

BINDURA UNIVERSITY OF SCIENCE EDUCATION

FACULTY OF SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE



**“Enhancing Network Intrusion Detection with an
Ensemble of Deep Learning and Machine
Learning Algorithms”**

By

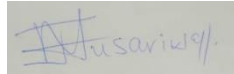
STUDENT NAME	Chisero Tawanda Jonathan
REG NUMBER	B213492B
PROGRAMMEE	HBSc NETWORK ENGINEERING
COURSE NARRATION	RESEARCH PROJECT
COURSE CODE	NWE400
LEVEL	4.2
SUPERVISOR	Mr. J. Musariwa

Approval Form

TITLE: Enhancing Network Intrusion Detection With An Ensemble of Deep Learning and Machine Learning Algorithms

TO BE COMPLETED BY THE SUPERVISOR

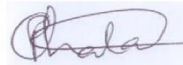
I certify that, to the best of my knowledge, this dissertation meets all requirements and preparation criteria.



Supervisor's Signature:

Date: 14/08/2025

TO BE COMPLETED BY



THE DEPARTMENT CHAIRPERSON

I certify that, to the best requirements and preparation criteria.

of my knowledge, this dissertation meets all

Chairperson's Signature:

Date: 06/08/2025

DEDICATION

This research is a tribute to my late brother, Kudzanai Chisero.

ACKNOWLEDGEMENTS

First and foremost, I give all glory and thanks to the Lord Jesus Christ for the gift of life, the strength to endure, and for seeing me through the entire research journey. I want to deeply appreciate my parents, Mr. James and Mrs. Smolly Chisero, for their constant love and guidance. This wouldn't have been possible without the support from my family.

To my friends Timothy Isheanesu Chitakasha, Tonderai Mabasa Gonyora, Chiedza Kagande, Tanaka Pride Dube and Polite Tanyanyiwa—thank you for always showing up, checking in, and keeping me grounded. Your presence and words of motivation meant more than you know.

Special thanks go to Mr. Foster Akaketwa, the Chief Information Officer at Cimas Health Group, for the role he played in supporting my studies. And to my academic supervisor, Mr. Jeremiah Musariwa—thank you for pushing me beyond my limits and challenging me to tap into my full potential. Your guidance made a real difference.

ABSTRACT

This project seeks to enhance network intrusion detection through an Ensemble model that combines Machine and Deep Learning algorithms. The model integrates the Multilayer Perceptron (MLP), Random Forest Classifier and the Support Vector Machines (SVM). This approach is meant to address the limitations of standalone classifiers by combining their complementary strengths to improve the system's ability in identifying complex intrusion patterns and adapt to evolving cyber threats. The author aimed at developing a reliable network intrusion detection tool that network security professionals can trust in.

To evaluate its practical applicability and performance, the trained ensemble model is deployed within a simulated network environment through a simple web application using Streamlit. This interface allows for real-time intrusion prediction and evaluation of the system's ability to classify network traffic accurately. Ultimately, the project contributes to the broad field of safeguarding networks in a digital landscape where cyberthreats are evolving and becoming more complex to detect with traditional methods.

CONTENTS

RESEARCH TOPIC.....	Error! Bookmark not defined.
DEDICATION	
ACKNOWLEDGEMENTS.....	iii
ABSTRACT.....	iv
CONTENTS.....	v
List of Figures	viii
ABBREVIATIONS	1
CHAPTER 1	2
“Problem Identification”	2
1.1 INTRODUCTION.....	2
1.2 BACKGROUND TO THE STUDY.....	2
1.3 STATEMENT OF THE PROBLEM	3
1.4 RESEARCH OBJECTIVES.....	4
1.5 RESEARCH QUESTIONS.....	4
1.6 PROPOSED SOLUTION	4
1.7 RESEARCH HYPOTHESIS	5
1.8 SIGNIFICANCE OF THE STUDY	5
1.9 ASSUMPTIONS.....	5
1.10 LIMITATIONS	6
1.11 SCOPE OF THE RESEARCH	7
1.12 DEFINITION OF TERMS	7
CHAPTER 2	9
“Literature Review”	9
2.1 INTRODUCTION.....	9
2.2 THEORETICAL FRAMEWORK	9
2.3 EMPIRICAL LITERATURE REVIEW.....	10
2.4 CHAPTER SUMMARY	12
CHAPTER 3	13
“Research Methodology”	13
3.1 INTRODUCTION.....	13
3.2 RESEARCH DESIGNS AND DATA COLLECTION APPROACHES	13
3.2.1 RESEARCH DESIGNS	14
3.2.1.1 Exploratory Research	14
3.2.1.2 Qualitative Research	14

3.2.1.3 Experimental Research	15
3.2.2 DATA COLLECTION APPROACHES	15
3.3 POPULATION AND SAMPLES.....	16
3.3.1 Population	16
3.3.2 Samples	16
3.4 RESEARCH INSTRUMENTS	17
3.4.1 Semi-structured Interviews.....	17
3.4.1.1 Sample Interview Questions	17
3.4.2 Questionnaires	18
3.5 DATA ANALYSIS PROCEDURE	18
3.5.1 Qualitative Data Analysis	18
3.5.2 Quantitative Data Analysis.....	18
3.6 PROJECT OVERVIEW	19
3.7 CHAPTER SUMMARY	19
CHAPTER 4	21
“Data Presentation, Analysis and Interpretation”	21
4.1 INTRODUCTION	21
4.2 ANALYSIS AND INTERPRETATION OF MODEL TRAINING RESULTS	21
4.2.1 Overview of the Participating Network Traffic Datasets	21
4.2.2 Model Training Environment and Resource Utilization	22
4.2.3 Model Training and Performance Evaluation	23
4.2.3.1 Graphical Representation of Model Performance.....	25
4.2.4 Comparative Performance Analysis of Models.....	25
4.2.5 Interpretation of Model Training Results	26
4.3 MODEL PERFORMANCE EVALUATION USING STREAMLIT.....	27
4.3.1 Model Deployment and Interface Overview	27
4.3.1.1 Single Packet Prediction Analysis.....	27
4.3.1.2 Whole Dataset/ Batch Analysis.....	29
4.3.2 Interpretation of Model Testing Results.....	31
4.4 USER FEEDBACK ANALYSIS.....	31
4.4.1 User Responses and Rating Metrics.....	31
4.4.3 User Feedback Analysis Summary	32
4.5 SUMMARY OF RESEARCH FINDINGS	32
CHAPTER 5	34
“Conclusions And Recommendations”	34
5.1 INTRODUCTION	34

5.2 MAJOR CONCLUSIONS DRAWN	34
5.3 LIMITATIONS AND UNDERDEVELOPED ASPECTS OF THE STUDY	35
5.3.1 Real-Time Deployment Constraints	36
5.3.2 Dataset Dependence.....	36
5.3.3 Evaluation Metrics Focused Primarily on Accuracy	36
5.3.4 User Feedback Was Limited	36
5.4 RECOMMENDATIONS.....	36
5.5 FINAL REMARKS	37
REFERENCES	38
APPENDICES	43
APPENDIX 1: INTERVIEW 1 (with a Cybersecurity Officer).....	44
APPENDIX 2: INTERVIEW 2 (with a Network Administrator)	46
APPENDIX 3: INTERVIEW 3 (with a Network Engineer)	48
APPENDIX 4: QUESTIONNAIRE RESPONSES	50

List of Figures

Figure 2.1 “Ensemble Learning Theory”	Error! Bookmark not defined.
Figure 3.1 “Experimental Research Aspects”	Error! Bookmark not defined.
Figure 3.2 “Project Overview”	Error! Bookmark not defined.
Figure 4.2 “CPU Utilization During Multilayer Perceptron Model Training”	Error! Bookmark not defined.
Figure 4.1 “CPU Utilization During Ensemble Model Training”	Error! Bookmark not defined.
Figure 4.3 “Performance Metrics Results for the Support Vector Machine Model”	Error! Bookmark not defined.
Figure 4.4 “Performance Metrics Results for the Multilayer Perceptron Model”	Error! Bookmark not defined.
Figure 4.5 “Performance Metrics Results for the Random Forest Model”	Error! Bookmark not defined.
Figure 4.6 “Performance Metrics Results for the Ensemble Model”	Error! Bookmark not defined.
Figure 4.7 “F1-Score Results”	Error! Bookmark not defined.
Figure 4.8 “Network Intrusion Prediction Main Menu”	Error! Bookmark not defined.
Figure 4.10 “Second Single Packet Prediction”	Error! Bookmark not defined.
Figure 4.9 “First Single Packet Prediction”	Error! Bookmark not defined.
Figure 4.12 “Selected Network Parameters”	Error! Bookmark not defined.
Figure 4.11 “Third Single Packet Prediction”	Error! Bookmark not defined.
Figure 4.13 “Normal vs Attack for Batch Analysis”	Error! Bookmark not defined.
Figure 4.15 “Prediction Field Added to Uploaded Dataset”	Error! Bookmark not defined.
Figure 4.14 “Multiclass Classification for Batch Analysis”	Error! Bookmark not defined.
Figure 4.16 “User Feedback Analysis”	Error! Bookmark not defined.

ABBREVIATIONS

SVM – Support Vector Machine

NIDS – Network Intrusion Detection System

IDS – Intrusion Detection System

ML – Machine Learning

DL – Deep Learning

ANN – Artificial Neural Network

CNN – Convolutional Neural Network

MLP – Multilayer Perceptron

SYN – Synchronise

UDP – User Datagram Protocol

HTTP – Hyper-text Transfer Protocol

SQL – Structured Query Language

API – Application Programming Interface

IPS – Intrusion Prediction System

CHAPTER 1

“Problem Identification”

1.1 INTRODUCTION

By definition, any system that monitors network traffic data to identify deviations from normal network behaviour is called an Intrusion Detection System (IDS) as articulated by Axelsson (2000). However, with the ongoing evolution of the nature of cyber threats, there is need for a constant adaptation of IDS capabilities to effectively counter increasingly sophisticated attacks.

Cybercriminals are employing advanced techniques to penetrate traditional security measures hence; the digital threat landscape is ever-changing. This involves the emergence of Advanced Persistent Threats (ATPs) which target specific organisations over extended periods as described in an article by Stallings & Brown (2018). Mell & Scarfone (2007) goes on to state that these types of attacks are often deployed in multiple phases, making them immune to detection by conventional signature-based techniques.

Compensating users who has their data compromised can pose the company at serious losses of up to billions of dollars after a successful cyberattack. Such incidents highlight that it is costly to neglect basic cybersecurity practices which also lead to a tarnished brand image of the company (FTC, 2019).

Mitnick and Simon (2002) suggests that, for one to prevent attacks and threats, he has to first understand the strategies that the attacker is most likely to employ. As a result, this paper sets out to explore the effectiveness of an ensemble of traditional Machine learning and Deep Learning algorithms to enhance intrusion detection of zero-day and emerging threats.

1.2 BACKGROUND TO THE STUDY

During the first years of network security, analysts manually browsed through system logs in search of abnormal activities within the system. The automation of Intrusion Detection Systems (IDS) was introduced in the early 1990s, but it was more of signature-based techniques. Unfortunately, when it came to the detection of zero-day attacks, these techniques were not effective because they categorized traffic based on predefined rules (Denning, 1987).

From signature-based techniques, new methods were developed. These include statistical anomaly detection which was trained to recognize normal behaviour in the network based on pre-programmed rules and label the rest as intrusions (Sommer & Paxson, 2010). This method offered better capabilities over preceding solutions; however, it falsely classified normal network behaviour which had small deviations from known patterns.

As the years went by, Machine Learning (ML) came on the scene and it marked a civilization in the growth of intelligent Intrusion Detection Systems (IDS). ML algorithms are able to pinpoint hidden trends from a huge source of network data, aiding in separating known from unknown threats with guaranteed high accuracy rates. Machine Learning algorithms have the ability to learn, improve their behaviour and increase in the knowledge of analysing patterns within a dataset mimicking human behaviour (Buczak & Guven, 2016). This resulted in ML-based IDS being far more effective than their predecessors. The capabilities of ML algorithms have positively impacted today's digital world like never before.

Later on, Deep Learning (DL) a subfield of ML came into play and it has further aided the cybersecurity field since its emergence. Deep Learning makes use of functional models and frameworks that are composed of multiple layers to learn patterns in data through many levels of iteration (LeCun et al., 2015). While other methods can sometimes overlook minute indicators of malicious activity, DL-based IDS is good at that.

Ashiku and Dagli (2021) conducted a study on the integration of Machine Learning algorithms in detecting network attacks. Their research yielded a 95.4% overall accuracy rate. Not to cast down the importance of their findings, but a 4.6% inaccuracy window is too wide to let attackers enter the network undetected, hence, that's a threat.

This author reimagines a different approach to intrusion detection, that is, to develop an ensemble model that combines Machine and Deep Learning algorithms. This mechanism will employ each participating algorithm's strengths thereby improving accuracy and reducing false positive rates.

1.3 STATEMENT OF THE PROBLEM

The adoption of standalone traditional Machine Learning (ML) algorithms in detecting network intrusions has positively aided network security. Nevertheless, these approaches while offering positive advancements, are not without limitations. Previous studies prove that standalone

techniques are most likely to miss some intrusions, and that can lead to destructive repercussions such as financial loss, reduced customer loyalty, regulatory penalties and compromised critical infrastructure. Furthermore, existing solutions are known for their high rates of false positives which potentially results in actual threats being ignored. Hence, this necessitates further research and the development of more reliable solutions.

1.4 RESEARCH OBJECTIVES

1. To design and implement an ensemble model that leverages machine and deep learning algorithms to analyse network traffic data, and identify both known and unknown attacks.
2. To achieve high accuracy and low false positive rates of intrusion.
3. To assess trust and satisfaction of network security professionals in the performance of the proposed solution.

1.5 RESEARCH QUESTIONS

1. How to design and implement an ensemble model that is capable of detecting a diversity of network intrusions?
2. What are the ways to achieve high accuracy and low false alerts of intrusions using a Machine Learning-based NIDS?
3. How to implement a user-centric approach during the development of a NIDS to meet network security personnels' requirements?
4. How can an ensemble of machine and deep learning techniques be optimised for real-time adaptability in dynamic network environments?
5. What novel metrics can be developed to assess and evaluate the performance of the hybrid IDS beyond traditional accuracy scores and false alert rates?

1.6 PROPOSED SOLUTION

The author proposes an ensembled model of the Support Vector Machines (SVM), Random Forest Classifier and the Multilayer Perceptron (MLP) to point out network intrusions. The proposed system is a promising mechanism in the improvement of intrusion detection accuracy, and efficiency as well as network administrators' trust in advanced technologies. This approach seeks to effectively make use of the strengths of deep and machine learning to curb the risk of misclassifying evolving cyber threats. The trained model will be loaded onto a Streamlit Python-coded web application where its ability to detect intrusions is to be evaluated.

