly, and encourage critical thinking. This research primarily tracks the Interpretivism research philosophy to Harness evolving ML techniques for enhanced IDS (Alharahsheh, and Pius, 2020). interpretivism is a qualitative research philosophy that can assist in gaining a more in-depth knowledge of people's energies and understandings. It can be helpful in domains such as "social work, management, and political culture". Interpretivism can support to analysis of complicated social dynamics and institutional perfections of democratic principles.

3.3 Research Approach

Research approaches, also comprehended as research methodology, are necessary because they permit investigators to determine the best approaches and methods for their analysis purposes and specialisation. They show investigators how to gather, investigate, and interpret data, and are necessary for designing new understanding. This research paper focused on the inductive research approach, which vitally benefits defining the key determination of the evaluated ML techniques which assist in the IDP, the implementation of the ML technique defines the construction of the latest ID system which assists in protecting the network and detection (Sibeoni *et al*.,2020). The leading purpose of using this approach is inductive analysis is a universal approach that permits investigators to create views or inferences established on detailed statements or data. It's usually employed in social science analysis to investigate complicated sensations, such as perspectives, ideas, and manners.

3.4 Research Method

This study uses several methods such as "K-Nearest Neighbor (KNN)", "decision tree (DT), Random Forest (RF)", and "Artificial neural networks (ANN)". These multiple models can help to improve detection accuracy and minimize computational complexity.

3.5 Data analysis

This study uses an ML data analysis approach which especially helps to make a better decision, increases productivity, and also creates the experience for the customer experience. It is idneified that ML can increase data analysis it helps to identify patterns. This study especially adopts the ML approach due to ML can enable investigators patterns and generate hypotheses through the exploration of data, visualisation, and mining. This data analysis process specifically allows to identify the "strengths and weaknesses" of ML examples by analyzing metrics such as "accuracy, precision, and recall" (Vellido, 2020). Apart from these, the ML approach especially finds discreet correlations between dishonest movements and behavioural practices. ML algorithms can follow inconsistencies in datasets to provide safety for the customer and are often employed in charge gateways to witness dishonest movement.

3.6 Data collection

Data collection is an essential element of research assignments and data analytics applications in all occupations, including "social sciences, humanities, business, and physical sciences". This study especially aims to evaluation determine the key finding of the estimated ML techniques which help in the ISP. The implementation of the ML technique represents the structure of the latest ID design which allows to rescue of the network and detect network-based intrusions. Therefore it is quite complex to work with primary data (Medeiros *et al*., 2021). For this significant reason, this study selected a secondary process to evaluate this study. Secondary data permits to

fulfilment of a longitudinal analysis which indicates the investigations are completed travelling over a considerable period.

3.7 Summary

After completing this section it has been concluded that combining ML with data analysis significantly helps to make effective decisions, improve productivity, and also help to develop customer experience. Apart from these, it is beneficial to recognize patterns and generate hypotheses through the investigation of data, visualization of data, and data mining. It also delp to address the “strengths and weaknesses” of ML representatives by analyzing metrics such as “accuracy, precision, and recall”.

Chapter 4: Design Specification

4.1 Designing Concept/Parameters

The design specification for this evaluation introduces the implementation of the technique which assists in identifying the intrusion in the network. The detection approach defines the designing of some specific models using ML techniques. Various ML approaches are implemented to determine the functional section of the overall approaches. The designing specification determines the involvement of the Python environment setting approach. The Python environment defines the implementation of necessary modules/libraries that are implemented for the construction of the testing bench (Karopoulos *et al.,* 2022). The designing approach also introduces the functional section of Python. The Python specification defines the introduction of the necessary data that are usable for the model configuration approaches.

4.2 Summary

The overall design approach determines the configuration of the model construction approach. The model evaluation defines the involvement of the most suitable model for the detection of the intrusion. The approach determines the involvement of the functional key elements that are applicable to the execution, and detection of network threats (Jimmy, 2024). The major finding focuses on the enhancing approach for the detection of the introduction by using the models. The models are applicable to the construction of the IDS.

Chapter 5: Implementation

5.1 Implementation approach

The libraries are applicable for the configuration of the testing platform for the IDS. This provides the introduction of the modules/libraries for the reading, and plotting of the data. The constructional part provides the information about the encoding library that is applicable to the encoding approach.

The training data is applicable for the training of the prototype. This reading approach is applicable to read the data portion. The functional proportional execution approach is implemented to read those data parts.

