



UNIVERSITY OF

LIVERPOOL

UNMASKING CYBER THREATS: AI-POWERED NETWORK ANOMALY DETECTION

Enhancing Cybersecurity with Neural Networks and Ethical AI Practices

By

Nikhil Sri Narayana Rasmalla
[201758909]

AN APPLIED RESEARCH DISSERTATION

Submitted to

The University of Liverpool

in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

20th September 2024

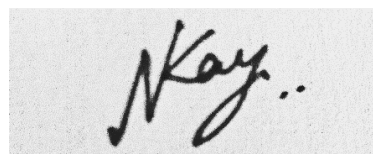
STUDENT DECLARATION

I hereby certify that this dissertation constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I confirm that I have not copied material from another source nor committed plagiarism nor commissioned all or part of the work (including unacceptable proof-reading) nor fabricated, falsified or embellished data when completing the attached piece of work.

I declare that the dissertation describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

A handwritten signature in black ink, appearing to read 'N. Kay..', is written on a light gray rectangular background.

Nikhil Rasmalla

ACKNOWLEDGEMENTS

I am profoundly grateful to my mother, Ms. Ishwarya Lakshmi, and my sister, Ms. Divya Ishwarya Lakshmi, for their continuous motivation of never giving up and for saving my chaotic life, leading me towards success and ensuring that financial constraints never stood in the way of my academic goals and dreams; despite how annoying I could be at times.

I would also like to express my heartfelt appreciation to my mentor, Mr. Narendra Raskatchula – Vice President and Cybersecurity Subject Matter Expert at Deutsche Bank – for his mentorship, motivation, and valuable insights throughout my cybersecurity learning process. I thank him for igniting the spark and drive towards pursuing this challenging yet fulfilling career.

I would like to express my deepest gratitude to my primary supervisor, Mr. Achilleas Koufonikos, and my secondary supervisor, Dr. Leszek Gasieniec, for their unwavering support, guidance, and invaluable insights throughout the course of this project. Their expertise and encouragement were instrumental in shaping both the direction and success of this work. I am also deeply appreciative of their constructive feedback and suggestions, which significantly enhanced the quality of this research.

I would like to extend my sincere thanks to Dr. Alexei Lisitsa, Dr. Tulika Saha, and Dr. Meng Fang, as well as my other professors at the University of Liverpool, for their related coursework and knowledge, which were pivotal in the successful completion of this project. I also would like to thank the Department of Computer Science under the school of Electrical Engineering, Electronics and Computer Science of University of Liverpool for providing the opportunity to pursue my academics at the university.

I also would like to thank my closest friends and their supportive families who always took me under their wing as one of their own during testing times of my life and motivated me to that I can achieve whatever I put my mind to the best of my ability.

Finally, A special note of thanks goes to my peers and colleagues at the University of Liverpool, whose collaborative spirit and intellectual exchanges made this journey both rewarding and enlightening. I am also deeply grateful to the participants and data providers, without whom this study would not have been possible. Their willingness to contribute to this research is truly appreciated.

UNMASKING CYBER THREATS: AI-POWERED NETWORK ANOMALY DETECTION

Enhancing Cybersecurity with Neural Networks and Ethical AI Practices

Table of Contents

ABSTRACT	i
KEYWORDS	i
STATEMENT OF ETHICAL COMPLIANCE	ii
CHAPTER 1: INTRODUCTION	1
1.1 Scope.....	1
1.2 Problem Statement.....	2
1.3 Approach.....	3
1.4 Outcome	3
CHAPTER 2: AIMS AND OBJECTIVES.....	4
2.1 Project Methodology	4
2.1.1. Analyse The Current State Of AI In Cybersecurity	4
2.1.2. Develop An AI Model For Multiple Cybersecurity Applications	5
2.1.3. Evaluate The Effectiveness Of The AI Model	6
2.1.4. Investigate Human And Ethical Factors In AI-Driven Cybersecurity	6
2.1.5. Explore The Practical Applications Of AI In Cybersecurity.....	7
CHAPTER 3: LITERATURE RESEARCH AND CASE STUDY ANALYSIS	7
3.1 AI In Cybersecurity	7
3.2 AI In Network Anomaly Detection And Social Engineering Defence	8
3.3 Challenges And Opportunities For AI In Traditional Cyber Security Systems.....	9
3.4 Gaps In Current Research.....	9
3.5 Use Case Study	10
3.5.1 Administrative Conference Of The United States	10
3.5.2 Department Of Homeland Security	10
3.5.3 The National Cyber Security Centre.....	11
3.5.4 Cybersecurity And Infrastructure Security Agency	11
3.5.5 Darktrace	12
3.5.6 Ernst & Young	12
CHAPTER 4: RESEARCH IMPLEMENTATION	13
4.1 Design Flowchart	13
4.2 Dataset Information	14
4.3 Data Pre-Processing	16

4.4 Data Visualization.....	17
4.5 Model Training.....	18
4.6 Exploratory Data Analysis	19
4.7 Algorithms Planned And Used.....	21
4.7.1 Neural Network Design	21
4.7.2 Cross-Validation And Early Stopping.....	23
CHAPTER 5: AI MODEL FRAMEWORK WORKING PATTERN	24
5.1 Model Framework Overview	24
5.2 Model Architecture And Optimization Strategies	24
5.2.1 Architecture	25
5.2.2 Optimisation Techniques.....	26
5.2.3 Validation Techniques	26
5.2.4 Regularization Techniques	26
CHAPTER 6: EVALUATIONS AND OUTCOMES.....	27
6.1 Performance Metrics.....	28
6.1.1 Accuracy	28
6.1.2 Precision.....	28
6.1.3 Recall	29
6.1.4 F1-Score	29
6.2 Confusion Matrix Analysis And Classification Report	29
6.3 Average Cross-Validation Performance	31
6.4 Key Insights On Model Performance And Capabilities	31
CHAPTER 7: CHALLENGES FACED	31
CHAPTER 8: CONCLUSION.....	33
CHAPTER 9: FUTURE SCOPE AND DEVELOPMENT	33
CHAPTER 10: BCS PROJECT CRITERIA & SELF-REFLECTION	35
REFERENCES.....	37
APPENDIX	42

TABLE OF FIGURES

Figure 1- Flowchart showcasing the design of the model.....	14
Figure 2 - Table showcasing the attack categories present in the UNSW-NB15 (Moustafa & Slay, 2015) dataset along with its corresponding definitions.	15
Figure 3 -Table showcasing the attack categories present in the UNSW-NB15 (Moustafa & Slay, 2015) Training dataset along with its count and percentage.....	15
Figure 4 - Table showcasing the attack categories present in the UNSW-NB15 (Moustafa & Slay, 2015) Testing dataset along with its count and percentage.....	15
Figure 5 - Correlation matrix of UNSW-NB15 dataset	16
Figure 6 - Graph showcasing Attack category distribution in UNSW-NB15 Training data	17
Figure 7 - Graph showcasing Attack category distribution in UNSW-NB15 Testing data.....	18
Figure 8 - Flowchart showcasing Supervised Learning in action.....	19
Figure 9 - Graph showcasing Protocol distribution in UNSW-NB15 dataset	20
Figure 10 -Graph showcasing Services distribution in UNSW-NB15 dataset.....	20
Figure 11 - Neural Network of the model	21
Figure 12 - True and False Positives/Negatives matrix	28
Figure 13 - Confusion Matrix of the final trained model	30
Figure 14 - Classification Report of the Final Model.....	30
Figure 15 - Code Snippet of Python packages used	42
Figure 16 - - Code Snippet of loading and handling data.....	42
Figure 17 - Code Snippet of Preprocessing data and Data pipeline	42
Figure 18 - Code Snippet of creating the model	43
Figure 19 - Code Snippet of Fine tuning the model	43

ABSTRACT

The current generation threats such as zero-day exploits, advanced persistent threats APTs, and social engineering attacks have become more complex and hence have rendered the conventional security mechanisms ineffective. This work aims to examine the use of Artificial Intelligence in improving cyber security with an emphasis on network anomaly detection. Through the analysis of AI in fraud detection, User Behaviour Analytics, and Social Engineering Prevention the research studies the possibilities of enhancing the efficiency of threat identification by reflecting on technical, human, and ethical aspects. The developed machine learning model is a neural network-based model that has been trained using the UNSW-NB15 dataset, containing normal and malicious network traffic. To avoid the overfitting of the model and enhance the model's generality, batch normalization, dropout layers, and early stopping were implemented. The performance of the model in detecting different anomalies in the network was measured with the help of metrics such as accuracy, precision, recall, and F1 score, and it was found that the cross-validated accuracy of the model is 99.89%. Some difficulties were noticed in the identification of the rare attack types because the dataset was unbalanced. The study also discusses the issues of ethics as they relate to data, protection, and equality while promoting the right use of AI in cybersecurity.