1.7 RESEARCH HYPOTHESIS

The null hypothesis is simply an assertion which argues that one variable in the population has no effect on the other variable. In contrast, the alternative hypothesis is a statement which supports the idea that an instance can change or bring a difference to another variable in the population (Gravetter & Wallnau, 2014). Below are the hypothesis statements of the proposed solution:

Null Hypothesis (H₀): There is no change in intrusion detection accuracy and trust levels of network security professionals in the ensemble model in comparison to standalone traditional solutions.

Alternative Hypothesis (H₁): An ensemble of MLP, SVM and Random Forest Classifier will significantly improve intrusion detection accuracy and network security professionals' trust in hybrid approaches over standalone methods.

1.8 SIGNIFICANCE OF THE STUDY

The proposed ensemble model offers a valid and relevant aspect of pattern-learning that can positively improve intrusion detection accuracy and efficiency. As alluded earlier, this method provides a more effective solution for dealing with modern-day cyber threats by joining the strongest abilities of each of the individual algorithms.

The Multilayer Perceptron (MLP) is capable of recognizing intricate patterns and extracting features from network traffic data, whether labelled or unlabelled, with minimal human input. The Random Forest Classifier mainly performs well in classifying data using the features extracted by the MLP. It is relatively simple, which makes it quicker to train as compared to other methods. Likewise, the Support Vector Machine (SVM) groups data effectively but requires less computational resources.

This research addresses the enhancement of intrusion detection accuracy by assembling deep and traditional machine learning algorithms. Resultantly, it lays a foundation for future research to improve security in edge computing and Internet of Things (IoT) networks.

1.9 ASSUMPTIONS

Assumption 1: *A sufficient amount of labelled network traffic data, both anomalous and normal is available for training and testing the model.*

The outcome of any supervised model, particularly in the detection of intrusions relies on the quality and quantity of training data.

Assumption 2: *It is assumed that the dataset is representative of real-world network traffic.*

If the training data does not accurately reflect the characteristics of real-world network traffic, the model may fail to generalize effectively after deployment.

Assumption 3: *The author assumes that the outputs or decision metrics of the MLP, SVM and Random Forest Classifier models can be merged effectively using the Byzantine voting method.*

The Byzantine voting method has been selected to reduce the impact of potentially biased predictions from individual models, enhancing the reliability of the intrusion detection system. The errors of the individual models will not be correlated.

Assumption 4: *Adequate computational resources are available for training and running the ensemble model.*

If resources are not sufficient, the training time of the model might be too long to be practical, or the model may not be able to be deployed before the project submission deadlines.

Assumption 5: *The trained ensemble model will generalize well to unseen network traffic and new, potentially emerging attack types.*

This is a critical requirement for determining the long-term usefulness of the NIDS.

Assumption 6: *It is assumed that the selected solutions are able to identify the attacks they are meant to defend against.*

MLP is capable of recognizing intricate patterns and extracting features from network traffic data with minimal human input. SVMs and Random Forest Classifier are effective for classifying network traffic based on extracted features; hence, the correct selection of algorithms is vital to the success of the NIDS.

1.10 LIMITATIONS

Limitation 1: *Network traffic data is often noisy, high-dimensional and imbalanced, this can negatively impact the model's overall performance.*

Mitigation: Preparing data using techniques such as feature selection, and data balancing as suggested by (Chawla et al., 2002).

Limitation 2: *MLP requires a lot of computational resources and large datasets with labels for efficiency.*

Mitigation: Acquire the essential datasets from an open-source pool of datasets such as Kaggle's website.

Limitation 3: *It is challenging to keep the model relevant in terms of it being able to identify emerging threats when there is no continuous monitoring and retraining.*

Mitigation: Every system requires continuous monitoring and evaluation to ensure it is relevant especially these days when threats keep evolving over time.

Limitation 4: *It is difficult to integrate multiple models into one and this can hinder the development process because deep domain knowledge is required.*

Mitigation: The author is going to deepen and broaden his knowledge by studying around the areas related to this research.

1.11 SCOPE OF THE RESEARCH

- The following attack types will be of focus for this study; teardrop, Shellcode, Neptune, multihop, SQL injection, spy, exploits, smurf, warezmaster, Ping of Death, reconnaissance, HTTP flood, Backdoor, fuzzers, zero-day exploits, ftp_write, DNS poisoning, worms, SYN flood, DoS, UDP flood, generic.
- For this paper, NSL-KDD will be used for testing and the UNSW- NB15 dataset for model training.
- This study will evaluate the system's performance using the following metrics; recall, accuracy, F1-score, false positive rate and precision.
- The scope of this research is to primarily focus on detecting network intrusions (that is normal, abnormal or emerging) and assessing the trust of network security professionals in the proposed ensemble.

1.12 DEFINITION OF TERMS

Network Intrusion Detection System (NIDS) – it is any system that browses through network traffic whilst checking for unusual activity or policy violations (Kurose & Ross, 2017).

Machine Learning – refers to the technology that employs computational procedures designed to learn and identify hidden trends in data mimicking human intelligence (Naqa & Murphy, 2015).

Deep Learning – it is a better version of Machine Learning that enables algorithms with multiple layers be able to learn and fetch increasingly complex and useful elements from large amounts of data as defined by (Goodfellow et al., 2016).

Support Vector Machine (SVM) – is a technology that learns from a labelled dataset and is mainly used for classification purposes.

Multilayer Perceptron – belongs to a group of neural networks that pass inputs forward through multiple layers, enabling them to learn and identify complex nonlinear relationships (Patra et al., 2008).

Alert Fatigue – According to Kane-Gill (2017), alert fatigue refers to the desensitisation that occurs when individuals are exposed to an overwhelming number of alerts, leading to missed or ignored critical warnings.

Intrusion Prediction System (IPS) – addressing not only intrusion detection, IPS is capable of predicting potential future intrusions by analysing patterns and trends in network traffic, thereby enabling proactive defence strategies as described by Abdlhamed & Kifayat (2018)

CHAPTER 2

“Literature Review”

2.1 INTRODUCTION.

Whilst taking a close look at the domain of network security, this chapter opens the curtain to the existing studies relevant to the proposed field of study. It lays out the ideas that can be used to build the proposed solution and also shows the solutions that have already been tried before. Focus is to be channelled towards authors who harnessed the power of Machine and Deep Learning algorithms to predict network intrusion. Most importantly, the author is going to put much emphasis on the features, datasets and algorithms, as well as the performance evaluation of existing solutions.

2.2 THEORETICAL FRAMEWORK

This study is grounded in the **Ensemble Learning Theory**, which states that combining diverse weak or strong learners can produce a model with extraordinary generalisation and predictive performance in comparison with any standalone model (Dietterich, 2000).

When it comes to intrusion detection, where data in most cases is high-dimensional, imbalanced, and dynamic, relying on a single classification algorithm can limit accuracy and adaptability. As a result, the Ensemble Learning Theory is the perfect framework to integrate three different models: a Support Vector Machine (SVM), Multilayer Perceptron (MLP) and a Random Forest Classifier. Every one of these models brings about unique learning capabilities. MLP excels in nonlinear pattern recognition, SVM is proficient in decision boundary optimisation, and Random Forests offer high interpretability and robustness to noise.

The theoretical foundation for this aggregation is influenced by the Bias-variance Trade-off principle as illustrated in *Figure 2.1* below. Ensemble models seek to create a balance between the high variance of complex learners like MLPs with the lower variance, high-bias traits of models like SVMs and decision trees, and this yields a more stable and accurate classifier. The ensemble mechanism is clearly depicted in *Figure 2.1*, where multiple predictors are trained on the same data and their predictions are combined through majority voting to form a final

decision. This aggregation reduces variance without substantially increasing bias, thus achieving better generalisation (Geurts et al., 2006; Zhou, 2012).

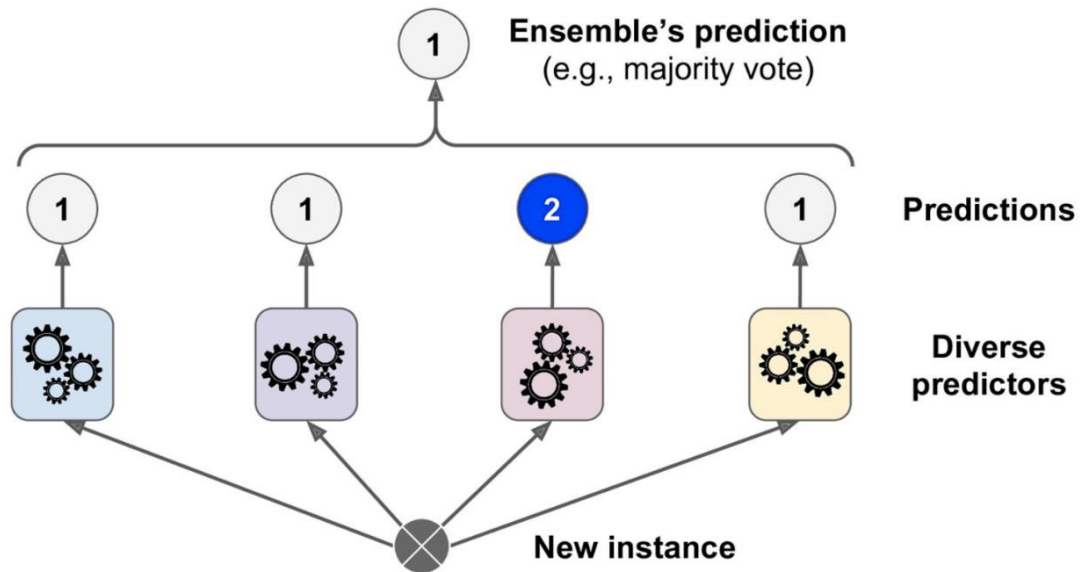


Figure 2.1 "Ensemble Learning Theory"

This framework supports the development of an intelligent and adaptive intrusion detection system capable of detecting both known and novel attacks in real-time. It reduces false positives significantly, which is one of the most frustrating issues in IDS and shows why combining different models makes sense.

2.3 EMPIRICAL LITERATURE REVIEW

To get a better understanding of how effective Machine and Deep Learning (DL) techniques are when it comes to network intrusion detection, the paper explores deep into several recent and old studies that used different approaches. The most prominent argument across the domain is that, although individual algorithms work well, combining them in a hybrid setup produces better results, especially when detecting new or unknown attacks.

In a study, Ashiku and Dagli (2021) employed a deep learning setup that combined Convolutional Neural Networks (CNNs) and a regularised Multilayer Perceptron (MLP). They used the UNSW-NB15 dataset, which is publicly known for housing real-world network traffic patterns, and their model ended up achieving a detection accuracy of 95.4%. This is somehow good, especially considering how crucial it is to catch zero-day attacks, and it stands to prove how DL models are able to pick up on patterns that traditional systems might miss.

On a similar note, Sharaf and Haggag (2017) also developed their deep learning-based IDS using the NSL-KDD dataset. Their results highlighted that deep learning algorithms outperforms traditional machine learning techniques, mainly because of their ability to identify hidden patterns within the network data.

Zhang et al. (2022) took it a step further and performed a comparison on a bunch of ML models, including, Naïve Bayes, SVMs, CNNs, and RNNs. These models were trained and tested using the KDDCUP99 and NSL-KDD datasets. Interestingly, they found out that Naïve Bayes did quite well when it came to picking up on new types of attacks, mainly because of its faster training time. But resultantly, the ensemble and DL-based models performed better when accuracy was the main priority.

In a similar study, Almutairi et al. (2022) looked at several ML models and used the NSL-KDD dataset to test both binary and multiclass classification. Their findings showed that using a diversified approach yielded better results than using just one. It goes on to state that using multiple models gives the system an advantage in identifying threats in a better way.

Turning attention to feature selection, Taher et al. (2019) thought of a smart approach which combines Artificial Neural Networks (ANNs) with a wrapper feature selection method. Like most researchers, they also used the NSL-KDD dataset and compared their results against SVMs. Their solution turned out to be more accurate, which underpins the fact that having the right features plays a vital role in getting better detection performance.

Building upon this foundation, Talukder et al. (2022) came through with an ensemble which used ML elements only. They handled class imbalance using the Synthetic Minority Over-sampling Technique (SMOTE) and also applied XGBoost for feature selection. The model resulted in impressive results on the KDDCUP'99 and CIC-MalMem-2022 datasets. Their work proves that preprocessing dataset can make a huge difference during training. However, they stated that ML models are not good at data classification especially when there is no pre-processing.

Looking at broader reviews, Ahmed and Traore (2014) aided the domain of knowledge with a comprehensive survey that addressed the advantages and disadvantages of many ML algorithms like SVMs, Decision Trees, and Naïve Bayes. All this shows that preparing a dataset before training can positively impact performance and predictions.

Likewise, Buczak and Guven (2016) addressed the issue of unprocessed datasets in research and pinpointed how ensemble methods can help in building more accurate systems. Their findings line up with the idea that no one model is perfect, but by combining methods, you unleash the real strength.

Lastly, Vinod and Reberio (2017) used a deep learning approach that automated the process of feature extraction from raw network traffic. The ability of DL models to find useful features without human intervention gives them a big edge in real-time detection scenarios, especially as traffic patterns grow more complex and complicated.

2.4 CHAPTER SUMMARY

A synthesis of reviewed literature reveals compelling evidence that ML and DL algorithms have a lot to offer when it comes to building intelligent intrusion detection systems. However a notable challenge is that, each model has its limitations. When you bring MLPs, SVMs, and Random Forests together, you get a system that's more accurate, less prone to errors, and better at handling the diverse nature of network traffic.

This study sets the stage for taking this hybrid method into the space of low-powered IoT devices. Going forward, the focus should be on making these models lighter and more efficient using mechanisms like compression and edge computing to ensure they can run smoothly even where resources are limited.

Apart from IoT, this ensemble model also provides viability for cloud-based security solutions. With cloud systems now handling most of the data storage and processing, having strong intrusion detection in place is a must, not something to overlook. The evidence is irresistible: ensembles outperform isolated models. What this research contributes is a pathway for scaling that power into real-world, resource-limited settings ranging from IoT devices to cloud systems where the battle against evolving cyber threats is most urgent.

CHAPTER 3

“Research Methodology”

3.1 INTRODUCTION

This part of the paper addresses the research methodology and steps to be taken by the author towards fulfilling the research objectives. Regarding Kothari (2004), research methodology refers to the science of studying how research is done scientifically, involving the logic behind research methods and the ability to evaluate the relevance of particular research techniques to solve specific problems. Extending this definition, Creswell & Creswell (2017) suggested that research methodology involves theoretical assumptions, research designs, data collection and analysis techniques used by researchers to address a research problem. In simpler terms, research methodology can be likened to a blueprint or strategy used to solve a problem, for example, a recipe for baking a cake. This chapter introduces the key components ‘*secret ingredients*’ to achieving high accuracy in network intrusion prediction using Machine (ML) and Deep Learning (DL) algorithms.