The test data read portion is implemented to read the major section of the data. This introduces the data read section which is applicable for the testing part of the prototype. The constructional approach introduces this first part of the section. The training data information provides the details of the data that are applicable for further investigation. The checking part provides the details of each data column which defines the count data values, and the type of the data.

This approach is applicable for the checking of the data elements of the major data. The type of the testing data and the count values of the testing data are executed and evaluated in this evaluation approach.
The checking approach is implemented to check the null/blank data portion of the training data. This checking is important to understand and find out the null/blank data values. As per the execution, there are no blank/null values present in the training data section.
The testing data-checking approach is applicable for checking the blank/null section in the data. This checking assists in finding the blank section or null section in the testing data (Rai, and Sahu, 2020). As per the checking, there are no null sections or blank sections present in the testing data.

Chapter 6: Evaluation

The protocol type-wise duration evaluation for IDS defines the sum of the IDS duration. This determines the types of the protocol and their durations.

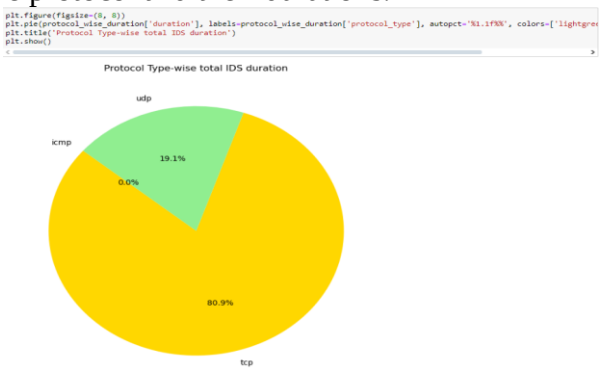


Figure 6.1: Protocol type-wise IDS duration plot

(Source: Own-Evaluated)

IDS duration execution with respect to various protocol types is executed by this plotting. This plot defines the percentage of the IDS duration such as TCP having 80.9%, UDP having 19.1%, and ICMP having 0%.
The flag type-wise duration evaluation determines the value of the IDS with respect to the flag section. This highlights the highest and lowest IDS duration values for a particular flag.

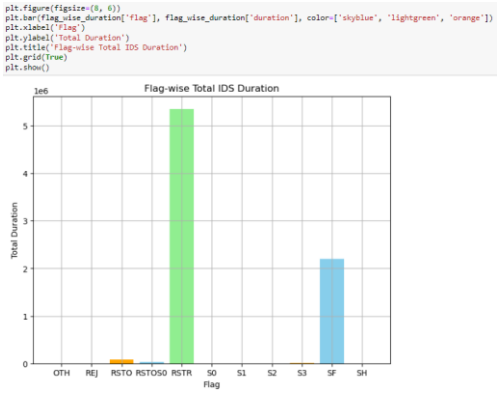


Figure 6.2: Flag type-wise IDS duration plot

(Source: Own-Evaluated)

As per the plot, the highest IDS duration is determined in the case of RSTR whose value is 5357139. The supportable details are evaluated by this plotting approach.
The protocol type-wise execution of the sum of the source bytes is evaluated at this point of execution. This provides the tabular formation for the type of the protocols, and the source byte values.

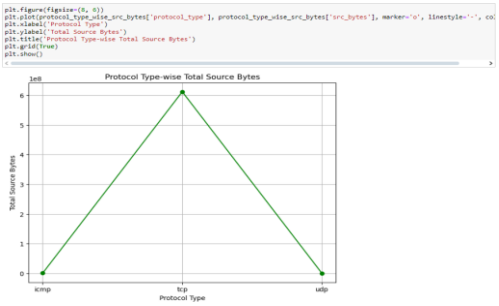


Figure 6.3: Protocol type-wise source bytes plot

(Source: Own-Evaluated)

The line plotting approach is implemented to determine the total source byte values with respect to the protocol types. As per the plot, TCP has the maximum source byte values.
The protocol type-wise execution of the destination bytes defines the total destination bytes for various protocols. This introduces the tabular formation where protocol types and the destination bytes are executed.