KEYWORDS

Cybersecurity, Supervised Machine Learning, Artificial Intelligence, Network Anomaly, UNSW-NB15 Dataset, Cybersecure Systems, Internet of Things, Deep Learning, Social Engineering, Ethical Considerations.

STATEMENT OF ETHICAL COMPLIANCE

This research project respects stringent ethical standards to guarantee the appropriate usage of data. Because anonymous human data from public APIs or open sources was used in this study, it comes under Category B data. Non-personally identifiable information will be obtained from publicly accessible datasets along with the literature available on the topic. The emphasis will be on complaints of human error, phishing datasets, and cybersecurity problems. Any accidental inclusion of potentially sensitive information will be obscured and removed to protect privacy and confidentiality. As there will be no use of additional human participation in any activity, this study falls under Category 0 because no human volunteers will be directly involved. The study won't include any human or animal participation in any surveys, trials, or other data-generating activities; instead, it will only analyse already-existing data sets. The General Data Protection Regulation (GDPR) policies and institutional ethical norms, as well as data protection legislation, shall be closely adhered to by all data processing operations. The project will guarantee that all data is treated lawfully, impartially, and transparently in compliance with GDPR. The principles of data minimization shall be adhered to, guaranteeing that only essential data is gathered and handled. Appropriate organizational and technical safeguards will be put in place to protect data from loss, destruction, or unauthorised access. The project guarantees the integrity and ethical accountability of the research process by abiding by these ethical principles and GDPR, protecting the privacy and rights of all persons who are indirectly engaged using data. This methodology ensures that the research will be carried out with the greatest ethical standards in mind, concentrating solely on the theoretical investigation of AI applications within the given constraints.

CHAPTER 1: INTRODUCTION

One of the biggest risks related to contemporary society is the probability of cybercrime. There is no doubt that hackers are now in the process of thinking of new strategies that they can use to penetrate the system and create more damage. Some of these are data loss, ransomware, and denial of service (DoS) among others that can, target an individual, organization, or industry. Such threats can sometimes not easily be addressed by conventional security measures and therefore require more advanced forms of security such as network anomaly detection.

Network anomalies are unusual patterns of traffic that are contrary to normal traffic patterns and may show signs of danger. These may include, for instance, the changes in the number of requests, the instability of the time of access, or the alteration in the users' actions. Such anomalies if not detected can cause insecurity in the network and hence the need to detect them in good time. The current situation cannot be adequately addressed by conventional models that are developed based on certain standards and conditions. According to (Saied, et al., 2024), the current networks and IoT devices' integration and compatibility require AI-based security measures to enhance security. AI's ability to learn from large volumes of data and its ability to distinguish small differences also makes it suitable for anomaly detection.

This project focuses on how AI can be used in enhancing cybersecurity with a concentration on detecting various anomalies in the network. It deconstructs the human, technical, and ethical aspects of AI use in the security domain. The practical application is the usage of the proposed machine learning model that is based on the UNSW-NB15 (Moustafa & Slay, 2015) dataset to model normal traffic and different types of attacks and demonstrate how its integration can improve the mechanisms used for the detection of anomalies.

1.1 Scope

This project is not limited to network anomaly detection but searching for ways of AI's incorporation into the other parts of the cybersecurity environment. AI's integration with Intrusion Detection Systems (IDS) and Security Information and Event Management (SIEM) applications seems to be particularly suitable

since it improves performance in identifying and addressing threats in real-time. Further, AI pushes up other spheres including fraud detection, user behaviour analysis as well as the search for phishing attempts based on heuristics that point to potential threats or social engineering attempts.

According to (Jada & Mayayise, 2023), on a similar note, AI influences the automation of threat identification, vulnerability handling, and actionable decision-making in organizational cybersecurity systems. Through these AI functions, organizations manage to be prepared for any emerging threat due to the functions' automation. Additionally, as pointed out by (Garcia-Teodoro, et al., 2009), anomaly-based IDS are essential in the discovery of novel threats because of their potential to detect irregularity in the network traffic. The machine learning model in this project is used to identify anomalies in network traffic but can be used in general for new cybersecurity applications making an organization more secure. This also enables it to be deployed in different network settings ranging from small businesses to large industrial networks.

1.2 Problem Statement

New threats are on the rise while conventional security solutions and tactics are nearing their capabilities or cannot be applied to certain situations at all. New threats including zero-day exploits, multi-vector attacks, and advanced persistent threats (APT) pose a lot of risks, and they can easily penetrate the conventional system to access the networks. According to (Jada & Mayayise, 2023), it is agreed that AI solutions are significant in improving existing security strategies and managing destructive attacks beyond the capacity of human operators. Another major issue of cybersecurity is the possibility of detecting these advanced threats in real time, which is what AI-based anomaly detection systems attempt to provide. The central research question posed by this project is:

Can AI effectively enhance the accuracy and speed of detecting network anomalies, while also addressing the ethical considerations and human factors involved in modern cybersecurity?

This question reflects the need for scalable and adaptable AI-driven systems capable of detecting a wide range of threats more efficiently than traditional methods. In addition, the research emphasizes the importance of ensuring ethical AI usage and supporting human decision-making in security operations.

1.3 Approach

This research is theoretically based and at the same time, implemented in its practical aspect. In the theoretical part, the project investigates modern tendencies identified in AI use in cybersecurity, such as fraud detection, user behaviour analytics, and anomaly detection. Thus, this work lays the conceptual groundwork for determining how AI can best tackle major cybersecurity issues. AI as pointed out by (Kaur, et al., 2023) can also be effectively applied in the following: Automated vulnerability detection: leveraging deep learning and other types of techniques in detecting security threats in software and cyberspaces.

As a practical aspect, the work focuses on creating an ML model using Supervised Learning and the UNSW-NB15 (Moustafa & Slay, 2015) data set for the identification of network abnormalities. This model uses a neural network structure which as mentioned earlier can be fine-tuned for the reduction of overfitting using some methods such as the batch normalization technique, dropout, and early stopping. The study also addresses the use of artificial intelligence in social engineering prevention, which does not involve the exploitation of a computers or a software's weak points but targets people. Altogether, the presented project recognized the steps in the technical advancement of AI for cybersecurity while clearing the human factors concern, which is an advantageous approach.

1.4 Outcome

Some of the expectancies of this project include a better understanding of the technological aspects of AI in cybersecurity as well as the social factors. On the technical level, after the completion of the project, it will be possible to have a functional AI model that would be able to identify network abnormalities. It can be used as a plug-and-play with other traditional IDS, SIEM, and other security tools to strengthen real-time monitoring qualification for higher-level security defence. Both (Jada & Mayayise, 2023) provide clarification to this by saying that with the application of Artificial Intelligence, human errors can be minimized in the various aspects of cybersecurity where critical functions including the monitoring of vulnerabilities as well as possible threats that can be performed through artificial intelligence to support decision making thus enhancing the organizational security.