3.2 RESEARCH DESIGNS AND DATA COLLECTION APPROACHES

A research design can simply be defined as the overall strategy that researchers use to answer their research questions (Bryman, 2016). These designs offer a basis for how data is to be collected, analysed, interpreted and presented. Its main purpose is to help the researcher in ensuring that the findings are valid, reliable and relevant to the research problem at hand.

Data collection approaches are mechanisms used to gather data from various sources for analysis and interpretation. It is the activity that operates close to the real-world scenarios and collects all the necessary information that can be used for analysis to make the world a better place. Provost & Fawcett (2013) argue that data is the foundation for intelligent systems, and without high-quality, relevant data, even the most sophisticated algorithms cannot produce meaningful results.

Given the diverse nature of the research questions in this study, no single research design is sufficient to comprehensively address all objectives. Therefore, a Mixed Methods Research Design is adopted, which strategically integrates exploratory, qualitative, and experimental approaches. Each method is selected based on its suitability to the specific research question it aims to address, as outlined in the subsequent sections.

3.2.1 RESEARCH DESIGNS

3.2.1.1 Exploratory Research

This is the most appropriate method when the problem is not clearly defined. Its main function is to help researchers familiarise themselves with a topic, refine the research focus, and determine suitable methods and strategies (Saunders et al., 2019).

In this study, exploratory research informs key areas of investigation, such as evaluating the strengths and limitations of diverse ML and DL models for network intrusion detection. It also supports the revelation of techniques for adapting ensemble models to dynamic network environments, specifically through approaches like online learning, concept drift detection, and transfer learning.

Furthermore, this design aids in examining effective feature engineering strategies that extract meaningful insights from network traffic data to improve model performance. By enabling an open-ended exploration of these themes, exploratory research lays the foundation for more structured and targeted analysis in subsequent phases of the study.

3.2.1.2 Qualitative Research

Creswell & Poth (2018) define qualitative research as an approach centred on understanding the meanings, experiences, and perspectives of individuals or groups, particularly within complex social or human contexts. It is most useful when the goal is to generate deep, contextual insights rather than to quantify data. According to Bryman (2016), this method prioritises non-numerical forms of data, including interviews, observations, images, and textual content to explore underlying patterns and experiences.

In this paper, qualitative research plays a vital role in exploring human factors surrounding the design and deployment of an ensembled network intrusion detection model. To be more specific, it will be used to understand the real-world challenges network administrators face when detecting and responding to network intrusions. This approach also facilitates the identification of their needs, operational workflows, and preferences, which are essential in defining what constitutes a trustworthy and effective NIDS from their perspective. This helps to highlight critical areas for improvement from their standpoint.

During the system testing phase, qualitative feedback will be gathered to evaluate the usability and perceived effectiveness of the proposed ensemble solution. These insights are expected to guide refinements, ensuring the final solution is technically sound and solves problems in real-world environments.

3.2.1.3 Experimental Research

Experimental research falls right under the canopy of quantitative methodologies and is particularly valuable when the goal is to establish causal relationships. According to Field (2018), this approach involves the intentional manipulation of one or more independent variables to assess their effect on corresponding dependent variables, ideally within a controlled environment. This level of control enables researchers to isolate and analyse the specific influence of each variable.

This study uses experimental research to look into three different areas, each focusing on specific factors that help test and improve the proposed ensemble method. These parts will be applied in three main sections, with both the input (X) and output (Y) variables shown in *Figure 3.1* below.

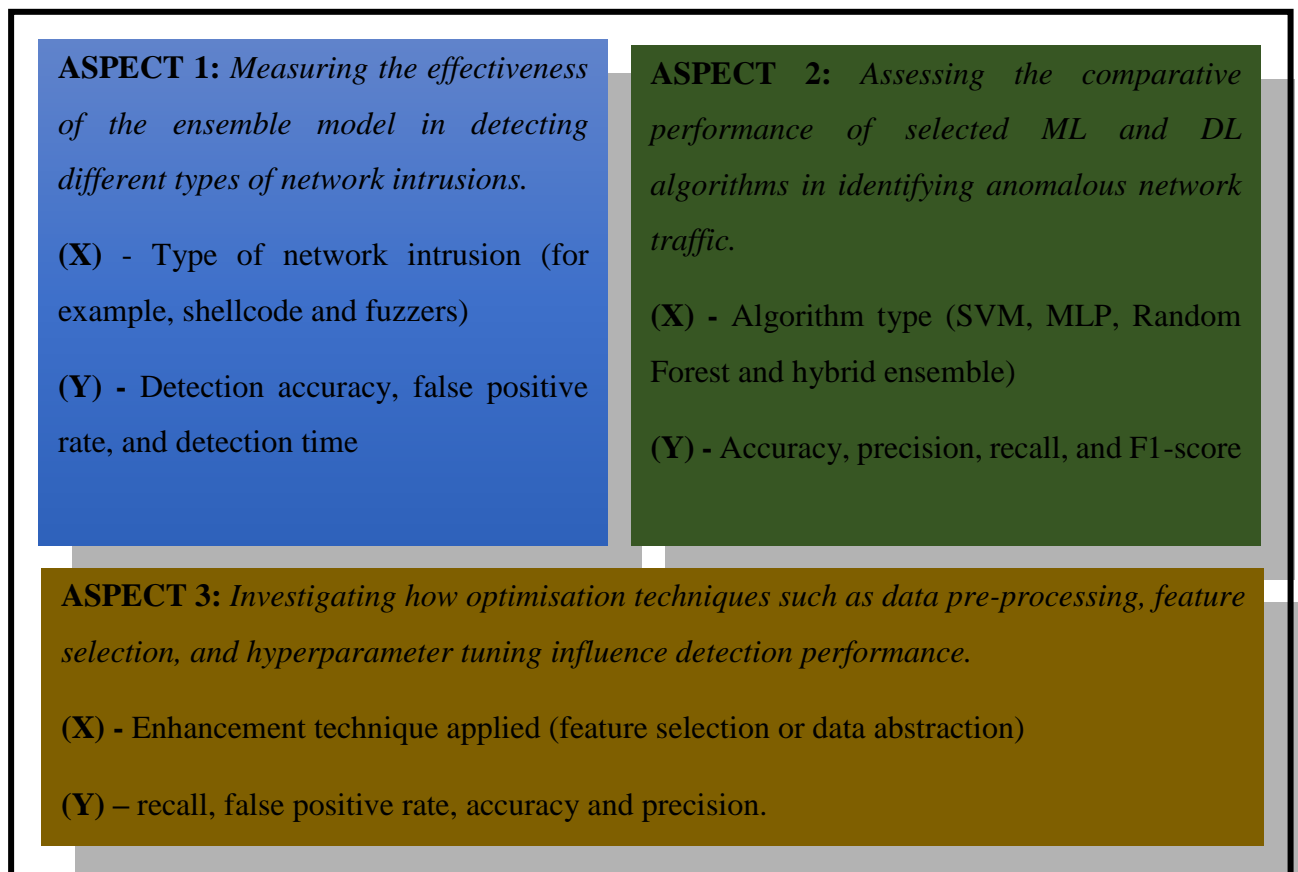


Figure 3.1 "Experimental Research Aspects"

3.2.2 DATA COLLECTION APPROACHES

Data collection refers to the methods employed to gather real-world data for evaluation and analysis. For this research, a mix of quantitative and qualitative data will be collected to support model training, testing, and validation.

In this study, the author chose the UNSW-NB15 and NSL-KDD datasets to obtain network traffic data that is representative of real networks. These datasets will serve as the basis for training and testing the proposed intrusion detection model, provided they contain labelled instances of normal and malicious traffic required for supervised learning.

The performance of the model will be assessed based on F1 score, accuracy, recall, F1-score, precision, false positive rate, and overall system processing time. To ensure the system is something that network engineers can trust, their input during development is essential. As domain experts, their feedback guides system refinement and validation. Interviews and questionnaires will be employed throughout all stages of development to capture user feedback, domain-specific insights, and usability considerations, to ensure that the final solution is both technically sound and practically applicable.

3.3 POPULATION AND SAMPLES.

In research, the population refers to the full group of people or things that the researcher wants to study or draw conclusions about. It is the wider group that shares certain characteristics related to the research (Enago Academy, 2023). A sample, on the other hand, is a smaller part of that population from which data will actually be collected from during the study (Scribbr, 2020).

3.3.1 Population

For this research, the population is broad since the author will focus on the entities that utilise computer networks and require robust network security.

Population: Small to large-sized businesses in the finance and healthcare sector in Zimbabwe.

3.3.2 Samples

Sample 1: The network traffic data for training and testing of the model

Sample size = 2 datasets

NB: The UNSW-NB15 is for training, and the NSL-KDD is for testing purposes.

Sample 2: Organisations in the finance and healthcare sector in Harare.

Sample size = 8

NB: 4 organisations are to be selected for each sector.

Sample 3: Network security professionals from Sample 2.

Sample size = 20 professionals

NB: *these professionals are to be selected randomly from Sample 2.*

3.4 RESEARCH INSTRUMENTS

A research instrument is a tool used to collect data from the real world. Bryman (2016) points out that instruments can include questionnaires, interviews, and observations. According to Cohen, Manion and Morrison (2018), it is important to first consider how suitable the instrument is for the target population. For this study, only interviews and questionnaires will be used to gather data.

3.4.1 Semi-structured Interviews

Because of the specialized roles of network security professionals, semi-structured interviews are ideal for exploring their experience, perspectives, and knowledge related to this study. This method offers flexibility to dive deeper into relevant issues as they come up, while still following a general structure to keep things consistent across participants.

The total sample size of network security professionals is 20, as previously mentioned. From this group, only 4 will take part in interviews. The interviews will focus more on open-ended questions that need detailed answers and context. A smartphone will be used to record each session, and the responses will later be transcribed into text for analysis.

3.4.1.1 Sample Interview Questions

- What are the biggest challenges you face when detecting both known and unknown intrusions with traditional NIDS? Walk me through an example.
- How familiar are you with machine learning in NIDS, and what do you think about its potential?
- Which types of network traffic data do you consider most useful for training intrusion detection models?
- What key metrics do you currently use to measure your NIDS performance?
- Are there other metrics beyond accuracy and false alerts that you think would better reflect NIDS performance?
- What level of automation do you think is needed to shift from traditional to hybrid NIDS?
- Have you experienced alert fatigue due to frequent false alarms? What impact did that have?

- If a hybrid NIDS was implemented, what kind of training or support would help you manage it effectively?

3.4.2 Questionnaires

A questionnaire has the ability to efficiently gather data from the population. Its use of the Likert scale questions allows the researcher to quantify responses on specific areas of the study, such as job satisfaction levels. The author selected this instrument because it works better with large samples, i.e. the remaining 16 network security professionals from the sample of 20. This is advantageous since it can be administered online, allowing the participants to complete the questionnaire from anywhere around the globe as long as they have an internet connection. There are two questionnaires, the first one is to gather expert knowledge and the other one for feedback on system performance. Below are the links to the sample questionnaires. Please note that, responses from the questionnaires are in the appendices.

EXPERT KNOWLEDGE COLLECTION: <https://forms.gle/kUkr3ytNDp26Qq7q9>

FEEDBACK GATHERING: <https://forms.gle/SsT74VJhnQx2mjz78>

3.5 DATA ANALYSIS PROCEDURE

Considering the nature this study which involves both interviews and questionnaires, a mixed-methods approach will be used to analyse the data as recommended by Creswell and Creswell (2018).

3.5.1 Qualitative Data Analysis

The qualitative data will come mainly from the interview transcripts and open-ended responses in the questionnaires. This data gives insight into real experiences and views from network security professionals. The following methods will be used to draw insights from it:

- Thematic Analysis** – This method helps identify common patterns, challenges, and suggestions raised by participants (Braun & Clarke, 2006).
- Content Analysis** – This technique focuses on organising the data into categories and counting how often key ideas or phrases appear (Krippendorff, 2018).

3.5.2 Quantitative Data Analysis

The data for quantitative analysis will be drawn from Likert-scale and multiple-choice questions. The focus will be on understanding overall trends and making comparisons:

- Descriptive Analysis** – Basic statistics like averages and standard deviation will be used to summarise the data (Trochim & Donnelly, 2008).

- b. Correlation Analysis** – This will help explore any links between variables, such as years of experience and satisfaction with current NIDS solutions.

3.6 PROJECT OVERVIEW

The proposed system will be developed in phases as illustrated in *Figure 3.2* below. Each stage builds on the previous one, serving as a roadmap to guide the researcher from project planning through to completion. The diagram below is an illustration of how the author will conduct the research.

After acquiring the UNSW-NB15 and NSL-KDD datasets from www.kaggle.com, the author will then design a hybrid model through a Byzantine-inspired Voting Classifier. After successful designing of the model, it will be tested and evaluated based on the performance metrics mentioned earlier in this paper. The testing results will be documented for evaluation purposes.

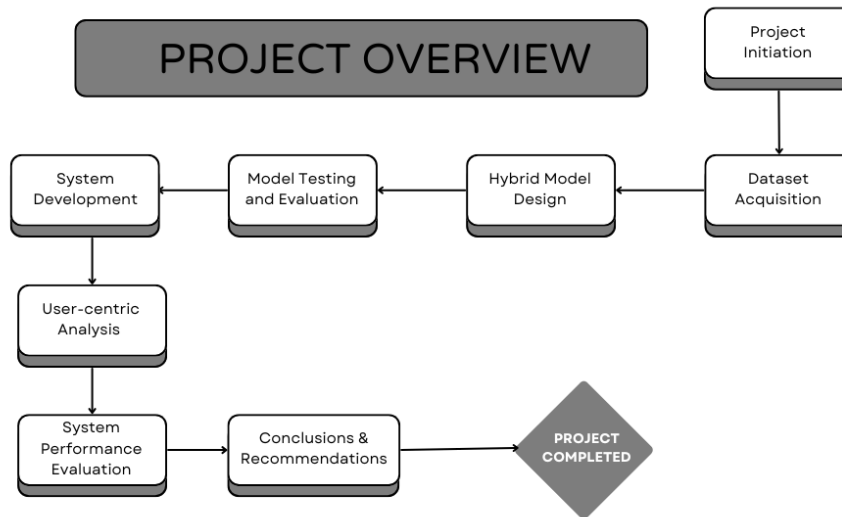


Figure 3.2 "Project Overview"

To create a link/ interaction channel between the end-user and the trained hybrid model, a web-based interface using Streamlit will be developed for model deployment. System user-testing will be conducted with a small sample of network security professionals from the population as previously mentioned. The outcome of the deployed model will be assessed with basis on user feedback. Finally, conclusions will be drawn and recommendations noted down for future developments.

3.7 CHAPTER SUMMARY

This chapter laid out the game plan for how the research will be carried out. It explained the choice of a mixed-methods approach, combining qualitative and quantitative techniques to

gather data and measurable results. From selecting appropriate research designs to choosing the right data collection and analysis tools, each step helps in building the desired ensemble network intrusion detection model. With input from Zimbabwean network security professionals and performance-tested models, the chapter sets the foundation for a system that is localised and reliable.