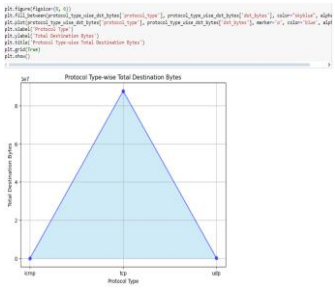


Figure 6.4: Protocol type-wise destination bytes plot

(Source: Own-Evaluated)

The plotting of area evaluates the total destination bytes value for various prototypes. This executes the maximum destination bytes value which is determined in the case of TCP.

```
total_class_counts = data_ids_train['class'].value_counts().reset_index()
total_class_counts
```

	class	count
0	normal	13449
1	anomaly	11743

Figure 6.5: Total IDS class count

(Source: Own-Evaluated)

The total IDS class count approach is implemented to count the classes of IDS. In this case, there are a total of two classes, one is normal, and the other one is anomaly. The normal class defines there is no abnormality present in the network. Whereas the anomaly defines that there are abnormal cases in the network. In this case, the normal count value is 13449, and the anomaly case is 11743.

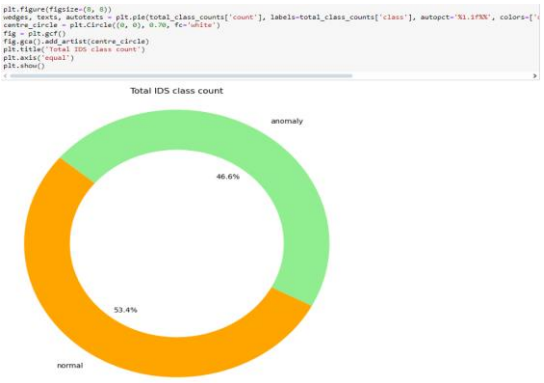


Figure 6.6: Total IDS class plot

(Source: Own-Evaluated)

The plot defines the percentage of the IDS classes for the training data. As per the execution value, there are 53.4% normal, and 46.6% anomaly cases are found.

The data preparation portion determines the setting or preparation of the data for the final execution. This focuses on the evaluation of the detection approach. The encoding approach is implemented to convert the string data into the necessary integer format (Khalil *et al.*, 2021). In this case, the encoding is implemented on the three columns such as protocol_type, flag, and service. After the execution, all the data columns are converted into numeric ones.

The mapping approach is implemented to map or convert the normal, and anomaly data. In this case, normal is defined as 0, and anomaly is defined as 1.

The unused column removal approach is implemented to remove those columns which has no effective impact on the execution. In this case, 'num_outbound_cmds', and 'is_host_login' columns have no effectiveness in the main data execution approach. So, those two columns are removed from the main data section.

```
data_ids_train.describe()
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst
count	25192.000000	25192.000000	25192.000000	25192.000000	2.519200e+04	2.519200e+04	25192.000000	25192.000000	25192.00000	25192.000000	...	
mean	305.054104	1.053827	29.039139	6.982455	2.433063e+04	3.491847e+03	0.000079	0.023738	0.00004	0.198039	...	
std	2686.555640	0.426998	15.555601	2.679322	2.410805e+06	8.883072e+04	0.008910	0.260221	0.00630	2.154202	...	
min	0.000000	0.000000	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.00000	0.000000	...	
25%	0.000000	1.000000	19.000000	5.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.00000	0.000000	...	
50%	0.000000	1.000000	22.000000	9.000000	4.400000e+01	0.000000e+00	0.000000	0.000000	0.00000	0.000000	...	
75%	0.000000	1.000000	46.000000	9.000000	2.790000e+02	5.302500e+02	0.000000	0.000000	0.00000	0.000000	...	
max	42862.000000	2.000000	65.000000	10.000000	3.817091e+08	5.151385e+06	1.000000	3.000000	1.00000	77.000000	...	

8 rows × 40 columns

Figure 6.7: Data describe

(Source: Own-Evaluated)

The 'describe' approach is implemented to describe the data value of the main data section. This determines various calculation values which are necessary for the execution.

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import metrics
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import recall_score, confusion_matrix, precision_score, f1_score, accuracy_score, classification_report
```

```
x = data_ids_train.drop(columns = ['class'])
y = data_ids_train['class'].values
```

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.45, random_state = 40, stratify=y)
```

Figure 6.8: Data preprocessing

(Source: Own-Evaluated)

The data preprocessing approach is implemented to setting of the model libraries initialization, and also the splitting modules. The setting of the categorical data into the 'x' variable and the setting of the target data into the 'y' variable is also evaluated in this point. The splitting approach is implemented to configure the testing, and training data approaches.