On the human side, this research looks at the way AI can prevent social engineering attacks, where people's behaviour is exploited to compromise a system. Furthermore, it discusses the ethical issues of applying Artificial intelligence in Cybersecurity which include Privacy, Explanation and Accountability, and Fairness of AI's mind. This project believes that the combination of both technical and non-technical elements will therefore be able to give a balance of the solution to contemporary cybersecurity challenges.

CHAPTER 2: AIMS AND OBJECTIVES

The primary aim of this project is to investigate how Artificial Intelligence (AI) can be applied effectively to various aspects of information security, with a focus on network anomaly detection and exploring its potential in fraud detection, user behaviour analytics, and social engineering prevention. By analysing these areas, the research seeks to improve both the speed and accuracy of detecting cyber threats and demonstrate how AI can be integrated into a wide range of security solutions, addressing both technical and human-centric challenges.

2.1 Project Methodology

This project follows a structured methodology combining theoretical research with practical implementation. The methodology focuses on a multi-faceted approach to AI applications in cybersecurity, with the following objectives:

2.1.1. Analyse The Current State Of AI In Cybersecurity

Cybersecurity is the most popular application these days and has incorporated using AI in improving threat identification and prevention measures. Machine learning models are actively used for network anomaly detection, fraud detection, and user behaviour analytics that enable finding abnormal behaviour that can be a sign of a security breach, fraud, or insider threat. This is a disadvantage with traditional systems as they are unable to change with the newest threats as AI can change with the new threats and implementing them in a real-time fashion. (Samunnisa, et al., 2022). state that, anomaly-based Intrusion Detection Systems (IDS) can identify known, unknown attacks such as zero-day attacks, but they are usually sensitive to a lot of tuning and have high false positives rates.

(Samunnisa, et al., 2022) explain that anomaly-based Intrusion Detection Systems (IDS) can identify known and unknown attack types including zero-day attacks, but such IDS need tedious tuning and, therefore, might exhibit high false positive rates. AI models can be trained to predict and classify multiple forms of anomalies that occur in network communications including DoS attacks, exploits as well as reconnaissance activities. As (Samunnisa, et al., 2022) detail, IDS considers these anomalous as classification issues, the kind of attack types are DoS, probe, and U2R. When these anomalies are detected, then AI models minimize or prevent security threats and can recognize patterns in user activity that show the intention of fraud or insider threats.

2.1.2. Develop An AI Model For Multiple Cybersecurity Applications

AI models can be developed to improve several cybersecurity functions because the applications of the model are highly adaptable to recognize variances in threats, apart from the layer anomaly detection in the network, AI models have become useful in the area of fraud detection and user behaviour analysis. According to (Kaur, et al., 2023), more organizations are now adopting AI as an essential tool in the cybersecurity system since it helps in handling activities like vulnerability management independently alongside human teams. In a situation where the threats act on the actual technical aspects of the network and then active threats like social engineering, the two threats can be combated by the model.

In cases where new threats including the well-known zero-day vulnerabilities manifest themselves, then the AI model must be retrained. (Biggio & Roli, 2018) underscores the importance of updating the model with fresh training as a way of preventing adversarial attacks as well as newly discovered vulnerabilities. This can help to guarantee that the model can identify new patterns of malign activity and counter instances of threats of this kind in real-time.

The facts defined above allow us to build an AI model with a neural network architecture using such techniques as batch normalization, dropout, early stopping, etc. Such capabilities help the model to detect multifaceted patterns in network traffic, users' activity, and financial transactions, allowing it to be versatile enough to be employed in different cybersecurity contexts.

2.1.3. Evaluate The Effectiveness Of The AI Model

The AI model can be trained to predict or classify various types of anomalies in the network traffic including possible intrusions and attacks. It is designed to clear the distinction between normal network traffic and different types of malicious actions, DoS, exploits, reconnaissance, etc. Thus, given these anomalies are identified as shown by the model above, the possible security breaches can be averted, or their impact minimized.

The performance of the proposed AI model will be examined by benchmark measures like accuracy, precision, recall, and F1 score. These metrics will evaluate the ability of the model to detect various types of anomalies, fraud patterns, and complex user behaviour with a very low number of false positives to ensure security teams are not overwhelmed.

2.1.4. Investigate Human And Ethical Factors In AI-Driven Cybersecurity

Social engineering risks taking advantage of human emotions because the targets are tricked into giving out their details or getting into a system without permission. (Mathews, et al., 2007) have also explained how the use of NLP techniques includes the content of the emails and URL in the process of phishing identification with the help of automated means for such threats. In this project, the AI model is to identify the pattern of user behaviour that might signify a social engineering attack to assist human operators in handling these threats more appropriately.

As AI has widely been adopted to enhance cybersecurity, there are paramount essential questions of ethics including data privacy, transparency, and AI bias. (Brundage, et al., 2018) state that the AI models must be privacy-sensitive so that the fundamental workings of the AI systems do not conceal biases that may culminate in unfair or discriminative outcomes. To mitigate these issues, this project will ensure that the AI model is well compliant with the privacy concerns while also providing output that will be easy to explain to the users hence enhancing confidence in the results generated by an automated cybersecurity system.

2.1.5. Explore The Practical Applications Of AI In Cybersecurity

AI possibilities can be extended far beyond network anomaly detection; AI also can include various fraud detection and even inside threat detection based on user behaviour analysis. In their paper, (Kaur, et al., 2023). note that speed, can evaluate a massive amount of data and provide insights and analysis for prevention against new forms of cyber threats. Further, AI can revolutionize IDS and other related SIEM platforms in that they can automatically recognize threats in real-time.

This AI model proposed in this project is more flexible for various other applications apart from network security such as fraud detection and social engineering. As highlighted by (Aldhaferi, et al., 2023), the emergence of IoT has increased the cybersecurity challenges because of the high connectivity in devices, most of which are not very secure. The ability to scale up and down to various networks such that it caters to the new challenges of IoT, and other related technologies makes it flexible for use in small enterprises, medium businesses, and even large industrial systems.

CHAPTER 3: LITERATURE RESEARCH AND CASE STUDY ANALYSIS

This chapter reviews key studies that demonstrate how AI is being used to enhance network anomaly detection and cybersecurity. It covers the major techniques applied, the challenges faced, and the gaps that still exist in research, offering context for the AI model developed in this project.

3.1 AI In Cybersecurity

AI has immediately emerged as a critical model for cybersecurity and is exceptionally valuable when it comes to struggling with network-based threats that are hard for conventional tactics to manoeuvre. Modern complex threats may overwhelm the traditional rule-based security solutions which are not as effective as one would like. These shortcomings can be alleviated by applying AI which can patterns and outliers in real time and, therefore, can identify existing and emerging threats that normal systems might fail to capture.

AI can act as an enhancement to IDS and SIEM systems, especially regarding the application of machine learning and deep learning in the process. For example, IDS based on artificial intelligence can learn network traffic and adjust to threats over time. This is especially beneficial in today's security situations, as in malicious attacks, new forms of attack occur frequently. (Buczak & Guven, 2015) note that the use of AI helps in the reduction of false positives and provides a more accurate and efficient way of identifying threats. Their work focuses on the application of machine learning to anomaly detection and proves that AI is beneficial in improving the effectiveness of detection systems in different network scenarios.

3.2 AI In Network Anomaly Detection And Social Engineering Defence

Network anomaly detection is mainly concerned with the detection of network behaviours that seem unusual and may be a sign of a threat. AI can reinforce signature-based as well as anomaly-based detection techniques. Signature-based systems look for these attack patterns, but AI-performed anomaly detection can clearly distinguish between normal network traffic and threatening traffic and is therefore crucial for new or unknown threats. Standard procedures such as clustering and classification serve to put similar activities together, thus making discrepancies easily discerned. Stemming from (Apruzzese, et al., 2022) literature, many works have shown the potential of applying AI based on unsupervised learning for clustering and outlier detection in improving the anomaly detection rates.