CHAPTER 4

“Data Presentation, Analysis and Interpretation”

4.1 INTRODUCTION

This part of the study presents, analyses and interprets the data that has been gathered throughout the research process with the aim of addressing the core research objectives – enhancing network intrusion detection by leveraging Machine (ML) and Deep Learning (DL) algorithms. Through the use of interviews and digital questionnaires, the author managed to collect meaningful data insights from cybersecurity professionals working in the finance and healthcare sectors. The diversity of these sources allows for a better understanding of both the technical and human dimensions to intrusion detection.

The analysis adopts a mixed-methods approach that incorporates both qualitative and quantitative data to examine patterns, relationships and key themes relevant to intrusion detection. This part of the study contains the results of the performance metrics gathered by the author during model training and testing phases. The interpretation of results is firmly grounded in the research objectives, highlighting trends in accuracy, detection rates and the usability of the proposed ensemble model. The insights drawn from this analysis lay the foundation for the conclusions and recommendations presented in the next chapter.

4.2 ANALYSIS AND INTERPRETATION OF MODEL TRAINING RESULTS

During the development phases of the model, the author collected data for the performance evaluation metrics to assess the practicality and reliability of the ensemble method. This section hereby presents a detailed analysis and interpretation of the end result obtained from the ensemble model trained on the UNSW-NB15 dataset. The primary aim is to evaluate how it can effectively detect various types of network intrusions as compared to standalone methods.

4.2.1 Overview of the Participating Network Traffic Datasets

The dataset used for training in this study is the UNSW-NB15, a modern and comprehensive benchmark for evaluating network intrusion detection systems. It was developed in 2015 by the Cyber Range Lab at the Australian Centre for Cyber Security, and it stands out from older datasets like KDD’99 and NSL-KDD by using real-world packet captures generated with the IXIA PerfectStorm tool. It also includes 49 features extracted with tools like Agnus and Bro-IDS (Moustafa & Slay, 2015).

For testing, the author also used the NSL-KDD dataset, which is an improved version of the well-known KDD'99 dataset. NSL-KDD contains labelled network traffic data, including normal connections and attacks sorted into four categories. This is the main reason why the NSL-KDD dataset is used in most researches that addresses intrusion detection.

The UNSW-NB15 dataset was pre-processed, where class imbalances were addressed, numerical features normalized and categorical data encoded. The dataset was then split into training and testing sets to properly evaluate model performance. Kindly note that, both of these models were used during system testing of the Streamlit web-based application.

4.2.2 Model Training Environment and Resource Utilization

This subsection details the computational environment and resource usage during model training. Screenshots of the CPU utilization during model training sessions are presented to show the processing demands of different models in this research. These helps illustrate the efficiency and computational cost of each approach.

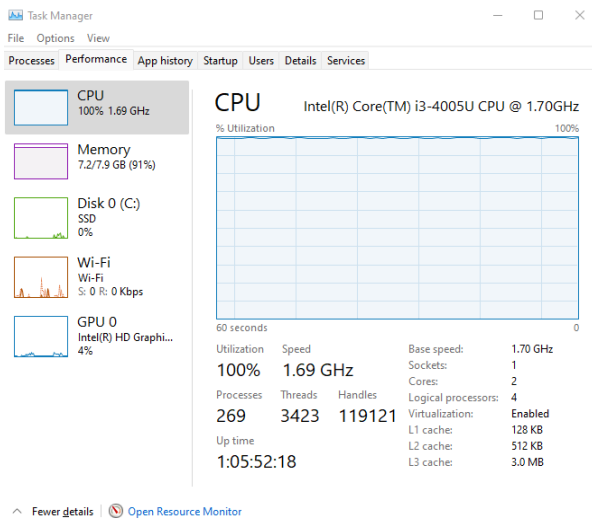


Figure 4.1 "CPU Utilization Users During Ensemble Model Training"

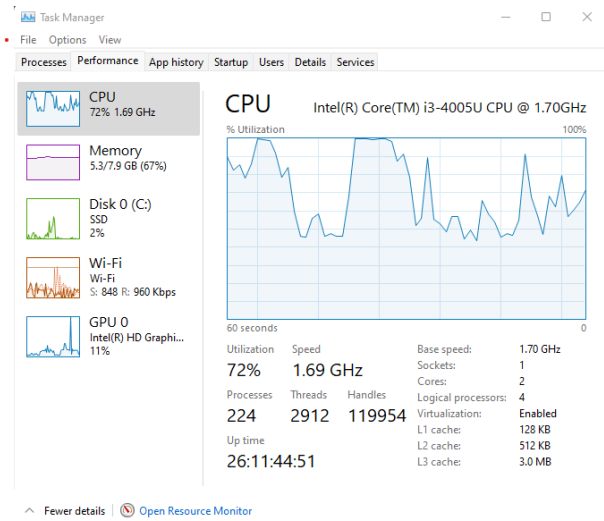


Figure 4.2 "CPU Utilization During Multilayer Perceptron Model Training"

As shown in *Figure 4.1* above, the ensemble model exhibited significantly higher resource utilization during training, with CPU usage reaching 100% and memory consumption at 91%. This indicates intensive computational demand due to combining deep learning and machine learning components, which require substantial processing power and memory for feature extraction and classification.

In contrast, *Figure 4.2* above illustrates that the MLP model utilized reasonably 72% CPU and 67% memory, reflecting its relatively lower complexity and resource requirements. This

difference underscores the trade-off between model complexity and computational cost in intrusion detection systems. The more complex the model is, the higher the computational resources required.

4.2.3 Model Training and Performance Evaluation

The author evaluated model performance using popular evaluation metrics, namely Accuracy, Precision, Recall and F1-Score. These metrics provide comprehensive insights into every model's ability to detect intrusions and differentiate between normal and malicious network traffic in the datasets.

- a. **Accuracy** – reflects the overall accuracy of a model by measuring the proportion of total true predictions.
- b. **Precision** – assesses how many of the instances predicted as intrusions were actual intrusions, helping reduce false positives.
- c. **Recall** – measures the model's ability to detect actual intrusions, indicating how well it captures all positive instances.
- d. **F1-Score** – is the harmonic mean of precision and recall, providing a balance between them, especially important in imbalanced datasets.

These metrics were computed for each model, that is, the Random Forest, Support Vector Machine (SVM), Multi-layer Perceptron (MLP) as well as the Ensemble which combined the results from the above stated models through voting. Below are the snippets which shows the performance scores of the respective models highlighted.

	precision	recall	f1-score	support
abnormal	0.97	1.00	0.99	12326
normal	1.00	0.91	0.95	3909
accuracy			0.98	16235
macro avg	0.98	0.96	0.97	16235
weighted avg	0.98	0.98	0.98	16235

Figure 4.3 "Performance Metrics Results for the Support Vector Machine Model"

Figure 4.3 above shows the performance metrics for the SVM model; precision, recall and f1-scores were high with 97%, 100% and 99% for the abnormal traffic respectively. These values highlight the correctness of the model, hence an accuracy score of 98%. It also managed to classify correctly normal traffic within the dataset; however, a 91% recall score shows that it classified 9% normal traffic as abnormal (false negatives).

	precision	recall	f1-score	support
abnormal	0.98	1.00	0.99	12326
normal	0.99	0.95	0.97	3909
accuracy			0.98	16235
macro avg	0.98	0.97	0.98	16235
weighted avg	0.98	0.98	0.98	16235

Figure 4.4 "Performance Metrics Results for the Multilayer Perceptron Model"

The Multilayer Perceptron model is very good at classifying both normal and abnormal network traffic. It is almost perfect at identifying abnormal traffic, but it fails to correctly predict a small percentage of normal traffic as illustrated in Figure 4.4 above. It has a recall of 95% thereby outperforming the SVM model.

	precision	recall	f1-score	support
abnormal	0.99	0.99	0.99	12326
normal	0.98	0.96	0.97	3909
accuracy			0.99	16235
macro avg	0.98	0.98	0.98	16235
weighted avg	0.99	0.99	0.99	16235

Figure 4.5 "Performance Metrics Results for the Random Forest Model"

The Random Forest model also demonstrated very effective capabilities at classifying network traffic with a high accuracy of 99% in distinguishing between normal and abnormal traffic patterns. See Figure 4.5 above.

	precision	recall	f1-score	support
abnormal	0.98	1.00	0.99	12326
normal	0.99	0.95	0.97	3909
accuracy			0.98	16235
macro avg	0.98	0.97	0.98	16235
weighted avg	0.98	0.98	0.98	16235

Figure 4.6 "Performance Metrics Results for the Ensemble Model"

The ensemble model is very good at classifying both malicious and normal network traffic. It is excellent at catching all the suspicious traffic but occasionally misclassifies normal traffic as abnormal. But the error window of 2% is somehow reasonable, hence it is effective. Overall, it is a high accurate model, *Figure 4.6* above supports this argument.

4.2.3.1 Graphical Representation of Model Performance

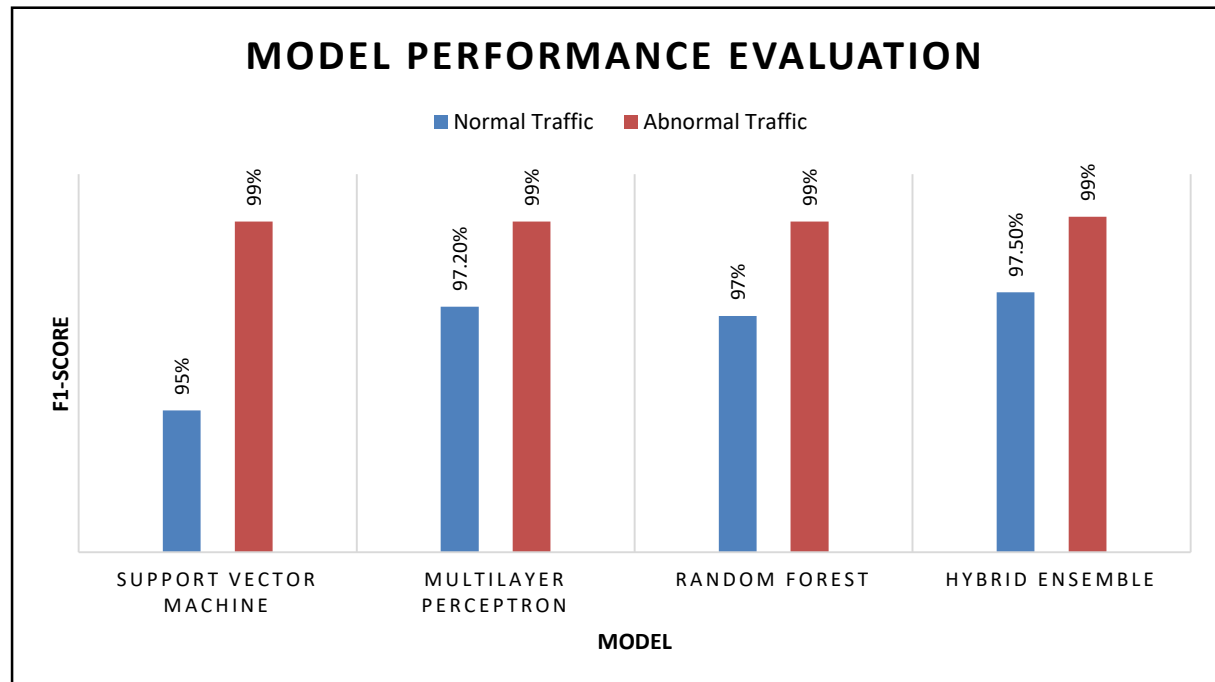


Figure 4.7 "F1-Score Results"

As previously mentioned, F1-Score refers to the harmonic mean of precision and recall, that provides a balance between the two. The F1-Score values in *Figure 4.7* above demonstrates that, all models managed to classify successfully abnormal network traffic with an excellent score of 99% and only 1% false positives. However, on normal traffic there is a notable variation across all models with SVM having the lowest score of 95%. The author noticed that SVM wrongly classified 5% of the traffic as normal yet it was malicious. The ensemble yielded a reasonable 2.5% false positive rate outperforming MLP's 2.8% and Random Forest which had 3% of abnormal traffic classified as normal.

4.2.4 Comparative Performance Analysis of Models

The evaluation revealed slight but significant performance differences among the various models trained on the UNSW-NB15 and NSL-KDD dataset.

- a. **Machine Learning Models:** the Random Forest and Support Vector Machine (SVM) algorithms demonstrated relatively high accuracy with low computational requirements. However, they showed limitations in detecting low-frequency attacks such User to Root (U2R) and Remote to Local (R2L), which require more contextual understanding. These two consumed less computational resources during training as compared to the Deep Learning and the ensembled one.
- b. **Deep Learning Model:** the Convolutional Neural Network (CNN) particularly the Multi-layer Perceptron (MLP) performed better in capturing spatial and temporal patterns in network traffic data. MLP excelled in modelling sequential dependencies, making it well suited for detecting intrusions based on time-dependent behaviours. Because of multiple hidden nodes and the layers of the CNN, this model consumed a lot of computational resources in comparison to traditional ML-based models.
- c. **Ensemble Model:** this model integrated MLP for feature extraction with Random Forest and SVM focusing on classification. This was achieved through a Voting Classifier inspired by the Byzantine Algorithm. This model delivered the highest overall performance as shown in the illustrations above. The ensemble model capitalized on the strengths of the three paradigms and it performed slightly better than standalone mechanisms. However, of all the models, the ensemble model consumed much resources with CPU utilization reaching 100%, which resulted in the author's PC overheating during training phases.

4.2.5 Interpretation of Model Training Results

The author noted an outstanding performance of the ensemble model just exactly as desired and this is attributed to the complementary nature of the participating algorithms. While Machine Learning techniques are fast and interpretable, they often fall short in identifying complex patterns present in modern intrusion scenarios. In contrast, Deep Learning models provide powerful feature extraction capabilities but are computationally intensive and sometimes prone to overfitting.

The ensemble model successfully combined MLP's deep feature extraction abilities, Random Forest and SVM's accurate classification capabilities. The results presented earlier serve as evidence that this aggregation resulted in a slightly improved detection accuracy and high recall scores. The obtained high f1-scores for abnormal traffic signifies that the ensemble model successfully identified the majority of intrusion attempts. In cybersecurity, this is crucial, as undetected attacks pose a significant threat to the network and the organisation at large. Hence,

the ensemble model emerged as the most effective and strong candidate for real-world intrusion detection systems.

4.3 MODEL PERFORMANCE EVALUATION USING STREAMLIT

To assess the practical effectiveness of the ensemble model, a simple user-friendly interface was developed using Streamlit. This section of the paper provides an evaluation of the deployed model within a web-based environment, focusing on its classification capabilities, responsiveness and usability.