```
model_KNN = KNeighborsClassifier(n_neighbors=2)
model_KNN.fit(x_train, y_train)
```

```
KNeighborsClassifier(n_neighbors=2)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
knn_y_pred = model_KNN.predict(x_test)
```

```
knn_accuracy = accuracy_score(y_test, knn_y_pred)
print(f"Accuracy KNN: {knn_accuracy * 100:.2f}%")
```

```
Accuracy KNN: 98.85%
```

```
knn_report_classification = classification_report(y_test, knn_y_pred)
print(knn_report_classification)
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      6052
           1       0.99      0.98      0.99      5285

    accuracy                           0.99     11337
   macro avg       0.99      0.99      0.99     11337
weighted avg       0.99      0.99      0.99     11337
```

Figure 6.9: Model KNN Construction

(Source: Own-Evaluated)

The KNN construction approach determines one of the ML model construction approaches. This defines some necessary steps that define the initialization of the model, and the setting of the model using the training data (Ahmad *et al.*, 2020). The predictive value of the model is applicable to determine the accuracy, and report of classification of KNN. As per the execution, the accuracy portion of the KNN is 98.85%.

```
dec_model = DecisionTreeClassifier(criterion='gini', max_depth=None)
dec_model.fit(x_train, y_train)
```

```
DecisionTreeClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
dec_y_pred = dec_model.predict(x_test)
```

```
dec_accuracy = accuracy_score(y_test, dec_y_pred)
print(f"Accuracy Decision Tree: {dec_accuracy * 100:.2f}%")
```

```
Accuracy Decision Tree: 99.51%
```

```
dec_report_classification = classification_report(y_test, dec_y_pred)
print(dec_report_classification)
```

```
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      6052
           1       1.00      0.99      0.99      5285

    accuracy                           1.00     11337
   macro avg       1.00      1.00      1.00     11337
weighted avg       1.00      1.00      1.00     11337
```

Figure 6.10: Model Decision Tree Construction

(Source: Own-Evaluated)

The decision tree is also an ML model which is constructed using the prototype initialization approach. This defines the fitting of the prototype using training data which is applicable for the prediction using the testing data. The accuracy value portion of the decision tree is 99.51%.

```
rf_model = RandomForestClassifier()
rf_model.fit(x_train, y_train)
```

```
RandomForestClassifier()
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

```
rf_y_pred = rf_model.predict(x_test)
```

```
rf_accuracy = accuracy_score(y_test, rf_y_pred)
print(f"Accuracy Random Forest: {rf_accuracy * 100:.2f}%")
```

```
Accuracy Random Forest: 99.68%
```

```
rf_report_classification = classification_report(y_test, rf_y_pred)
print(rf_report_classification)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      6052
           1       1.00      1.00      1.00      5285

    accuracy                           1.00     11337
   macro avg       1.00      1.00      1.00     11337
weighted avg       1.00      1.00      1.00     11337
```

Figure 6.11: Model Random Forest Construction

(Source: Own-Evaluated)

Another model is Random Forest which is also applicable for the prediction using the testing data. In this case, the accuracy value portion for the RF is 99.68%.

```
svm_cls = SVC()
svm_cls.fit(x_train, y_train)
```

```
SVC()
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

```
svm_y_pred = svm_cls.predict(x_test)
```

```
svm_accuracy = accuracy_score(y_test, svm_y_pred)
print(f"Accuracy SVM: {svm_accuracy * 100:.2f}%")
```

```
Accuracy SVM: 53.43%
```

```
svm_report_classification = classification_report(y_test, svm_y_pred)
print(svm_report_classification)
```

```
              precision    recall  f1-score   support

           0       0.53      1.00      0.70      6052
           1       0.78      0.00      0.00      5285