However, AI's added functionality includes the ability to combat social engineering attacks in addition to identifying network aberrations. Social engineering takes advantage of human behaviour to obtain such things as passwords, and this is seen in phishing. The identified manipulative tactics can be detected by analysing communication patterns; metadata of emails exchanged; and tones, especially in language. A few examples of the findings of the authors include (Nazmul Alam, et al., 2020) have revealed in their recent studies that through training, it is possible to develop a machine learning approach for identifying phishing attempts from the metadata and content of emails. As a result, AI can stall specific behaviours of the security team as phishing attempts or other social engineering attacks, which can minimize breaches that target variations in human characters.

In addition, AI-based systems must address issues such as noise, a case whereby normal behaviour is classified as malicious. This may prove cumbersome for security teams as it floods them with more

alerts than what is required. This issue can be solved using fine-tuning and better model training where AI can enhance the probability of the correct anomaly detection. (Xin, et al., 2018) point out how the optimization of AI models can help in lowering the number of false positives while enhancing the overall detection rate in cybersecurity networks.

3.3 Challenges And Opportunities For AI In Traditional Cyber Security Systems

AI has proven very beneficial, especially in facilitating the enhancement of the Intrusion Detection Systems (IDS) and SIEM platforms. It is worth mentioning that IDS and SIEM are not capable of identifying unknown threats while using AI, a system is programmed to evolve with information gathered from actual network traffic analysis. This capability makes the IDS system developed under AI control particularly useful in large networks with complicated connections including the IoT networks. About AI, (Jada & Mayayise, 2023) think that AI comes in handy in approaching and handling the increased scale and interaction of the contemporary cybersecurity environment, especially IoT systems.

Finally, it can also enhance the performance of the SIEM platforms through the automation of data processing and analysis that involves big security data. This makes identification of the cyberattacks easier and reduces the overall response time thus minimizing the effect of the damage. There are several concerns concerning AI that it must overcome among them being transparency and interpretability. Of the two, neural networks pose a special problem since they frequently act as ‘black boxes’, where the reasoning behind an analysis decision is hard to fathom. Introducing more transparency to AI systems is important in making people trust these systems in cybersecurity and use them more often. Several recent papers highlight the importance of explanation ability as the means to build trust with cybersecurity professionals, according to (Zhang, et al., 2022).

3.4 Gaps In Current Research

In essence, AI has the potential to drastically enhance cybersecurity defence; however, some key aspects require improvements. For example, most AI models are trained on static datasets which do not capture the dynamic nature of actual cyberspace threats. Using available data, (Sharafaldin, et al.,

2018) noted that the datasets employed are static in most cases, while the modern cyber threats are dynamic, thus limiting the ability of AI to generalize from one attack to another.

Again, the use of AI in today's security landscapes and even zero trust architectures is relatively new. Zero Trust should be a continuous process of validation, and it is these areas that may benefit from utilizing AI. Nonetheless, there is a scarcity of research on AI integration with Zero Trust, as observed by (Rose, et al., 2020).

The second gap concerns the ability of AI-based systems to identify more sophisticated attacks that are, for example, multi-step and advanced persistent threats (APT). Currently, most conventional models are more proactive in identifying individual outliers but are less capable of identifying these coordinated attacks. Further studies in this line could enhance AI's efficiency in combating complicated cyber threats. Contrary to this, (Alshamrani, et al., 2019) advocate for enhanced methodologies, which can comprehensively map APTs not only at different stages.

3.5 Use Case Study

3.5.1 Administrative Conference Of The United States

The study by Engstrom et al. on the use of AI in federal agencies stresses one of the fundamentals, that improved data feeds must be fed into AI models and that risks emerging with the use of AI must be ongoingly mitigated via model updates. From their research one can understand how if used correctly AI can enhance the processes of anomaly detection, efficient resource utilization, and create new solutions for the modern matters of governance and security. They also identify some of the difficulties faced in AI governance like how to handle large datasets and how to keep the AI systems performing optimally through the continuous updating of models and so on, which makes all these principles paramount in the creation of sound security architectures (Engstrom, et al., 2020).

3.5.2 Department Of Homeland Security

The Department of Homeland Security (DHS) is the United States government agency responsible for protecting the nation's residents from physical and cyber threats uses AI in several cybersecurity settings for identification and mitigation (Department of Homeland Security, 2024), which is valuable for

my background research as it demonstrates how AI is applied in actual cybersecurity operations. For example, AI is adopted for the next-generation network anomaly alert, enabling analysts to identify high-fidelity anomalies in the network traffic via machine learning (Department of Homeland Security, 2024). This proves that AI can also advance threat hunting by processing large data sets, and improving alarms and possible threats which portrays how automation through AI is crucial in managing modern-day threats. Moreover, DHS states that AI is applicable for monitoring critical infrastructure and reverse-engineering malware to depict the extensive use of AI in identifying and preventing modern cyber threats (Department of Homeland Security, 2024). This context is useful to my work as it shows that AI-based tools are already influencing cybersecurity and amplifies the need for higher intelligent and adaptive AI models in securing crucial systems.

3.5.3 The National Cyber Security Centre

The National Cyber Security Centre (NCSC) is a central body in the UK that plays the role of an expert in the field of cybersecurity and is aimed at protecting the UK's cyberspace as well as offering advice to protect the national rear in the sphere of information security. It oversees and deploys AI agents in almost every aspect of cyberspace and incurs a critical responsibility of managing opportunities and threats of AI technology. For example, NCSC uses AI to spot and eliminate cyber threats working with a Great Volume of data and improving the speed and efficacy of the identification of unobtrusive malicious practices, an important factor that is related to my project on network anomaly detection (National Cyber Security Centre, 2023). AI in this regard can also be employed in analysing complex patterns in the various systems as the protective DNS services, to detect before exploitation situations that possibly lead to cyber-attacks. Further, according to NCSC, AI is applied for mutated malware recognition which proves how enhanced Machine learning models enhance the rate of identifying threats (National Cyber Security Centre, 2023). This shows how AI is not only about pattern detection but also in predicting points of cyber weakness, giving credence to the need for strong Artificial Intelligence in developing current and future security frameworks.

3.5.4 Cybersecurity And Infrastructure Security Agency

The Cybersecurity and Infrastructure Security Agency (CISA) is a United States federal agency that specializes in protecting the nation's key infrastructures and enhancing cyber stability. Relying on

artificial intelligence in several areas to improve cybersecurity. For example, the AIS system utilizes AI as its means to enhance trust in the indication of cyber threat indicators by using machine learning and natural language processing for real-time threat intelligence (Cybersecurity & Infrastructure Security Agency, 2024). This illustrates that advanced analytical tools are employed for forensic purposes; this enables specialists to identify discrepancies and possible risks within a large data set at a faster rate. This shows how AI can be employed in identifying network anomalies and threats, as a way of proving the viability of AI technology in the real world. Additionally, CISA uses AI in the automation of alerting, PII detection, and vulnerability reporting, which conforms to privacy laws when correlating analyst's tasks, which is critical for today's cybersecurity frameworks (Cybersecurity & Infrastructure Security Agency, 2024). This reiterates the role of using AI automation to address complex cybersecurity issues, reinforcing the need to design AI models that are flexible in addressing threats and securing infrastructure and network systems.