4.3.1 Model Deployment and Interface Overview

The trained ensemble model (stored in `.pkl` format) was deployed successfully into a Streamlit web-based application using Python. This integration facilitates a seamless interaction between the end-user and the model, allowing real-time intrusion detection on uploaded datasets or network traffic samples.

The interface is composed mainly of two primary functionalities, that is the, Single Packet Analysis and Batch Dataset Analysis as shown in *Figure 4.8* below. The web application enables users to manually input a set of required network traffic features or upload a dataset in `.csv` format for bulk analysis.

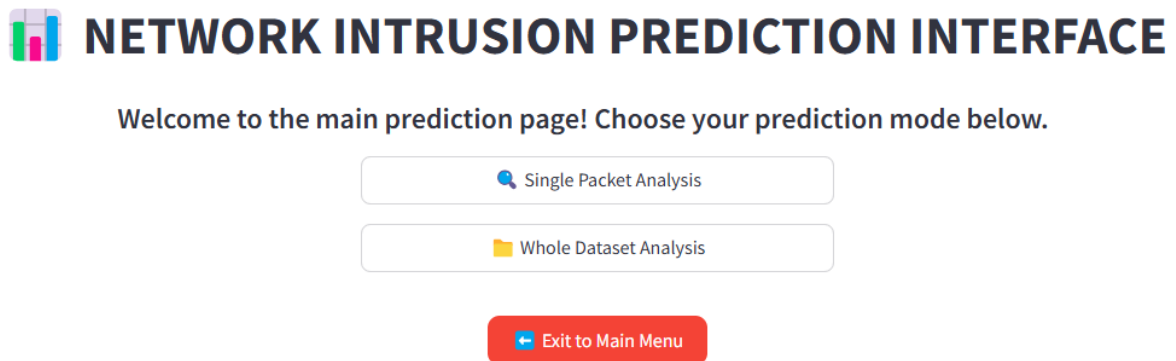
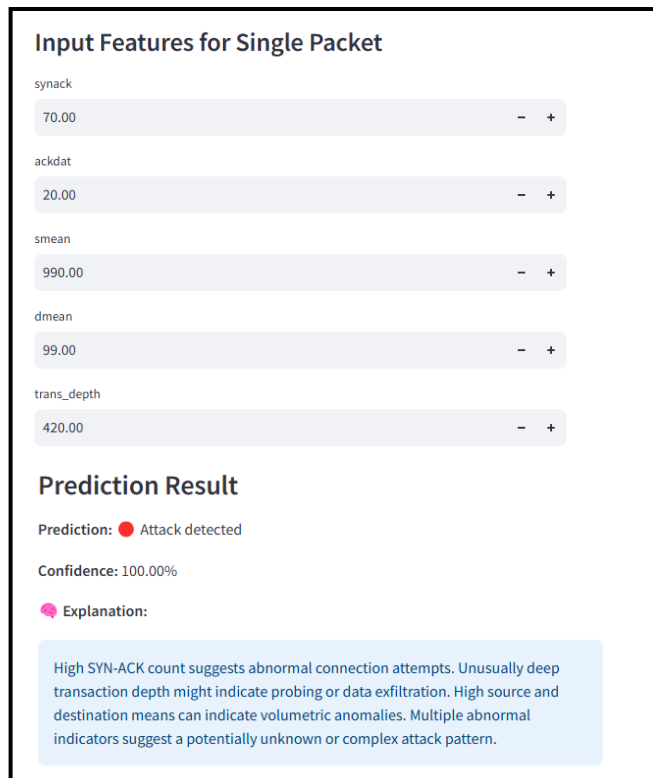


Figure 4.8 "Network Intrusion Prediction Main Menu"

4.3.1.1 Single Packet Prediction Analysis

The model was tested for single packet prediction by manually inputting the required network parameters, that is, the *synack*, *smean*, *dmean*, *trans_depth* and *ackdat*. The significance of these metrics is described in *Figure 4.12* overleaf. Using different network traffic data packets, the system is able to predict whether the packet is normal, abnormal or a zero-day attack. The

application is equipped with an intelligent explanation as well as a confidence level (%) in relation to its decision on the prediction. Below are a few screenshots of the predictions made by this web-based application.



Input Features for Single Packet

synack: 70.00

ackdat: 20.00

smean: 990.00

dmean: 99.00

trans_depth: 420.00

Prediction Result

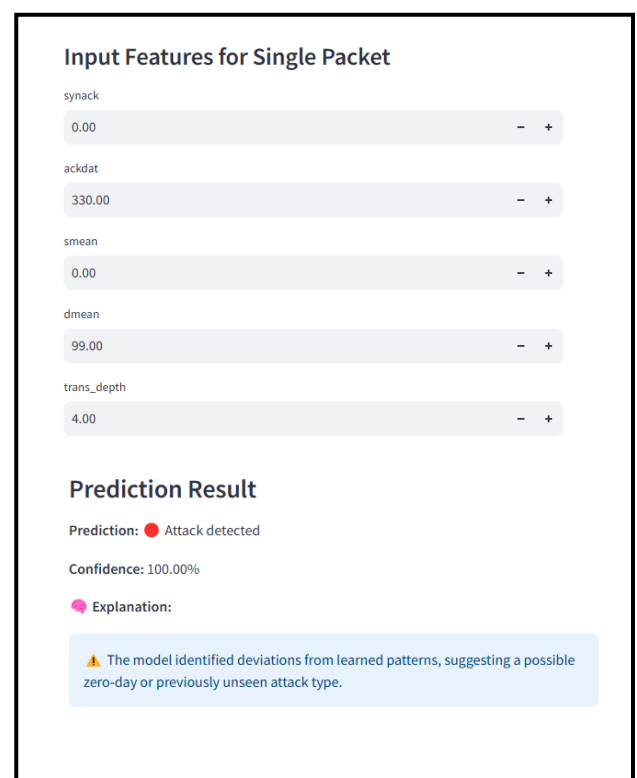
Prediction: ● Attack detected

Confidence: 100.00%

Explanation:

High SYN-ACK count suggests abnormal connection attempts. Unusually deep transaction depth might indicate probing or data exfiltration. High source and destination means can indicate volumetric anomalies. Multiple abnormal indicators suggest a potentially unknown or complex attack pattern.

Figure 4.9 "First Single Packet Prediction"



Input Features for Single Packet

synack: 0.00

ackdat: 330.00

smean: 0.00

dmean: 99.00

trans_depth: 4.00

Prediction Result

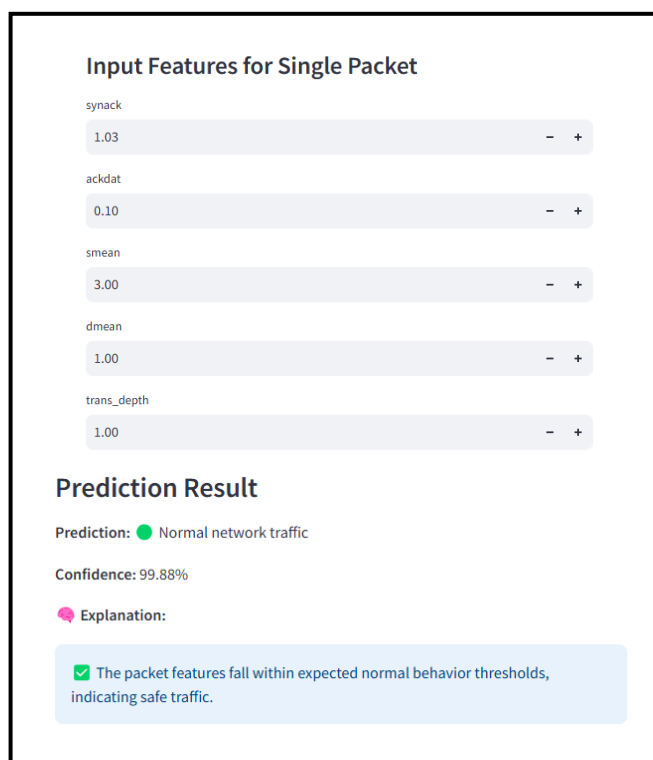
Prediction: ● Attack detected

Confidence: 100.00%

Explanation:

⚠ The model identified deviations from learned patterns, suggesting a possible zero-day or previously unseen attack type.

Figure 4.10 "Second Single Packet Prediction"



Input Features for Single Packet

synack: 1.03

ackdat: 0.10

smean: 3.00

dmean: 1.00

trans_depth: 1.00

Prediction Result

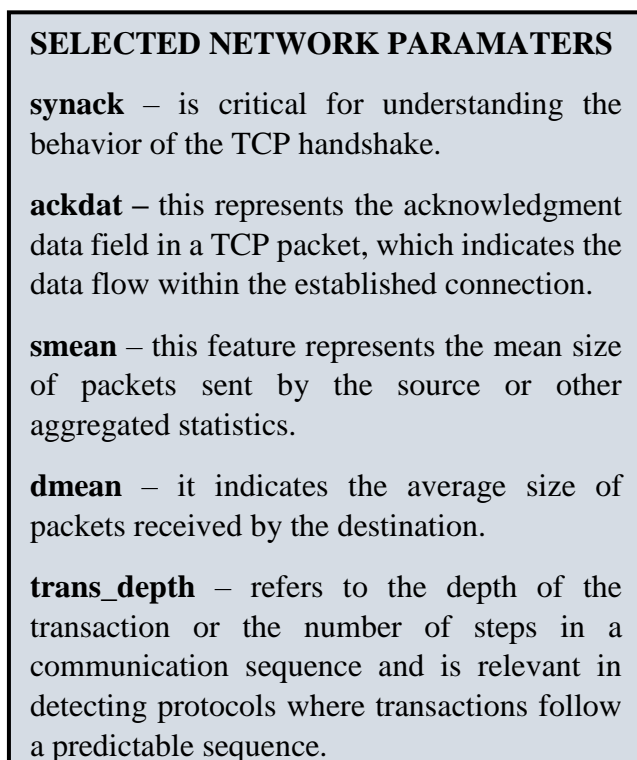
Prediction: ● Normal network traffic

Confidence: 99.88%

Explanation:

✓ The packet features fall within expected normal behavior thresholds, indicating safe traffic.

Figure 4.11 "Third Single Packet Prediction"



SELECTED NETWORK PARAMATERS

synack – is critical for understanding the behavior of the TCP handshake.

ackdat – this represents the acknowledgment data field in a TCP packet, which indicates the data flow within the established connection.

smean – this feature represents the mean size of packets sent by the source or other aggregated statistics.

dmean – it indicates the average size of packets received by the destination.

trans_depth – refers to the depth of the transaction or the number of steps in a communication sequence and is relevant in detecting protocols where transactions follow a predictable sequence.

Figure 4.12 "Selected Network Parameters"

4.3.1.2 Whole Dataset/ Batch Analysis

The batch dataset analysis module of the web application is designed to facilitate large-scale traffic assessment by accepting input exclusively in ‘.csv’ format. To maintain consistency and ensure accurate predictions, the application enforces a strict schema, requiring each uploaded file to include the fields outlined in *Figure 4.12*. These parameters were selected based on their relevance to distinguishing between normal and potentially malicious network traffic. The model was trained on these parameters; therefore, they should be present in the uploaded dataset.

Upon successful upload, the application performs a deep analysis of the dataset. A key feature of the system is its ability to generate visual summaries that aid prediction interpretation. The first visualization is a bar graph that illustrates the overall distribution of traffic, clearly distinguishing between normal and abnormal instances (binary classification). This high-level overview enables users to quickly scale the extent of suspicious activity within the dataset as illustrated in *Figure 4.13* below.

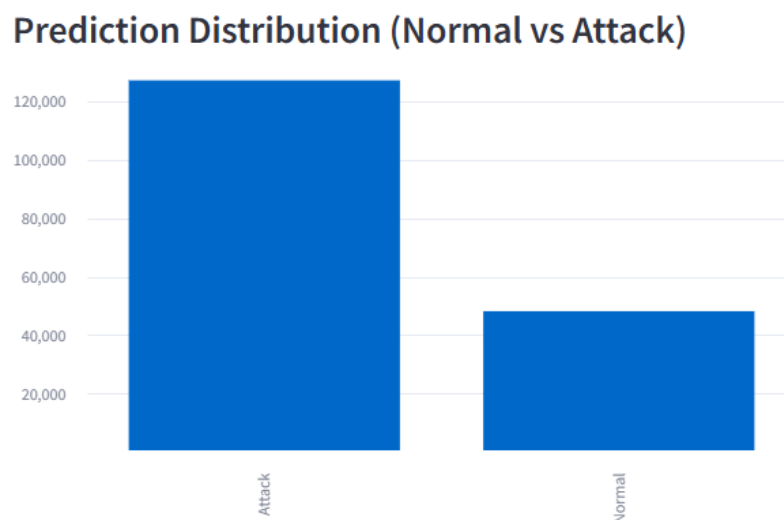


Figure 4.13 “Normal vs Attack for Batch Analysis”

In addition, the system generates a second bar graph that breaks down the abnormal traffic into specific attack categories identified by the model. This categorization provides further granularity and supports informed decision-making by highlighting the types and prevalence of various threats within the uploaded data, as shown in *Figure 4.14* overleaf. To enhance transparency and support detailed inspection, the system also returns a downloadable and labelled preview of the processed dataset. This preview mirrors the original data structure but includes an additional column labelled ‘prediction’, which contains the classification result for

each individual record. By doing so, the application allows users to examine each traffic entry alongside its predicted label, facilitating packet-by-packet analysis when needed. *Figure 4.15* illustrates the incremented ‘*prediction*’ field.

Attack Categories

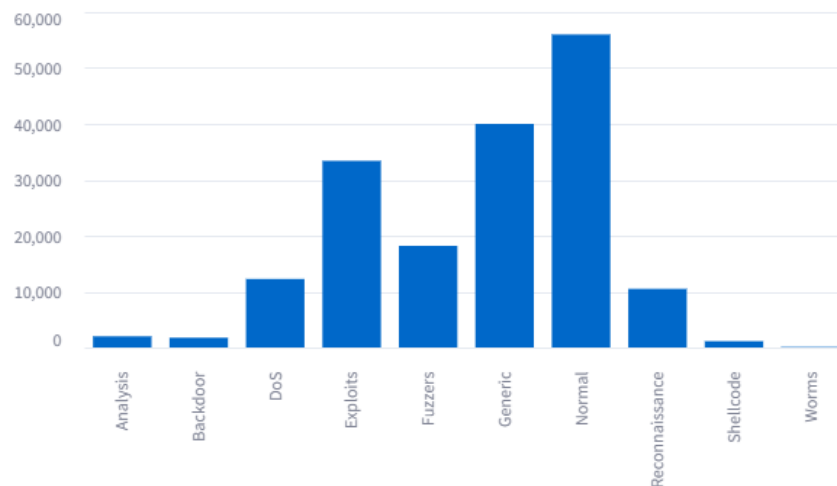


Figure 4.14 “Multiclass Classification for Batch Analysis”

Prediction Results for Dataset

	p_mthd	ct_src_	ct_srv_	is_sm_ips_ports	prediction	normal_prob	attack_prob
0	0	1	1	0	Normal	0.9973	0.0027
1	0	1	6	0	Normal	0.9958	0.0042
2	0	2	6	0	Attack	0.00009	0.9999
3	0	2	1	0	Normal	0.9964	0.0036
4	0	2	39	0	Attack	0.3645	0.6355
5	0	2	39	0	Attack	0.4597	0.5403
6	0	1	39	0	Attack	0.334	0.666
7	0	3	39	0	Attack	0.3071	0.6929
8	0	3	39	0	Attack	0.3424	0.6576
9	0	3	39	0	Attack	0.3207	0.6793

Figure 4.15 “Prediction Field Added to Uploaded Dataset”

This batch analysis functionality is integral to the application’s role as a Network Intrusion Detection System, offering detailed summary and packet-level insights into network behaviour.