    accuracy                           0.53     11337
   macro avg       0.66      0.50      0.35     11337
weighted avg       0.65      0.53      0.37     11337
```

Figure 6.12: Model SVM Construction

(Source: Own-Evaluated)

The SVM is one of the prototypes that is supportable by the approach of the execution. This also defines the initialization of the prototype by using the fitting of the prototype using training data. The testing data is applicable for the predictive determination. For this case, the accuracy value is executed as 53.43%.

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
ann_model = Sequential()
ann_model.add(Dense(units=64, activation='relu', input_dim=x_train.shape[1]))
ann_model.add(Dense(units=32, activation='relu'))
ann_model.add(Dense(units=16, activation='relu'))
ann_model.add(Dense(units=1, activation='sigmoid'))
```

```
ann_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
ann_model_history = ann_model.fit(x_train, y_train, epochs=5, batch_size=32, validation_split=0.2, verbose=1)
```

```
Epoch 1/5
347/347 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 0.8071 - loss: 211.4858 - val_accuracy: 0.9441 - val_loss: 2972.8616
Epoch 2/5
347/347 ━━━━━━━━━━━━━━━━━━━━ 0s 732us/step - accuracy: 0.8751 - loss: 144.3681 - val_accuracy: 0.9231 - val_loss: 67.7975
Epoch 3/5
347/347 ━━━━━━━━━━━━━━━━━━━━ 0s 714us/step - accuracy: 0.8623 - loss: 55.1098 - val_accuracy: 0.9527 - val_loss: 1811.7222
Epoch 4/5
347/347 ━━━━━━━━━━━━━━━━━━━━ 0s 717us/step - accuracy: 0.9517 - loss: 27.6138 - val_accuracy: 0.9563 - val_loss: 702.8862
Epoch 5/5
347/347 ━━━━━━━━━━━━━━━━━━━━ 0s 720us/step - accuracy: 0.9449 - loss: 66.5174 - val_accuracy: 0.9589 - val_loss: 11.4735
```

```
data_loss, data_accuracy = ann_model.evaluate(x_test, y_test, verbose=0)
print(f"Accuracy ANN: {data_accuracy * 100:.2f}%")
```

```
Accuracy ANN: 95.76%
```

```
ann_y_pred = ann_model.predict(x_test)
ann_y_pred_classes = (ann_y_pred > 0.5).astype(int)
```

```
355/355 ━━━━━━━━━━━━━━━━━━━━ 0s 537us/step
```

```
ann_report_classification = classification_report(y_test, ann_y_pred_classes)
print(ann_report_classification)
```

```
              precision    recall  f1-score   support

           0       0.96      0.96      0.96      6052
           1       0.95      0.95      0.95      5285

    accuracy                           0.96     11337
   macro avg       0.96      0.96      0.96     11337
weighted avg       0.96      0.96      0.96     11337
```

Figure 6.13: Model ANN Construction

(Source: Own-Evaluated)

The ANN construction approach defines the configuration of the Deep Learning prototype. This executes the prototype using an appropriate epoch value (Kasula, 2023). In this case, the epoch value is 5, and the batch size is 32. As per the executable accuracy, the value for ANN is 95.76%.

6.1 Experiment / Test Case 1

The first test data approach is implemented to set the data parameters for the testing. The test data is applicable for the execution.

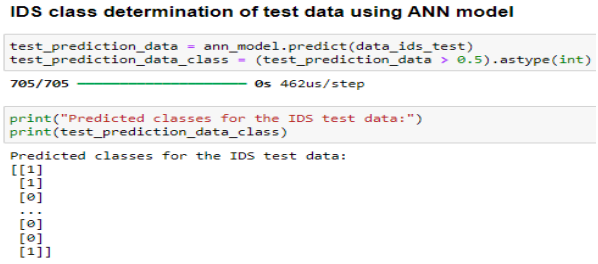


Figure 6.14: IDS class execution using ANN

(Source: Own-Evaluated)

In the first case, the ANN prototype is implemented to detect the IDS class using the testing data.

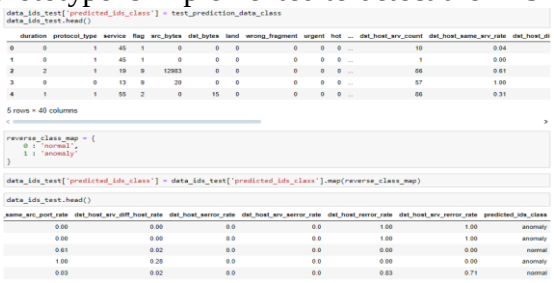


Figure 6.15: Executed data outcome for test 1

(Source: Own-Evaluated)

The overall testing table is constructed using the test data execution approach. This determines the construction of the new tabular section with the predicted IDS class column. This defines the evaluation of normal and anomaly detection for each row.

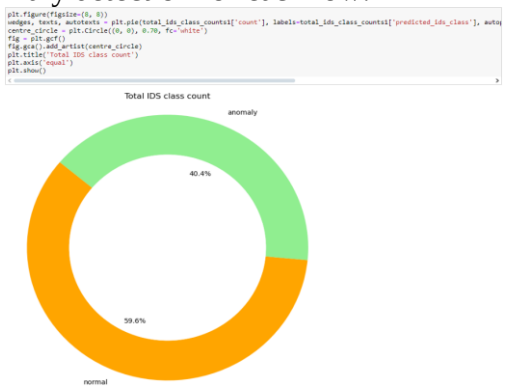


Figure 6.17: Executed data outcome plot for test 1

(Source: Own-Evaluated)

As per the execution of the first test data evaluation, the anomaly percentage is 40.4%, and the normal data value percentage is 59.6%.