3.5.5 Darktrace

Darktrace is an AI-led cyber security company that was formed to offer an AI-based solution for threat detection and response in real time. Uses artificial intelligence to identify and prevent complex threats like metamorphic malware which is very important to my study as it characterizes the usefulness of artificial intelligence in detecting advanced persistent, self-evolving attacks that bypass conventional analytical models (Fier, Justin, 2017). For instance, it was acknowledged and dealt with 'Smoke Malware Loader', a type of malware effective at changing its identity, penetrating the university networks as well as avoiding perimeter checks at a major university in the US. This exemplifies how using AI demonstrates that it can identify suspicious activity, highlighting the fact that the malware's strange HTTP traffic was detected, and the affected devices were isolated before more harm could be caused underlines the effectiveness of AI in prophylactic cybersecurity.

3.5.6 Ernst & Young

EY [Ernst & Young] is a global consulting firm and part of the Big 4 consulting firms. Emphasizes the importance of AI in moving cybersecurity from a purely reactive and cost-centre to a value-creating activity, which is relevant to my study on the implementation of AI in cybersecurity systems (Watson, et

al., 2024). Their action points focus on what an organization should do with AI in the context of detection, response, and automation in SOCs thus, enhancing efficiency and reducing the resources needed, an aspect my project will seek to address. For instance, when discussing disruptive technologies, EY provides an insight into how AI deploys detection processes to help the teams scrutinize vast volumes of data within a live setting to detect threats faster and with higher degrees of precision than are possible using other techniques. This fosters the integrity of systems that are meant to prevent or mitigate the impact of an attack before it reaches catastrophic levels (Watson, et al., 2024). In addition, EY discusses how organizations should integrate AI safely into their operations and how new AI solutions should resist emerging risks (Watson, et al., 2024), thus fitting the overall objective of establishing flexible AI security frameworks within my study.

CHAPTER 4: RESEARCH IMPLEMENTATION

This chapter provides a detailed overview of the steps involved in the design and development of the AI model for network anomaly detection using the UNSW-NB15 dataset (Moustafa & Slay, 2015). The methodology covers data pre-processing, model design, training, and evaluation, focusing on how these methods contribute to improving cybersecurity through AI-driven solutions.

4.1 Design Flowchart

A clear plan was followed when developing the AI model to provide an easy comparison and understanding of the workflow. The steps adopted for the inception of the project were data loading where the UNSW-NB15 dataset (Moustafa & Slay, 2015) was loaded into the environment and data pre-processing. The model was trained on the training dataset and tested on the testing dataset. Among them were instances of removing the unnecessary features which are just added columns, categorical feature conversion, and feature scaling about numeric variables for model training. The following step entailed creating the neural network model by using the Keras framework and this was designed with multiple dense layers, and dropout layers to avoid overfitting in the model as well as batch normalization to enhance the stability during training. After the model was designed, it was also validated by cross-validation which is a technique of checking the model performance randomly on different parts of the data. As for the model evaluation, a confusion matrix and a classification report were used to evaluate

the model. Finally, the best-fit model was saved for future use after training was completed and the validity of the model was checked using a cross-validation method.

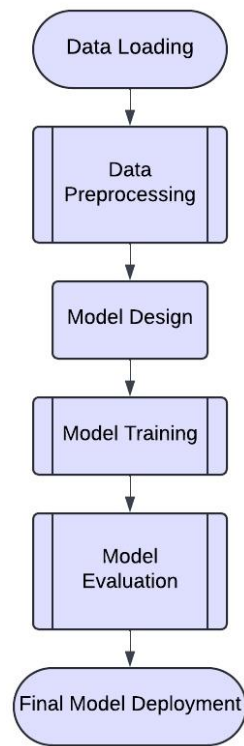


Figure 1- Flowchart showcasing the design of the model.

4.2 Dataset Information

To use features and classify the network traffic, the UNSW-NB15 dataset (Moustafa & Slay, 2015) was used for training and testing of the model because the dataset defines both normal and malicious network traffic. This dataset contains 49 features to portray miscellaneous attributes of network packets where we have numerical and categorical data. The count data contains quantitative values such as packet length or duration; the others are categorical, e.g., proto (type of protocol), service, and state showing the nature/character of connections in the network. Further, it contains both normal and anomalous traffic data to make the model highly realistic and can be easily trained or tested in the presence of realistic traffic patterns. The UNSW-NB15 dataset (Moustafa & Slay, 2015) was selected purposely due to its closeness to the real-world network traffic and the available attack types such as fuzzers, backdoor attacks, and DoS attacks which would help determine the performance of the model

in detecting several types of network anomalies. Again, as pointed out by (Moustafa & Slay, 2015), the models trained on big databases such as UNSW-NB15 enable the detection of a wide range of threats with good precision and, therefore, confirm the choice of this dataset.

Attack Category	Description
Analysis	Analyzing the target network or system to identify vulnerabilities
Backdoor	Malicious software that provides unauthorized access to a system
DoS	Denial of Service attacks that overwhelm a system to make it unavailable
Exploits	Attempts to exploit vulnerabilities in software or systems
Fuzzers	Programs designed to fuzz or test the stability/security of systems
Generic	Generic attacks that do not fit into other categories, often encompassing multiple attack methods
Normal	Normal, legitimate network traffic without malicious intent
Reconnaissance	Gathering information about a target network to identify potential attack vectors
Shellcode	Code injected into a system to execute unauthorized commands
Worms	Malware designed to replicate and spread across networks

Figure 2 - Table showcasing the attack categories present in the UNSW-NB15 (Moustafa & Slay, 2015) dataset along with its corresponding definitions.

Attack Category	Count (Training)	Percentage (Training)
Normal	56000	31.93776698
Generic	40000	22.8126907
Exploits	33393	19.04460451
Fuzzers	18184	10.37064919
DoS	12264	6.994370969
Reconnaissance	10491	5.983198453
Analysis	2000	1.140634535
Backdoor	1746	0.995773949
Shellcode	1133	0.646169464
Worms	130	0.074141245

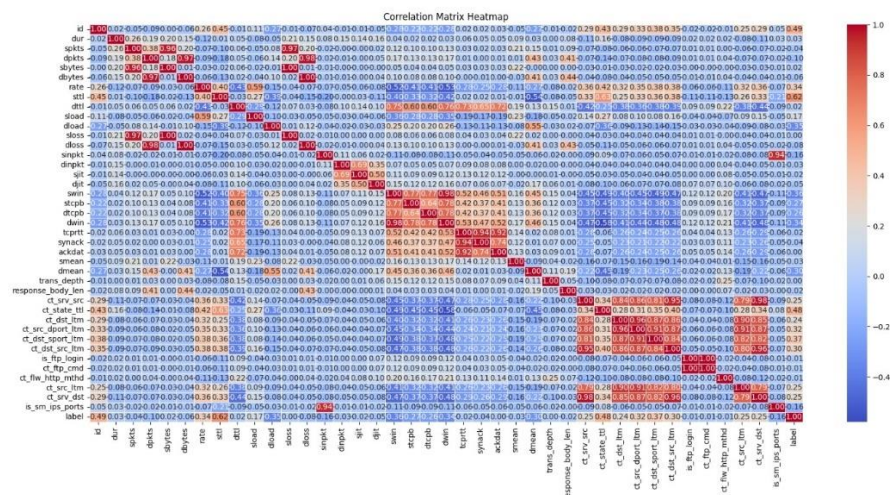
Figure 3 - Table showcasing the attack categories present in the UNSW-NB15 (Moustafa & Slay, 2015) Training dataset along with its count and percentage

Attack Category	Count (Testing)	Percentage (Testing)
Normal	37000	44.93999903
Generic	18871	22.9206141
Exploits	11132	13.52086673
Fuzzers	6062	7.362872273
DoS	4089	4.96647719
Reconnaissance	3496	4.246222611
Analysis	677	0.822280523
Backdoor	583	0.708108633
Shellcode	378	0.459116747
Worms	44	0.053442161

Figure 4 - Table showcasing the attack categories present in the UNSW-NB15 (Moustafa & Slay, 2015) Testing dataset along with its count and percentage

4.3 Data Pre-Processing

It was important to prepare the data in the right form to be used in training the AI model in this case data pre-processing was done. The first operation was to drop uncritical features such as id and label as they were not helpful in the learning process. This was done since our model was aimed to detect the attack type based on traffic using multilabel classification of the attack categories rather than binary classification of the label. Some of the features including proto, service, and state were converted into binary format since the model could not interpret categorical features well. To the numerical set as a feature, the StandardScaler was used to standardize the data so that they were to have a mean of zero and a standard deviation of one. Also, the sparse matrices that are produced during the time of encoding were converted to dense matrices to make them compatible with the neural network. In the pre-processing of data, special attention was paid to the fact that the AI model was given the best data set for training which enhanced both the response accuracy and versatility. Storing the pre-processing steps using joblib also helps in the future project to reuse those steps easily.