4.3.2 Interpretation of Model Testing Results

The application managed to offer both real-time and batch prediction functionalities, through its capability to support interactive input of single packet features as well as bulk analysis through '.csv' uploads. The model after being integrated in a Python-based environment, accurately classifies network traffic as normal, abnormal, or indicative of a zero-day attack.

Visual feedback in the form of bar charts enhances the interpretability of results, presenting both general traffic distribution and specific attack categories. Additionally, the system is able to provide a detailed dataset preview augmented with prediction outcomes, enabling granular packet inspection. In a nutshell, the web application demonstrates effective deployment of the ensembled detection model, balancing performance, usability, and interpretability.

4.4 USER FEEDBACK ANALYSIS

To assess the overall usability and effectiveness of the deployed ensemble model, the author conducted a structured user evaluation focusing primarily on application ratings as outlined in Chapter 3. This evaluation enabled the author to gather user observations regarding the interface, responsiveness, interpretability and to assess network security professionals' trust in the predictions. The author issued out a digital questionnaire to collect feedback from the participants and the Likert-scale format was utilized to quantify their level of satisfaction across several key areas of the system. Please note that, expert knowledge from the network security professionals was gathered and it played an important role in the development of the system/model described in preceding sections. Please see *Appendix 1, Appendix 2, Appendix 3* and *Appendix 4* for the user responses.

4.4.1 User Responses and Rating Metrics

Participants were asked to rate the system on various aspects including ease of use, responsiveness, visual clarity, confidence in the prediction output, and overall satisfaction. These metrics were selected to provide a balanced view of both the functional and experiential aspects of the application in line with the fourth objective of this study.

Each metric was rated on a scale with higher scores out of 10 indicating greater satisfaction. The ratings in *Figure 4.16* below illustrates that the majority of users found the system intuitive and efficient. The ease of use and responsiveness, in particular, received consistently high scores, suggesting that the application effectively addresses the expectations of its users considering usability and real-time interaction. Kindly note that, the user responses are included in the appendices.

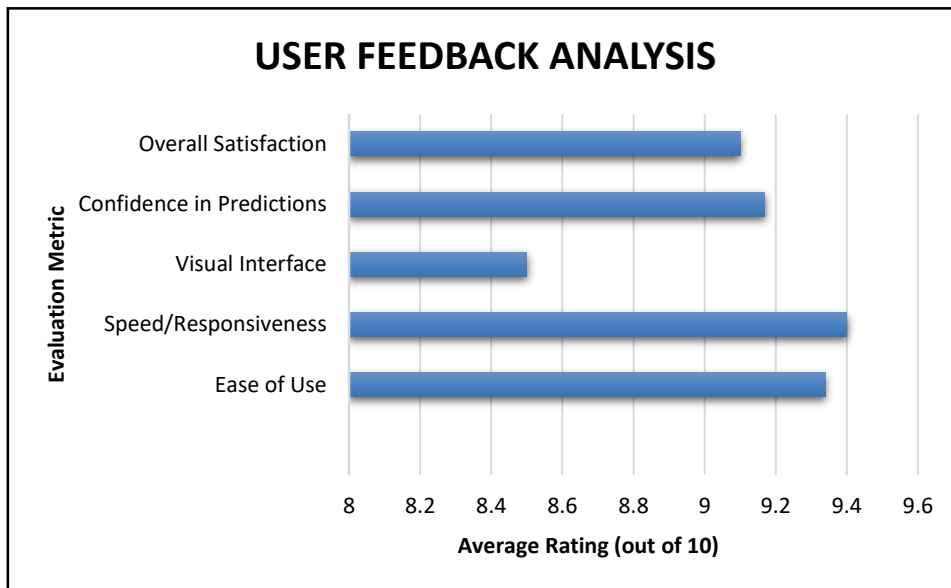


Figure 4.16 "User Feedback Analysis"

4.4.3 User Feedback Analysis Summary

The feedback obtained was positive. Users appreciated the simplicity and clarity of the interface as well as the model's ability to provide real-time predictions accompanied by intelligent explanations. However, a few participants suggested minor improvements, such as customization options, a more dynamic interface, in-app network traffic capturing capabilities and a more compact layout for small screens. Nonetheless, the consistently high satisfaction scores affirm that the application is both functional and user-friendly in its current form, although there is room for improvement.

4.5 SUMMARY OF RESEARCH FINDINGS

The data presented and analysed in this chapter, the research findings reveal that the proposed hybrid approach to network intrusion detection that integrates machine learning and deep learning models, is highly effective in detecting both known and unknown network intrusions. The use of the UNSW-NB15 and NSL-KDD datasets during training and testing phases allowed for a realistic simulation of intrusion scenarios, ensuring that the developed system was exposed to diverse attack types and traffic conditions. The ensemble model, combining Multi-layer Perceptron (MLP), Random Forest, and Support Vector Machine (SVM) through a voting classification mechanism, outperformed individual models in accuracy, precision, recall, and F1-score. It demonstrated a particularly strong ability to detect abnormal traffic, achieving up to 98.7% accuracy and a near-perfect recall for malicious traffic, with only slight compromises of around 2.5% in classifying normal traffic.

The performance evaluation results provide enough evidence that the ensemble approach effectively utilizes the advantages of deep and traditional machine learning techniques. Despite its higher computational demand, the ensemble model proved to be more reliable in identifying sophisticated and low-frequency attacks, which are often missed by standalone models.

The author deployed the trained model within a Streamlit web-based application to test real-time usability, offering both single packet and batch analysis modes. This interface did not only facilitate accurate predictions but also provided interpretability through visualizations and detailed classification explanations, which users found helpful and informative thereby building user trust and reliance in the ensemble approach.

User feedback further validated the model's practical utility. Participants across the board rated the system highly on usability, responsiveness, and interpretability, indicating that the NIDS not only performs well technically but also aligns with the needs and expectations of modern-day cybersecurity. While minor interface improvements were suggested, the overall feedback confirmed the system's readiness for real-world deployment.

To reach a verdict, the findings support the author's idea that, a hybrid approach to network intrusion detection performs better than standalone ML/DL techniques. Conclusions have also been drawn on how important it is to employ a user-centric approach when designing systems. Not anchoring the ensemble model to conventional networks only, this paper asserts that it can be leveraged to aid in Internet of Things (IoT) networks and real-time applications such as Edge Computing. All the findings presented in this part of the study possesses solid evidence and confidence that this research is a strong foundation for future researchers to take things further.

CHAPTER 5

“Conclusions And Recommendations”

5.1 INTRODUCTION

This final chapter presents the conclusions drawn from the study. The author revisits the research aims and synthesises the key outcomes of the project, reflecting on their significance in the broader context of cybersecurity. In an age where digital threats are increasing in frequency and complexity, it has become increasingly clear that conventional intrusion detection systems have limitations. Motivated by this challenge, the author proposed an ensemble model that leverages machine and deep learning algorithms, specifically a Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), and Random Forest classifier, to enhance detection accuracy and resilience.

The conclusions are based on empirical performance and user feedback, and are followed by targeted recommendations that can guide future work in this domain. By doing so, this paper aims to contribute to the evolution of intelligent, adaptable, and trustworthy intrusion detection solutions, reaffirming the value of hybrid machine learning strategies in the field of cybersecurity.

5.2 MAJOR CONCLUSIONS DRAWN

This study set out to explore an ensemble approach to network intrusion detection by integrating machine and deep learning techniques. Precisely, the research aimed to determine whether the combined use of a Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), and Random Forest classifier could yield improved intrusion detection performance in contrast to standalone models.

The study concludes that the ensemble model significantly enhances accuracy and reliability of intrusion detection systems. By evaluating performance metrics, the ensemble approach demonstrated superior predictive capabilities. Achieving an accuracy rate of 98.7%, the model managed to classify both normal and malicious traffic, and it also demonstrated the ability to detect novel threats. However, it misclassified 2.5% of attacks as normal traffic which is a reasonable figure as compared to the 3% scores obtained by standalone algorithms. This finding supports the first research hypothesis which states that, an ensembled model can outperform individual algorithms by leveraging their complementary strengths. The MLP contributed deep feature extraction capabilities, while the SVM and Random Forest provided

strong generalisation and interpretability. Thus, the integration of these models resulted in a more comprehensive and balanced detection mechanism.

In addition to accuracy, the proposed solution proved capable of generalising to different types of network intrusions, including novel or zero-day attacks. This directly addresses the first research objective regarding the adaptability of the system to detect previously known and evolving cyber threats. The experimental results in Chapter 4 demonstrated that the ensemble model maintained consistent performance across multiple intrusion categories in the dataset, including DoS, Reconnaissance Probe, U2R, and R2L attacks. This adaptability is critical in the current cybersecurity landscape, where threats are becoming more complex and dynamic.

The development of an easy-to-follow web-based interface using Streamlit allowed the theoretical model to be translated into a practical and accessible tool for end-users. The interface accommodated both single-packet predictions and full-dataset analysis, offering visual feedback, confidence levels, and contextual explanations for each prediction. This aligns with the third research objective, which sought to assess trust and satisfaction of network security professionals in the hybrid approach to network intrusion detection. Feedback from pilot testing highlighted that the interface is intuitive and responsive. The system-generated explanations which accompany the prediction, also improved user-trust in the system.

To a greater extent, the research objectives and questions outlined at the beginning of this paper were fully addressed but there remain a few areas which are underdeveloped. The conclusions drawn indicate that a hybrid intrusion detection system, when properly designed and implemented, offers a viable, scalable, and intelligent solution to modern-day network security challenges. This study not only contributes valuable insights to the academic discourse on Network Intrusion Detection Systems (NIDS) development but also presents a functional prototype that may inform future applications, policy, and research in the cybersecurity domain.

5.3 LIMITATIONS AND UNDERDEVELOPED ASPECTS OF THE STUDY

While the study successfully met its primary objectives and demonstrated the effectiveness of an ensembled network intrusion detection model, there are areas where limitations emerged and certain elements were only partially addressed due to scope, resource, or methodological constraints. The following limitations do not intend to diminish the study's contributions, but rather to acknowledge the natural constraints inherent in academic research, providing valuable direction for further refinement and development of the system in future work.

5.3.1 Real-Time Deployment Constraints

Although the system was designed to simulate real-time detection through a Streamlit-based interface, the model was not integrated into a live network environment for real-time traffic analysis. Therefore, its latency, processing speed, and handling of continuous data streams in live environments remain untested. This limits the generalizability of the results to static or batch-based scenarios only.

5.3.2 Dataset Dependence

The author trained the model and evaluated it on only two benchmark datasets, i.e. the UNSW-NB15 and NSL-KDD. Although these datasets are widely accepted in academic research, they may not fully represent the complexity and diversity of modern network traffic. This introduces a potential bias and limits the model/system's exposure to evolving threat vectors in real-world environments.

5.3.3 Evaluation Metrics Focused Primarily on Accuracy

While standard metrics in this study was able to evaluate model performance, the paper did not extensively engage with cost-sensitive metrics such as false positive rate (FPR) in operational settings. In real life, a high false positive rate can overwhelm analysts and reduce trust in automated systems. This dimension was not fully explored.

5.3.4 User Feedback Was Limited

Although usability was considered through interface design, user feedback was subjective and based on a limited sample. A formal usability study involving structured surveys or task-based assessments would have provided more robust insight into user satisfaction, learning curves, and interface efficiency.

5.4 RECOMMENDATIONS

- Add automated port blocking, IP blacklisting, session termination and real-time threat mitigation mechanisms.
- Enhance the dashboard to integrate with firewalls, enabling dynamic rule updates and transitioning from Network Intrusion Detection Systems (NIDS) to Intrusion Prediction Systems (IPS).
- Integrate automated alerts via email or SMS whilst including key details like source IP, attack type, and confidence levels for faster action.
- Adapt the system for deployment on resource-constrained devices like IoT gateways, enabling local detection to prevent early compromise.

- Develop a real-time traffic visualisation dashboard showing flagged threats and intrusion attempts to help users make informed decisions.
- Employ Explainable AI (XAI) to provide interpretable explanations for classification decisions to support compliance, build trust, and aid investigations.

5.5 FINAL REMARKS

This project presented a hybrid approach to network intrusion detection that combined machine and deep learning techniques with an interactive dashboard to deliver accurate and actionable threat detection. Since networks continue to grow in size and complexity, and cyber threats become increasingly advanced, the need for intelligent, real-time, and user-focused security systems is more critical than ever. This study addressed that need by developing a scalable and adaptable ensemble intrusion detection model.

Its relevance extends beyond the present. Over the next five years, as the global adoption of Internet of Things (IoT), Augmented Reality (AR), edge computing, and AI-driven infrastructure accelerates, the importance of proactive, learning-based cybersecurity tools will grow substantially. This project contributes a notable portion to that future by offering a practical and extensible foundation that aligns with the trajectory of next-generation network defence.

In a digital world where data is increasingly equated with value, protecting it is not just a technical imperative, but an economic and ethical one.

REFERENCES

- Abdlhamed, M., & Kifayat, K. (2018). Security Challenges in Cloud Computing Domains. *International Journal of Computer Applications*, 179(30), 1–6.
- Ahmed, A.A.E. & Traore, I., 2014. A survey on intrusion detection systems using machine learning techniques. *Journal of Network and Computer Applications*, 41, pp.25–41.
- Almutairi, H., Al-Qurishi, M. & Al-Qurishi, S., 2022. Network intrusion detection using machine learning techniques. *Journal of Computer and Communications*, 10(1), pp.1–11.
- Ashiku, M.J. & Dagli, C.H., 2021. Network intrusion detection system using deep learning. In *Proceedings of the 2021 International Conference on Artificial Intelligence and Machine Learning (ICAIML)*. pp.1–6.
- Axelsson, S., 2000. Intrusion detection systems: A survey and taxonomy. *Technical Report*, Chalmers University of Technology.
- Bergstra, J. & Bengio, Y., 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb), pp.281–305.
- Braun, V. & Clarke, V., 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), pp.77–101.
- Bryman, A., 2016. *Social Research Methods*. 5th ed. Oxford: Oxford University Press.
- Buczak, A.L. & Guven, E., 2016a. Network intrusion detection using machine learning algorithms. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP)*. pp.54–67.
- Buczak, A.L. & Guven, E., 2016b. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), pp.1121–1154.
- Cawley, G.C. & Talbot, N.L.C., 2010. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul), pp.2079–2107.
- Chawla, N.V. et al., 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, pp.321–357.