6.2 Experiment / Test Case 2

The testing approach for test case 2 executes the test data and the processing of the test data. In this execution, the SVM model is applicable for the detection of the abnormality in the network. The mapping approach is implemented to determine the normal, and abnormal cases in the test data.

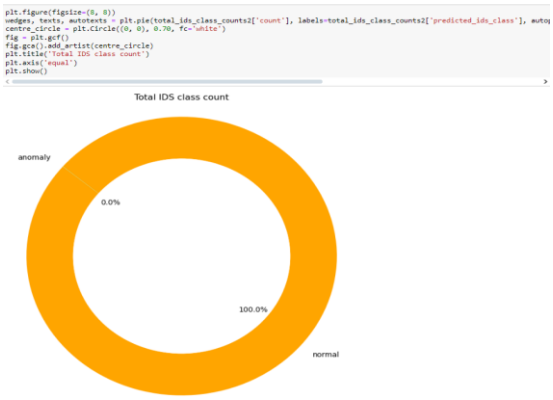


Figure 6.18: Executed data outcome plot for test 2

(Source: Own-Evaluated)

As per the plot, there is 100% normal data of IDS, and 0% is the anomaly or threat data in the network.

6.3 Experiment / Test Case 3

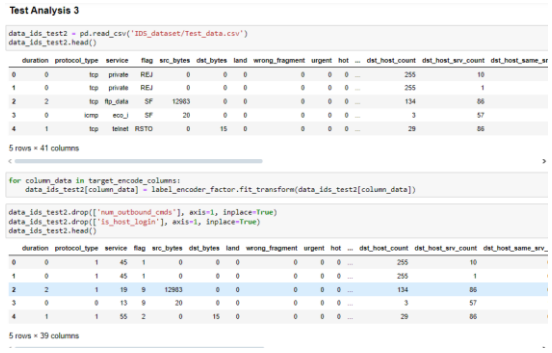


Figure 6.19: Test data setting for the test 3

(Source: Own-Evaluated)

Test case 3 also defines the initialization approach which is applicable for the execution.

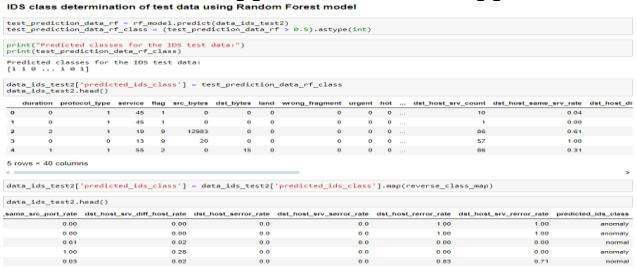


Figure 6.20: IDS class execution using RF and outcome for test 3

(Source: Own-Evaluated)

In this case, an RF prototype is implemented for the detection of the IDS classes. The predicted IDS class column is executed and defines the IDS class value for various data rows. It highlights the normal, and abnormal cases (Zhang *et al.*, 2021).

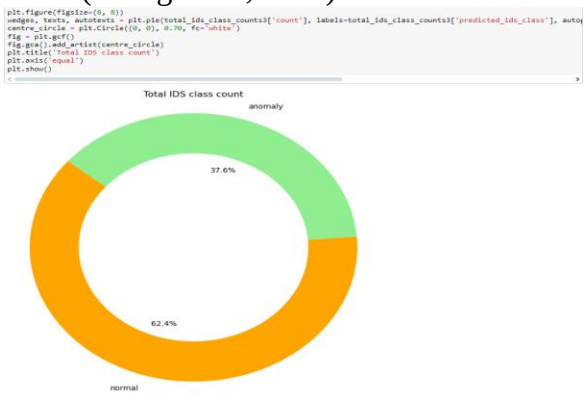


Figure 6.21: Executed data outcome plot for test 3

(Source: Own-Evaluated)

As per the plot, there are 62.4% normal cases and 37.6% anomaly cases. This value assists in understanding the presence of the normal, and abnormal IDS classes in the testing data.