4.4 Data Visualization

In this project, data preparation and extraction were another central concept that involved data visualization and feature engineering that helped in analysing the characteristics of the dataset and in extracting features that can improve the performance of the model if included in the dataset. More of these included the bar chart used in representing the attack categories' distribution. An example was the fact that in terms of class balance, there were more apparent attacks like DoS as opposed to rare kinds of attacks like Worms. It is in defining the imbalances above that one can explain the differences in precision and recall across various categories. Moreover, other measurements such as the correlation matrix which shows correlations between numbers in a heatmap format were applied in the comparisons of numerical characteristics. This kind of visualization was useful for gaining insights about feature dependencies to filter the dataset and be left with what is most descriptive. Following this process of feature engineering that these visualizations entailed, the data was properly pre-processed for the anomaly detection model thereby enhancing its capability of detecting intricate patterns in the network traffic.

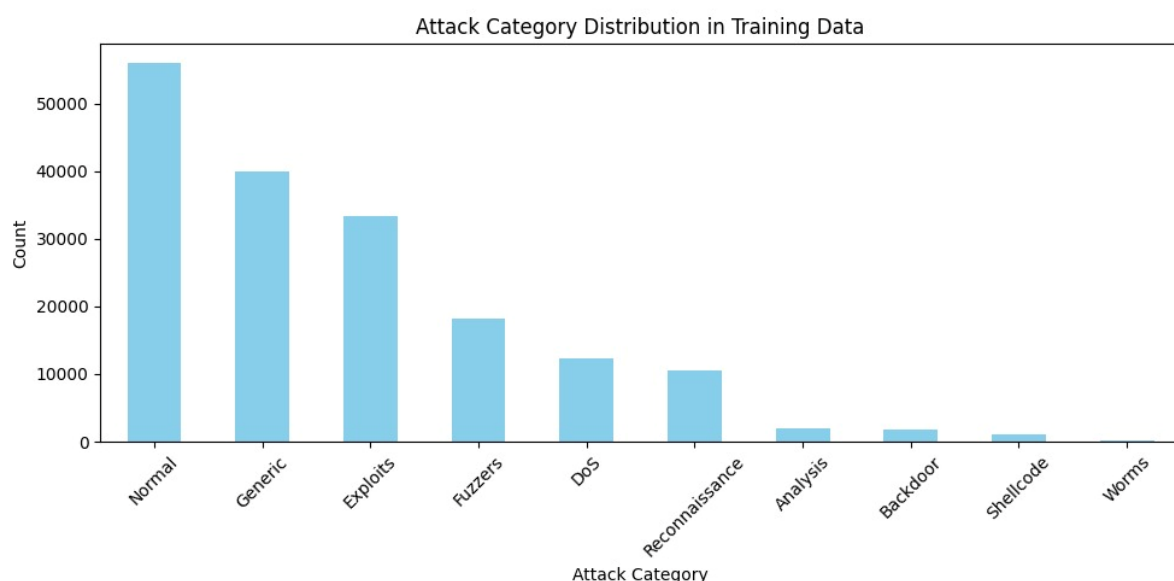


Figure 6 - Graph showcasing Attack category distribution in UNSW-NB15 Training data



Figure 7 - Graph showcasing Attack category distribution in UNSW-NB15 Testing data

4.5 Model Training

Supervised learning was applied in this model with the help of the labelled data from the UNSW-NB15 dataset which contains various features of network traffic and their corresponding attack categories. First, the non-required features including the 'id' and 'label' columns were removed from the training and the testing datasets. These categories of attacks were then converted into numerical labels using the label encoding method. In the training process the model was provided with pre-processed data, which means that numerical features were normalized, and categorical features were encoded by one-hot encoding. This made the model analyse the labelled dataset by relating the input features to the attack that was previously trained for to classify new data.

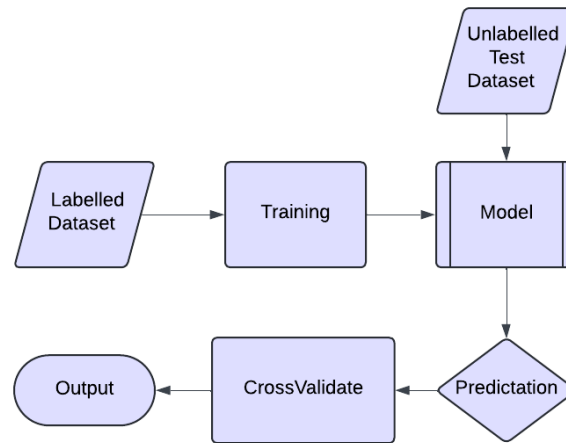


Figure 8 - Flowchart showcasing Supervised Learning in action

4.6 Exploratory Data Analysis

Exploratory data analysis was significant in the understanding of the numerical and categorical variables that were integrated into the data frame. To address the numerical features, the distribution of the histogram and the box plots were used to identify these outliers. These plots were useful in further understanding the details of certain aspects of packet structure such as the size of the packet and bytes originating from the source between normal and anomalous traffic. On the other hand, bar plots were preferred to show the distribution frequencies of the categorical features such as proto and service which helped in identifying the prevalent types of protocols and services in the given set. Finally, the EDA showed the degree of variation in both attacks and normal traffic enough to conclude that the created dataset can effectively train a strong anomaly detection model. This analysis also served to give insight into how each feature should be weighted and given significance for the best performance of the model with concern to the data it was working with.