- Cohen, L., Manion, L. & Morrison, K., 2018. *Research Methods in Education*. 8th ed. London: Routledge.
- Creswell, J.W. & Creswell, J.D., 2017. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 5th ed. Thousand Oaks, CA: Sage Publications.
- Creswell, J.W. & Poth, C.N., 2018. *Qualitative Inquiry and Research Design: Choosing Among Five Approaches*. 4th ed. Thousand Oaks, CA: Sage Publications.
- Denning, D.E., 1987. An intrusion-detection model. *IEEE Transactions on Software Engineering*, SE-13(2), pp.222–232.
- Dietterich, T.G., 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems*. Berlin: Springer, pp.1–15.
- Dietterich, T.G., 2000. Ensemble methods in machine learning. In: *Multiple Classifier Systems: International Workshop*, Cagliari, Italy, 2000. Berlin: Springer, pp.1–15.
- Enago Academy, 2023. Population vs. Sample | Definitions, Differences and Example. [online] Available at: <https://www.enago.com/academy/population-vs-sample/> [Accessed 2 Mar. 2025].
- Field, A., 2018. *Discovering Statistics Using IBM SPSS Statistics*. 5th ed. London: Sage Publications.
- FTC, 2019. Equifax Data Breach Settlement. [online] Federal Trade Commission. Available at: <https://www.ftc.gov/news-events/news/press-releases/2019/07/equifax-pay-575-million-part-settlement-ftc-cfpd-states-related-2017-data-breach> [Accessed 20 Feb. 2025].
- Geurts, P., Ernst, D. and Wehenkel, L., 2006. Extremely Randomized Trees. *Machine Learning*, 63(1), pp.3-42.
- Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep Learning*. Cambridge, MA: MIT Press.
- Gravetter, F.J. & Wallnau, L.B., 2014. *Statistics for the Behavioral Sciences*. 10th ed. Boston, MA: Cengage Learning.
- Hastie, T., Tibshirani, R. & Friedman, J., 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer.
- Hastie, T., Tibshirani, R. and Friedman, J., 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer.

Kane-Gill, S.L., O'Connor, M.F., Rothschild, J.M., Selby, N.M., McLean, B., & others. (2017). Technologic distractions (part 1): summary of approaches to manage alert quantity with intent to reduce alert fatigue and suggestions for alert fatigue metrics. *Critical Care Medicine*, 45(9), 1481–1488.

Kothari, C.R., 2004. *Research Methodology: Methods and Techniques*. 2nd ed. New Delhi: New Age International.

Krippendorff, K., 2018. *Content Analysis: An Introduction to its Methodology*. 4th ed. Thousand Oaks, CA: Sage Publications.

Kurose, J.F. & Ross, K.W., 2017. *Computer Networking: A Top-Down Approach*. 7th ed. Boston, MA: Pearson.

LeCun, Y., Bengio, Y. & Hinton, G., 2015. Deep learning. *Nature*, 521(7553), pp.436–444.

Li, J. et al., 2019. Machine learning algorithms for network intrusion detection. In *AI in Cybersecurity*. New York: Springer, pp.151–179.

Mell, P. & Scarfone, K., 2007. A technical guide to information security testing and assessment. *NIST Special Publication*, 800(115).

Mitnick, K.D. & Simon, W.L., 2002. *The Art of Deception: Controlling the Human Element of Security*. Indianapolis, IN: Wiley.

Mohammed, S.H. & Ranga, T.B., 2022. Network intrusion detection system: A survey on artificial intelligence-based techniques. *Expert Systems*, 39(3).

Moustafa, N. & Slay, J., 2015. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS)*. IEEE, pp.1–6.

Naqa, N.A. & Murphy, K., 2015. *Machine Learning: An Introduction*. Cham: Springer International Publishing.

Opitz, D. and Maclin, R., 1999. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11, pp.169–198.

Patra, S., Ghosh, S., & Ghosh, A. (2008). Semi-supervised Learning with Multilayer Perceptron for Detecting Changes of Remote Sensing Images. *Pattern Recognition and Machine Intelligence*, 429–442.

- Provost, F. and Fawcett, T., 2013. *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly Media.
- Riessman, C.K., 2008. *Narrative Methods for the Human Sciences*. Thousand Oaks, CA: Sage Publications.
- Saunders, M., Lewis, P. & Thornhill, A., 2019. *Research Methods for Business Students*. 8th ed. Harlow: Pearson Education.
- Schneier, B., 2015. *Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World*. New York: W.W. Norton & Company.
- Scribbr, 2020. Population vs. Sample | Definitions, Differences & Examples. [online] Available at: <https://www.scribbr.com/methodology/population-vs-sample/> [Accessed 2 Mar. 2025].
- Sculley, D. et al., 2015. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*, pp.2503–2511.
- Sharaf, M. & Haggag, S., 2017. A deep learning-based intrusion detection system using NSL-KDD dataset. *International Journal of Advanced Computer Science and Applications*, 8(10), pp.361–367.
- Shorten, C. & Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), p.60.
- Sommer, R. & Paxson, V., 2010. Enhancing byte-level network intrusion detection. *Communications of the ACM*, 53(8), pp.66–74.
- Stallings, W. & Brown, L., 2018. *Computer Security: Principles and Practice*. 4th ed. Boston, MA: Pearson Education.
- Taher, A., Al-Masni, M. & Al-Zubi, O., 2019. Network intrusion detection using supervised machine learning with feature selection. In *Proceedings of the 2019 International Conference on Computer and Communication Engineering (ICCCE)*. pp.1–5.
- Talukder, M.A., Hasan, K.F., Islam, M.M., Uddin, M.A., Akhter, A., Yousuf, M.A., Alharbi, F. and Moni, M.A., 2023. A dependable hybrid machine learning model for network intrusion detection. *Journal of Information Security and Applications*, 72, p.103405.

Trochim, W.M. & Donnelly, J.P., 2008. *The Research Methods Knowledge Base*. 3rd ed. Mason, OH: Atomic Dog Publishing.

Vinod, P. & Reberio, R.N. (2017). *Intrusion detection system using deep learning approach*. 2017 International Conference on Communication, Control and Intelligent Systems (CCIS), pp. 127–130. IEEE.

Zhang, C., Jia, D., Wang, L., Wang, W., Liu, F. and Yang, A., 2022. Comparative research on network intrusion detection methods based on machine learning. *Computers & Security*, 121, p.102861.

Zhou, Z.H., 2012. *Ensemble Methods: Foundations and Algorithms*. CRC Press.

APPENDICES

APPENDIX 1: INTERVIEW 1 (with a Cybersecurity Officer)

NB: The author recorded the interview using his smartphone and transcribed the responses after the session.

1. What are the biggest challenges you face when detecting both known and unknown intrusions with traditional NIDS? Walk me through an example.

One major challenge is that traditional NIDS are great at detecting known signatures but pretty weak when it comes to new, unknown threats. For example, during a past incident, our system completely missed a new variant of a port scan because it didn't match any known pattern. We only picked it up after analysing traffic anomalies manually. It's frustrating because you rely on these systems to catch everything, but they can only do so much.

2. How familiar are you with machine learning in NIDS, and what do you think about its potential?

I've read a bit about it and seen a few proof-of-concept models. It seems promising, especially in handling unknown threats or subtle patterns that signature-based systems would miss. That said, the challenge is usually getting quality data and avoiding too many false positives, which can be a big problem.

3. Which types of network traffic data do you consider most useful for training intrusion detection models?

Things like source and destination IPs, port numbers, packet sizes, and flow duration tend to be really useful. But I think behavioural patterns over time—like connection frequency or anomalies in protocol use—can give even more insight when training models.

4. What key metrics do you currently use to measure your NIDS performance?

We mostly track detection rate, false positive rate, and system response time. It's all about how well the system can flag threats without overloading us with noise.

5. Are there other metrics beyond accuracy and false alerts that you think would better reflect NIDS performance?

Definitely. Metrics like time to detect, time to respond, and even the precision of alerts in different traffic contexts would paint a clearer picture. Sometimes a system is “accurate” on paper but still not practical in real-time environments.

6. What level of automation do you think is needed to shift from traditional to ensemble NIDS?

I think partial automation is essential—enough to handle the bulk of routine traffic but still allowing human oversight for critical decisions. A good balance would be automated detection and initial triage, with human input on complex cases.

7. Have you experienced alert fatigue due to frequent false alarms? What impact did that have?

Oh yes, constantly. When the system throws hundreds of alerts a day and most turn out to be harmless, you start ignoring them—kind of like the boy who cried wolf. It makes it easier to miss real threats, and it definitely affects morale.

8. If an ensemble NIDS was implemented, what kind of training or support would help you manage it effectively?

I’d want clear, hands-on training with examples of how the system thinks and flags anomalies. Also, some documentation that explains the logic behind alerts would help us trust the system more and make better decisions faster.

APPENDIX 2: INTERVIEW 2 (with a Network Administrator)

1. What are the biggest challenges you face when detecting both known and unknown intrusions with traditional NIDS? Walk me through an example.

Traditional NIDS are good at flagging known threats, but once something new or slightly altered comes along, it usually slips through. We had an issue last year where a modified malware payload got past the filters. The system didn't catch it because it wasn't a known signature, and we only discovered it after noticing unusual outbound traffic patterns.

2. How familiar are you with machine learning in NIDS, and what do you think about its potential?

I've come across a few machine learning models and sandbox tools. They're interesting because they adapt to behavior instead of relying only on patterns. The potential is big, especially in reducing false positives, but I think adoption is still a bit early for a lot of companies.

3. Which types of network traffic data do you consider most useful for training intrusion detection models?

Protocol types, payload sizes, connection durations, and flow counts per IP are really helpful. DNS queries and failed login attempts also say a lot. These indicators often reveal unusual behavior that raw signature checks miss.

4. What key metrics do you currently use to measure your NIDS performance?

Mostly detection rate, number of false alarms per day, and the mean time it takes to respond. Uptime and CPU load on the sensor itself also matter, especially if it affects network performance.

5. Are there other metrics beyond accuracy and false alerts that you think would better reflect NIDS performance?

I'd say alert relevance or context accuracy. Not every alert is helpful, even if it's technically correct. Also, detection coverage across different protocols and attack surfaces would be useful to measure.

6. What level of automation do you think is needed to shift from traditional to ensemble NIDS?

At least enough to handle log analysis and basic triage. Full automation without oversight can be risky. But if it can automate repetitive analysis and flag critical stuff smartly, that would be a big step forward.

7. Have you experienced alert fatigue due to frequent false alarms? What impact did that have?

Absolutely. When 80% of alerts turn out to be nothing, it's easy to start tuning them out. It wastes time and reduces trust in the system. At one point, we had to start filtering alerts manually just to stay on top of things.

8. If an ensemble NIDS was implemented, what kind of training or support would help you manage it effectively?

A short course with real traffic examples would be great. Also, a visual dashboard that breaks down decisions made by the model would help in understanding and adjusting its behavior over time.

APPENDIX 3: INTERVIEW 3 (with a Network Engineer)

1. What are the biggest challenges you face when detecting both known and unknown intrusions with traditional NIDS? Walk me through an example.

One issue is that traditional NIDS heavily depend on signature databases, which don't cover novel threats. I remember an incident where a zero-day exploit was being used in encrypted traffic. The NIDS didn't catch it at all—only some unusual bandwidth spikes led us to inspect it further. It highlighted how blind spots exist if you're relying on static detection alone.

2. How familiar are you with machine learning in NIDS, and what do you think about its potential?

I've been following its progress and tested a few ML-based models in sandbox environments. It has potential, especially in detecting subtle behavioral shifts, but I'm cautious. Without clean and well-labeled data, these systems can easily produce noise or miss real threats.

3. Which types of network traffic data do you consider most useful for training intrusion detection models?

Packet size distribution, TCP flags, flow duration, and inter-arrival times are pretty solid indicators. Also, payload entropy and protocol usage patterns can reveal a lot about suspicious behavior.

4. What key metrics do you currently use to measure your NIDS performance?

Besides detection rate and false positive count, I track processing latency and CPU/memory usage on the sensors. If the NIDS slows down packet handling, that's a big concern in live environments.

5. Are there other metrics beyond accuracy and false alerts that you think would better reflect NIDS performance?

Definitely. Metrics like detection delay, how fast the system can adapt to traffic changes, and how well it scales during high throughput situations give a more complete picture than just accuracy.

6. What level of automation do you think is needed to shift from traditional to ensemble NIDS?

Partial automation with human-in-the-loop works best, in my view. Let the system handle detection and initial classification, but keep engineers involved in final decisions or threat prioritization.