6.4 Discussion

Model	**Accuracy**
KNN	98.85%
Decision Tree	99.51%
Random Forest	99.68%
SVM	53.43%
ANN	95.76%

Table 6.1: Various model/prototype comparison

(Source: Own-Executed)

The tabular data values provide information about the accuracy section of each model. As per the data, Random Forest has the maximum accuracy section which is 99.68%. So, this is one of the suitable models for the execution. On the other hand, SVM has the lowest accuracy value which is 53.43% which is not suitable for the detection/prediction.

Test Case	**Model**	**Anomaly**	**Normal**
Test 1	ANN	40.4%	59.6%
Test 2	SVM	0%	100%
Test 3	Random Forest	37.6%	62.4%

Table 6.2: Test Case Evaluation

(Source: Own-Executed)

The three cases are applicable for the detection of the anomaly, and normal case percentage for the IDS. As per the execution, Test 1, and Test 3 have a good percentage of normal and anomaly detection. On the other hand, Test 2 highlights that there are no anomalies present in the network which means that this system cannot detect the anomaly in the network. In tests 1 and 2, ANN, and Random Forest prototypes are implemented. So, those two models are best suitable for the detection of network intrusion.

Chapter 7: Conclusion and Future Work

7.1 Conclusion

The all-over execution supports the process of determination of the intrusion detection for a network. The network data analysis approach is implemented to evaluate the necessary data. The data execution approach provides information about the details of the necessary data factors. This determines the context of the functional factors that are usable for the construction of the IDS. The system designing approach is implemented for the evaluation of the data. The major finding executes the process of determination which assists in understanding the most suitable ML prototypes for the detection. The execution function provides information about the handling of necessary data factors that are usable for the execution. Three test cases are implemented to determine the most suitable model for the detection. As per the executable evaluation, ANN, and Random Forest are the best suitable prototypes for the designing of the IDS. This has a high level of accuracy with good and perfect outcomes.

The approach of determination provides the major fining of the overall detection approach. The detection approach provides information about the handling of the construction of the execution approach. The process of execution determines the process of execution for the determination. The major finding supports the executable approaches that are implemented for the evaluation. The final determination provides information about the approach of the IDS. The overall approach also determines the enhancement technique of the execution process. This provides information about the handling of various key execution approaches to understand the main finding for the execution. The overall execution supports the process of finding IDS factors to deter the intrusion easily.

7.2 Future Work

The future development supports the configuration of the functional setup which is applicable for the formation of the most effective IDS. This system will have error-free approaches to increase the quality of the detection process. The future development also defines the implementation of the network data handling approach that is applicable for the execution of network threats. The threats determine the risk which is known as the anomalies of the network (Yaseen, 2023). The threat detection approach is applicable for the removal of vulnerable factors from the network. The advancement and AI technology need to be implemented for more advance detection approach in the intrusion detection process.

Reference List

Michael Swanagan, C. (2023) Intrusion detection vs prevention systems: What's the difference?, PurpleSec. Available at: https://purplesec.us/intrusion-detection-vs-intrusion-prevention-systems/ (Accessed: 29 July 2024).

Bunny.net (2020) What is network Intrusion Detection System (NIDS)?, bunny.net. Available at: https://bunny.net/academy/security/what-is-network-intrusion-detection-nids/ (Accessed: 29 July 2024).

Saranya, T., Sridevi, S., Deisy, C., Chung, T.D. and Khan, M.A., 2020. Performance analysis of machine learning algorithms in intrusion detection system: A review. Procedia Computer Science, 171, pp.1251-1260.

Einy, S., Oz, C. and Navaei, Y.D., 2021. The anomaly-and signature-based IDS for network security using hybrid inference systems. Mathematical Problems in Engineering, 2021(1), p.6639714.

What is signature-based malware detection?: RiskXchange (2023) riskxchange.co. Available at: https://riskxchange.co/1006984/what-is-signature-based-malware-detection/#:~:text=This%20approach%20focuses%20on%20specific,is%20mistaken%20for%20an%20attack. (Accessed: 29 July 2024).

Kwon, H.Y., Kim, T. and Lee, M.K., 2022. Advanced intrusion detection combining signature-based and behavior-based detection methods. Electronics, 11(6), p.867.

Bangui, H., Ge, M. and Buhnova, B., 2022. A hybrid machine learning model for intrusion detection in VANET. Computing, 104(3), pp.503-531.

Saputra, I.P., Utami, E. and Muhammad, A.H., 2022, October. Comparison of anomaly based and signature based methods in detection of scanning vulnerability. In 2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) (pp. 221-225). IEEE.