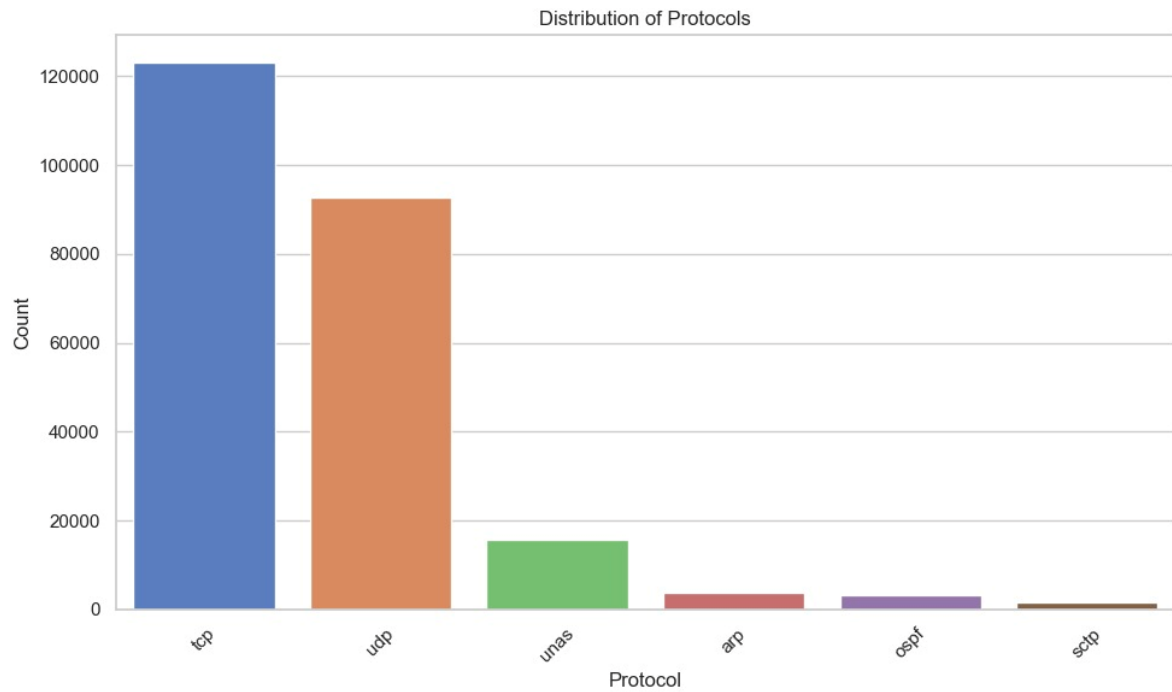


Figure 9 - Graph showcasing Protocol distribution in UNSW-NB15 dataset

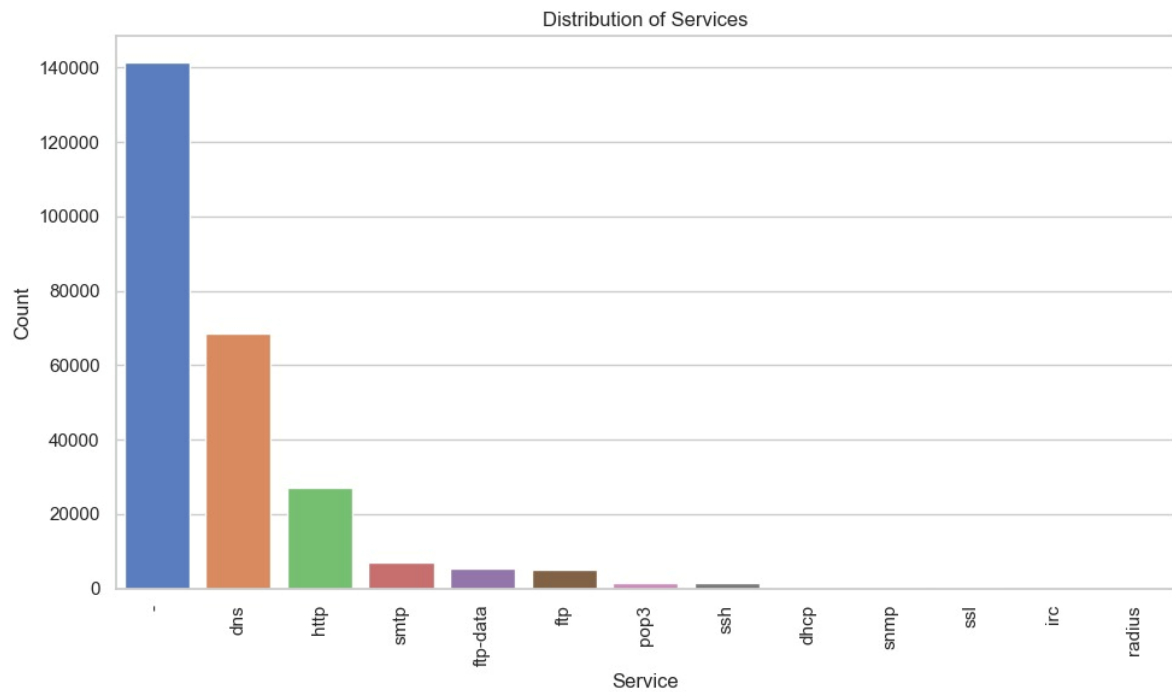


Figure 10 -Graph showcasing Services distribution in UNSW-NB15 dataset

4.7 Algorithms Planned And Used

In the context of building this AI model for network anomaly detection, various machine learning algorithms were considered in advance. These included Random Forest and the Support Vector Machines (SVM) which are used in classifying models among others. Random Forest is that the algorithm works well with large datasets that contain many attributes by constructing multiple decision trees, and the results are combined to increase the precision of the outcome (Breiman, 2001). Another approach of Classifiers is SVM which is most relevant in high dimensional space and best suited for classes with clear margins (Cortes & Vapnik, 1995). However, both algorithms have certain limitations especially when dealing with complex patterns and large data sets which are the main reasons for going for deep learning.

The neural network was used in the end due to the reason that it can mimic non-linear relationships within the data rather well. Deep learning models, for instance, have several layers, which makes it possible for the neural networks to detect even the slight variation of the normal network behaviour from the anomalous behaviour. This makes them highly suitable for the dynamic and ever-changing nature of the threats that cybersecurity must deal with (Lansky, et al., 2021).

4.7.1 Neural Network Design

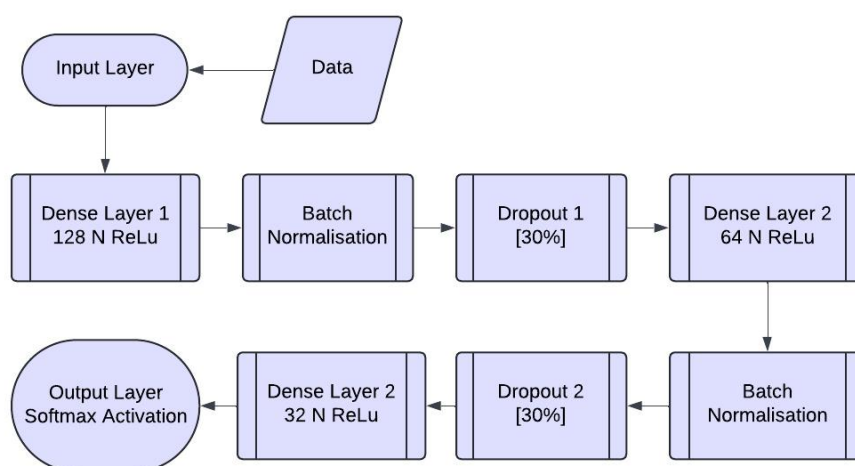


Figure 11 - Neural Network of the model

The architecture of the neural network was carefully designed to ensure that it could efficiently process and classify the vast amount of data in the UNSW-NB15 dataset (Moustafa & Slay, 2015). The key components of the network design include:

4.7.1.1. Dense Layers

Fully connected dense layers were adopted and the model incorporated more than one of such layers. The neurons in these layers are excitatory and connected to every neuron in the previous layer making it possible for the network to capture all the relations in the input data. In the above-mentioned layers, ReLU (Rectified Linear Unit) activation functions were used to increase the non-linearity of the model. ReLU is chosen because it provides computational ease and at the same time, reduces the problem of vanishing gradients which is a common occurrence in deep networks whereby gradients become too small for further training (Nair & Hinton, 2010).

4.7.1.2 Batch Normalization

Batch normalization was applied to normalize the input to the layers which enhances the speed, and stability of the learning process. During training, batch normalization normalizes the input values for the neural network, and it also pulls the model out of local minima and reduces problems associated with gradient vanishing or exploding. It also enhances convergence in a way that the model will train faster (Ioffe & Szegedy, 2015).

4.7.1.3 Dropout Layers

To avoid overfitting in the constructed neural network, the dropout layers were used. Dropout makes randomly a certain percentage of neurons in the training phase “drop” their output, that is, the output of neurons is set to zero. This is good for depriving the model of idling some neurons and thus learning the more generalized patterns in the data. Dropout is particularly helpful in complicated models; this is because overfitting typically occurs in large precisely because of its large capacity to learn data (Srivastava, et al., 2014).