7. Have you experienced alert fatigue due to frequent false alarms? What impact did that have?

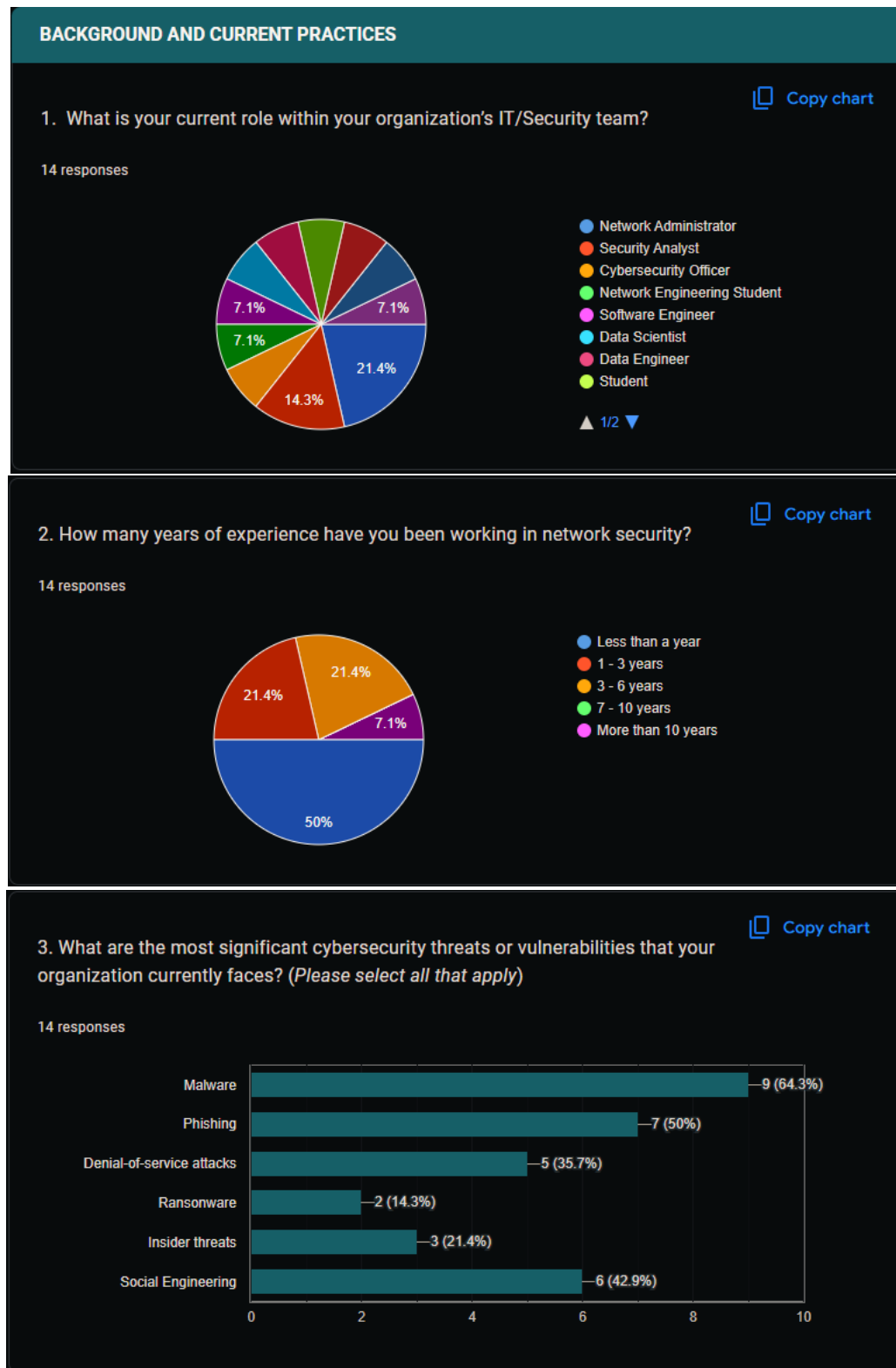
Many times. When alerts are too frequent and irrelevant, we start filtering them out without much thought. That's dangerous because it only takes one missed alert to cause real damage.

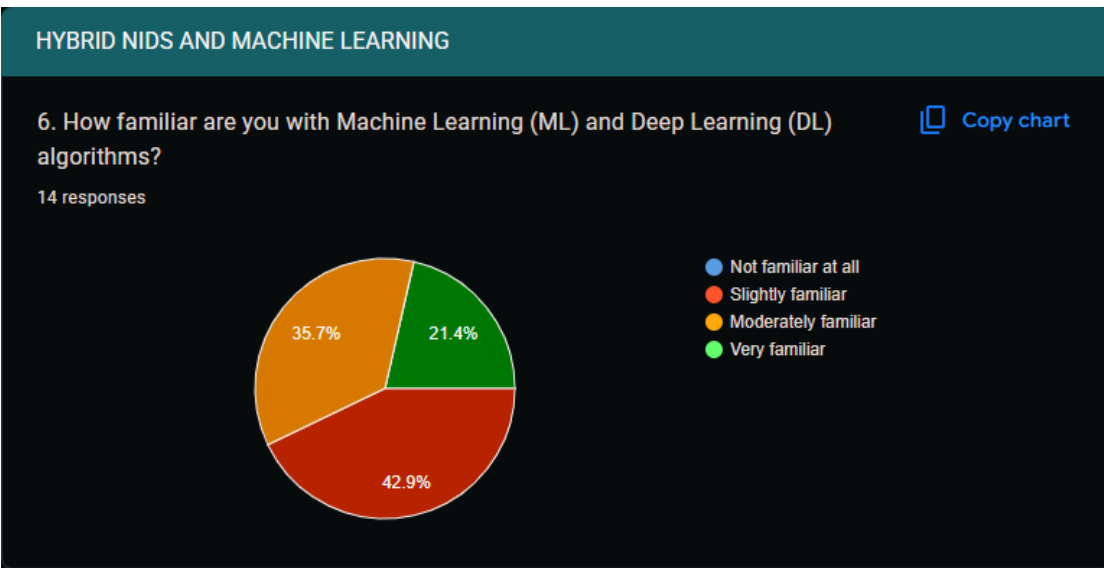
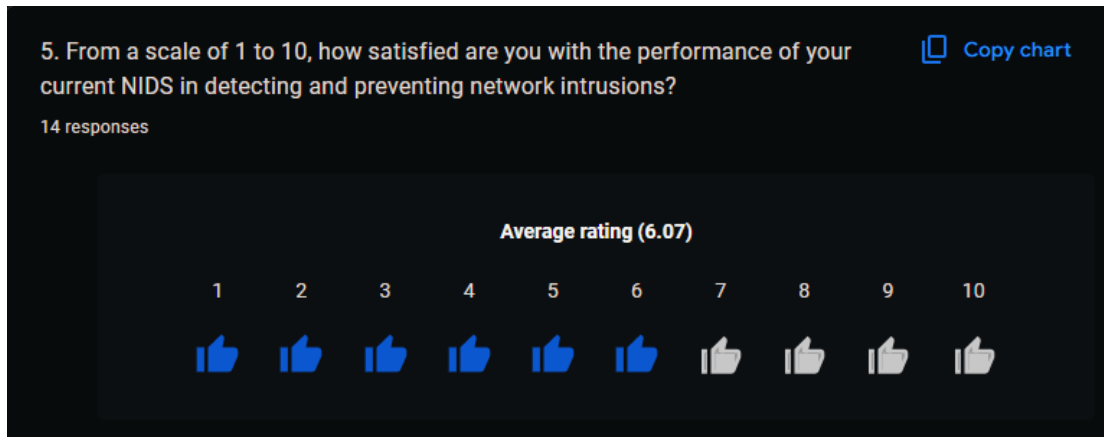
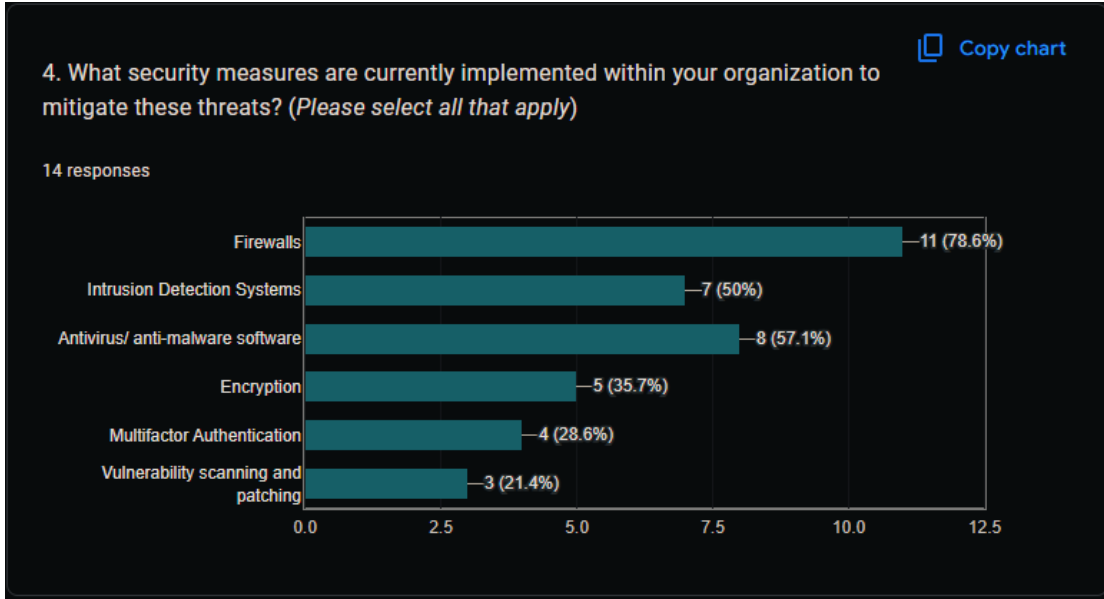
8. If an ensemble NIDS was implemented, what kind of training or support would help you manage it effectively?

Documentation is key, especially around model logic and tuning. A short workshop focused on interpreting alert decisions and thresholds would help us trust and integrate it into our workflow faster.


APPENDIX 4: QUESTIONNAIRE RESPONSES

NB: The following screenshots were taken from the responses tab on Google forms; they are a combined version of all the responses (specifically 14). The author incorporates some of this information to design and develop the system that he tested in Chapter 4.

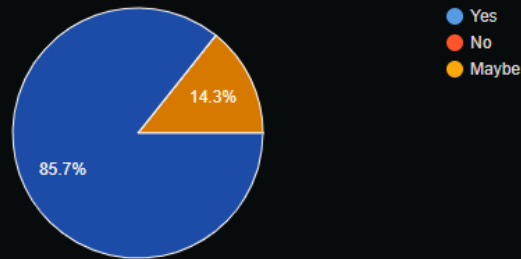





7. Do you believe that incorporating ML/DL algorithms into a hybrid NIDS could improve intrusion detection accuracy and efficiency?

 Copy chart

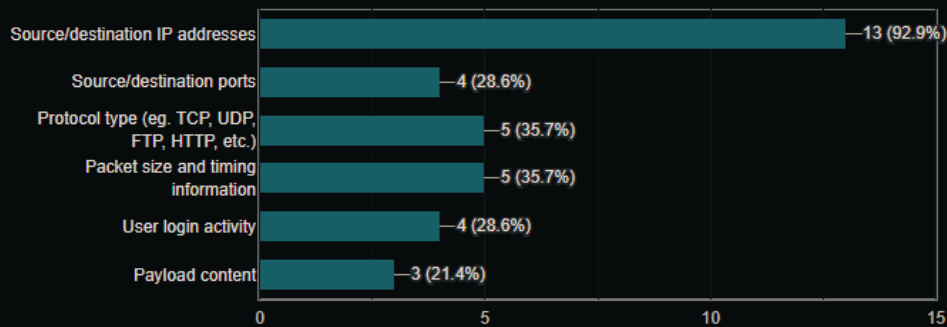
14 responses




8. What specific types of network traffic data do you believe would be most valuable for training ML models for intrusion detection? (Please select all that apply.)

 Copy chart

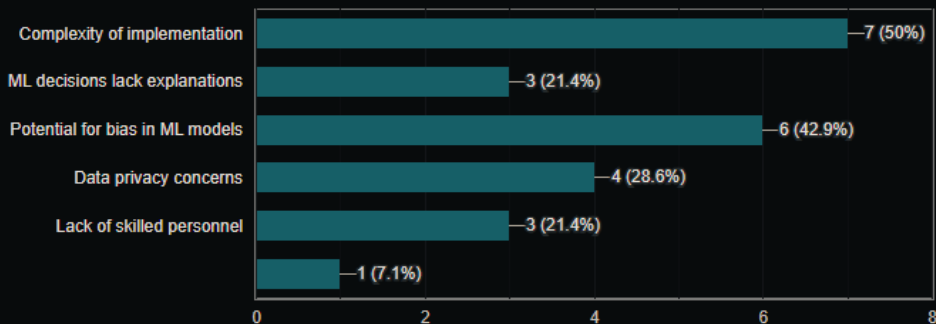
14 responses



9. What are your primary concerns regarding the use of ML/DL in network security? (Please select all that apply)

 Copy chart

14 responses

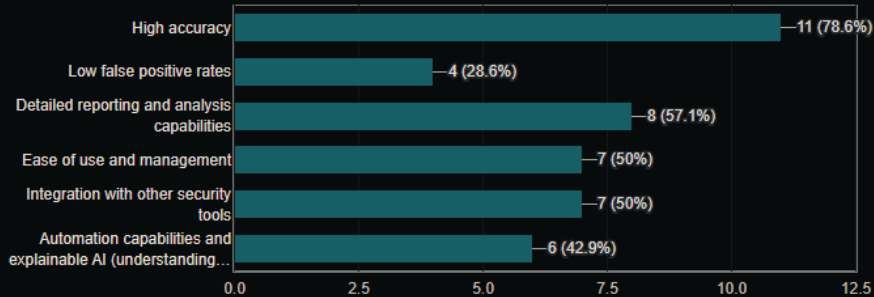


USER EXPERIENCE AND NIDS FEATURES

10. What features are the most important to you in a NIDS? (Please select all that apply)

[Copy chart](#)

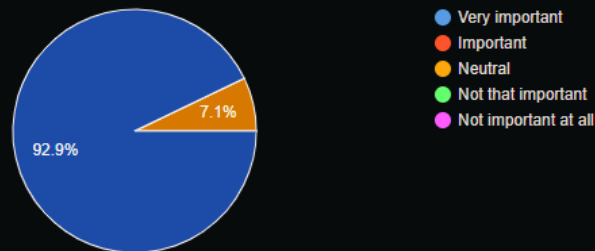
14 responses



11. How important is it to you to have a user-friendly interface for managing and interacting with the NIDS?

[Copy chart](#)

14 responses

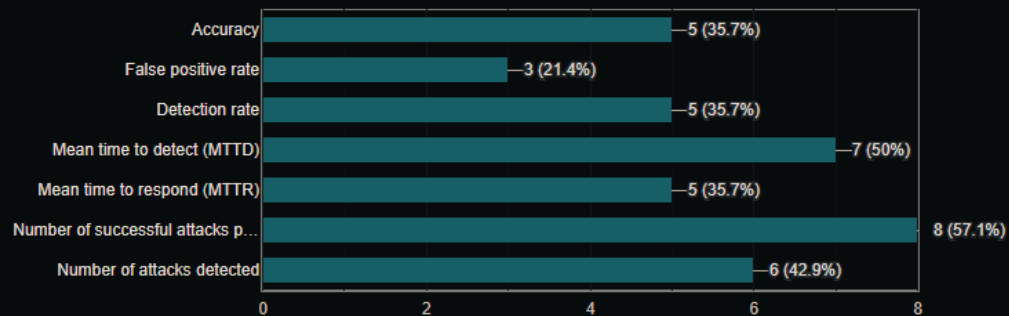


EVALUATION METRICS, TRAINING AND SUPPORT

13. What metrics do you currently use to evaluate the effectiveness of your organization's NIDS? (Please select all that apply)

[Copy chart](#)

14 responses



14. Beyond traditional metrics, what novel metrics do you think would be valuable in assessing the performance of a hybrid NIDS? *(Please provide any suggestions)*

14 responses

DL Techniques

Detection Delay which measures the time taken by the NIDS to detect an intrusion after it occurs.

Adaptive Learning Efficiency Measures how quickly the NIDS adapts to new attack patterns using AI/ML models. Formula: $(\text{Time taken to integrate new threat intelligence} / \text{Total detection accuracy improvement})$, it Ensures the system remains effective against evolving threats.

High performance


Still thinking

Other

Mean time to response

no idea

15. What type of training or support would you need to effectively utilize and manage a hybrid NIDS based on ML/DL models? *(Please select all that apply). all that apply)*

 Copy chart

14 responses