N-able (2024) Intrusion detection system (IDS): Signature vs. anomaly-based - N-able, N. Available at: https://www.n-able.com/blog/intrusion-detection-system#:~:text=Signature%20vs.-,anomaly%2Dbased%20intrusion%20detection%20systems,content%20of%20email%20subject%20headings. (Accessed: 29 July 2024).

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J. and Ahmad, F., 2021. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), p.e4150.

Saranya, T., Sridevi, S., Deisy, C., Chung, T.D. and Khan, M.A., 2020. Performance analysis of machine learning algorithms in intrusion detection system: A review. Procedia Computer Science, 171, pp.1251-1260.

Jadhav, A.D. and Pellakuri, V., 2021. Highly accurate and efficient two phase-intrusion detection system (TP-IDS) using distributed processing of HADOOP and machine learning techniques. Journal of Big Data, 8(1), p.131.

Mebawondu, J.O., Alowolodu, O.D., Mebawondu, J.O. and Adetunmbi, A.O., 2020. Network intrusion detection system using supervised learning paradigm. Scientific African, 9, p.e00497.
Kumar, A., Abhishek, K., Ghalib, M.R., Shankar, A. and Cheng, X., 2022. Intrusion detection and prevention system for an IoT environment. Digital Communications and Networks, 8(4), pp.540-551.

Shwartz Ziv, R. and LeCun, Y., 2024. To compress or not to compress—self-supervised learning and information theory: A review. Entropy, 26(3), p.252.

Rosa, L., Cruz, T., De Freitas, M.B., Quitério, P., Henriques, J., Caldeira, F., Monteiro, E. and Simões, P., 2021. Intrusion and anomaly detection for the next-generation of industrial automation and control systems. Future Generation Computer Systems, 119, pp.50-67.

Wang, M., Zheng, K., Yang, Y. and Wang, X., 2020. An explainable machine learning framework for intrusion detection systems. IEEE Access, 8, pp.73127-73141.

Sibeoni, J., Verneuil, L., Manolios, E. and Révah-Levy, A., 2020. A specific method for qualitative medical research: the IPSE (Inductive Process to analyze the Structure of lived Experience) approach. BMC Medical Research Methodology, 20, pp.1-21.

Alharahsheh, H.H. and Pius, A., 2020. A review of key paradigms: Positivism VS interpretivism. Global Academic Journal of Humanities and Social Sciences, 2(3), pp.39-43.

Vellido, A., 2020. The importance of interpretability and visualization in machine learning for applications in medicine and health care. Neural computing and applications, 32(24), pp.18069-18083.

Medeiros, M.C., Vasconcelos, G.F., Veiga, Á. and Zilberman, E., 2021. Forecasting inflation in a data-rich environment: the benefits of machine learning methods. Journal of Business & Economic Statistics, 39(1), pp.98-119.

Karopoulos, G., Kambourakis, G., Chatzoglou, E., Hernández-Ramos, J.L. and Kouliaridis, V., 2022. Demystifying in-vehicle intrusion detection systems: A survey of surveys and a meta-taxonomy. Electronics, 11(7), p.1072.

Jimmy, F.N.U., 2024. Cyber security Vulnerabilities and Remediation Through Cloud Security Tools. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 2(1), pp.129-171.

Rai, R. and Sahu, C.K., 2020. Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (cps) focus. IEEe Access, 8, pp.71050-71073.

Khalil, R.A., Saeed, N., Masood, M., Fard, Y.M., Alouini, M.S. and Al-Naffouri, T.Y., 2021. Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications. IEEE Internet of Things Journal, 8(14), pp.11016-11040.

Ahmad, I., Shahabuddin, S., Malik, H., Harjula, E., Leppänen, T., Loven, L., Anttonen, A., Sodhro, A.H., Alam, M.M., Juntti, M. and Ylä-Jääski, A., 2020. Machine learning meets communication networks: Current trends and future challenges. IEEE Access, 8, pp.223418-223460.

Kasula, B.Y., 2023. Harnessing Machine Learning for Personalized Patient Care. Transactions on Latest Trends in Artificial Intelligence, 4(4).

Zhang, J., Pan, L., Han, Q.L., Chen, C., Wen, S. and Xiang, Y., 2021. Deep learning based attack detection for cyber-physical system cybersecurity: A survey. IEEE/CAA Journal of Automatica Sinica, 9(3), pp.377-391.

Yaseen, A., 2023. AI-driven threat detection and response: A paradigm shift in cybersecurity. International Journal of Information and Cybersecurity, 7(12), pp.25-43.