4.7.1.4 Adam Optimizer

The choice of the Adam (Adaptive Moment Estimation) optimizer was applied for the training of the model. This optimization algorithm is called Adam which is common in deep learning because of its computational speed and its suitability for large data sets with small gradients. It changes the learning rate for every single weight within the weights matrix independently of the others using moment estimates of the gradients. This adaptive learning rate means that Adam converges faster and with more efficiency as compared to other optimization algorithms such as Stochastic Gradient Descent (SGD) (Kingma & Ba, 2014).

4.7.2 Cross-Validation And Early Stopping

Cross-validation is one of the techniques used for determining the performance of the machine learning model where the data is split into several subsets called folds., then the model is trained on a fold of data and validated on the other fold and this segregation is done in such a way that every record of data is used for both, training and validation at a given time. This method is preeminent for assessing the capability of a model when implemented on a new set of data and avoiding overfitting the model by training the model on different subsets of the data (Kohavi, 1995).

Some of the regularization techniques include early stopping which is helpful especially when training a model. Cross-validation: It keeps track of the validation set and stops the training process when the model's validation loss does not increase after a fixed number of epochs. It decides to stop the process of the machine's training before it memorizes learning samples fully and excessively, allowing for the best balance between the number of learned patterns and the lack of overfitting to the training data. Early stopping is also used to minimize the over-training of the model especially when the model performance is optimized already at some point during the training process (Prechelt, 2000).

The use of k-fold cross-validation was also utilized in this project alongside early stopping to minimize the risks of overfitting in the model to enhance the identification of anomalous traffic from the network traffic. This model used cross-validation and it was preferred due to the reason that it involves the separation of the dataset into two sets where the model is trained on one part and validated on the other making the model more reliable in different conditions. This prevents it from being too dependent

on a certain set of features, thus influencing it to be flexible to work on new data. On the other hand, there is early stopping which was used to prevent the model from overfitting or 'memorizing' the training data. Early stopping further contributes to striking the balance between learning more from the data than is necessary and failing to learn from other data than those the model learns from to over-specialize where the validation loss is used to stop the training process once there is little to no improvements to make within epochs. They are thereby incorporated into one model to optimize its performance as well as to make it more applicable to real-life cybersecurity where the data and threats change over time.

CHAPTER 5: AI MODEL FRAMEWORK WORKING PATTERN

This chapter outlines the AI model framework designed for network anomaly detection, focusing on its architecture, methods, and strategies for optimization. The model leverages deep learning techniques to automatically identify patterns in network traffic, aimed at improving the detection of both known and emerging cybersecurity threats. Below is a detailed breakdown of the methods used to ensure the model's robustness, adaptability, and practical application.

5.1 Model Framework Overview

For this project, the chosen AI model was a neural network model that is suitable for multi-class classification tasks (LeCun, et al., 2015). This enables the model to tell apart normal communication traffic from several forms of cyber incidents. The first two goals of the model include anomaly detection where the model is supposed to flag potential cyber threats by recognizing anomalies from the normal behaviour of the network, and adaptability since this type of model is supposed to learn from new data so that it can detect emerging threats when retrained. It is also business oriented as the model is developed for deployment in networks of different sizes.

5.2 Model Architecture And Optimization Strategies

For the current neural network model used in network anomaly detection, the architecture comprises three hidden layers with different techniques being employed both in the architecture design and the optimization process to reduce the risk of the model over-fitting. In the input layer, the features from the pre-processed features from the dataset are processed such as numerical values, which are src_bytes

and dst_bytes, and the categorical values, which are proto, service, and state. These features were encoded and scaled to fit the model to be able to facilitate analysis and comparison.

5.2.1 Architecture

5.2.1.1 Hidden Layers And Activation Functions

The first convolutional layer comprises 128 neurons with the activation type set at ReLU which stands for Rectified Linear Unit. For the same reason, ReLU was chosen because of its capability to model non-linearities in the data and is computationally efficient (Nair & Hinton, 2010). To reduce overfitting, L2 regularization was used, which constrains large weights in the dense layers and ensures the model generalizes well among the used resampling validation folds (Ng, 2004). Saying the regularization strength needed some tuning, which included a process of tweaking, give and take produced a model that did better on both the training and validation set. Batch normalization was used in this work to stabilize and accelerate training by normalizing inputs to each activation function, thus minimizing internal covariate shift (Ioffe & Szegedy, 2015). To this end, the dropout rate is 0.3 was used to make chosen neurons inactive during training so that the model does not over-depend on specific neurons (Srivastava, et al., 2014). The second hidden layer consists of 64 neurons just like the first layer and comprises elements of L2 regularization, batch normalization, and dropout that provide better patterns as compared to the first layer. The third hidden layer having 32 neurons brings out lesser features of network traffic data and is more sensitive to the variations in network traffic data. Dropout and regularization are also applied in this layer to make the model more capable of generalizing. The last layer was the output layer; using the SoftMax function will provide a probability of each sample into each attack type. This means the model can predict each sample as normal traffic or different types of attack present in the dataset.

5.2.1.2 Justification Of Architectural Choices

The choice of having multiple hidden layers with a smaller number of neurons was to extract from the features in the input data in a stepwise manner. This is mostly applied in Deep learning models which are intended for the classification of data, as they assist in learning hierarchical feature extraction

(Goodfellow & Courville, 2016). Hence, regularization techniques and different activation functions were chosen after a careful analysis to gain an optimal performance and avoid overfitting.

5.2.2 Optimisation Techniques

For fine-tuning the learning capability of the model, the Adam optimizer was chosen for its unique feature of being adaptive as well as a favourable choice for big data sets (Kingma & Ba, 2014). Adam uses the properties of AdaGrad and RMSProp, which adapt for each feature a learning rate. For this model, the learning rate is set at 0.0005 was decided based on the experience while ensuring that the below models are appropriate for gradual progression in learning without overstraining the models to learn more than they can handle.

5.2.3 Validation Techniques

To overcome issues with overtraining, both K-fold cross-validation and early stopping techniques were used to enhance the model's resilience and ability to perform on unseen data. In K-fold cross-validation, the dataset was split into five parts, training on four parts and tested on the remaining one with different iterations (Kohavi, 1995). This process was repeated 5 times so that every portion of the data was used for training and validation. It also prevents overfitting since the model learns from different splits of the data and delivers high accuracy across the subsets.

To avoid overfitting, early stopping was used whereby the model was trained until the validation loss stopped improving (Prechelt, 2000). This allowed for the model to generalize well while at the same time not overfitting the training data. The application of early stopping also played a crucial role in reducing the training time to its most effective and not performing unnecessary iterations beyond the best point.

5.2.4 Regularization Techniques

5.2.4.1 Dropout Layers

Dropout layers helped handle overfitting which occurred previously during training and especially at the beginning developmental stage of a model. When it comes to neurons, dropout disabled them randomly

and, therefore, made the model memorize common patterns instead of training data (Srivastava, et al., 2014). Finally adjusting the dropout rate to 0.3 demonstrated its utility in maintaining the best-fit capability of the model while avoiding overfitting; however, during the initial stage of the study, researchers encountered some difficulties in the identification of the appropriate rate that yielded the best result.

5.2.4.2 Batch Normalization

Batch normalization aided in retaining training stability and speed through the normalization of layer outputs and thus a faster convergence (Ioffe & Szegedy, 2015). This technique helps to decrease dependence on the initial starting weights or the value and increase the learning rates, or in other words, the training speed.

5.2.4.3 L2 Regularization

L2 regularization was used to minimize overfitting by adding a penalty term to large weights in the fully connected layers (Ng, 2004). This technique was used to prevent the model from overfitting, or in other words, to ensure that it performs well across the different validation folds. Changing the regularization strength was an iterative process, but it did help make the model more accurate since it achieved fairly similar results on both the training and validation datasets.

Together, these techniques helped produce a model that is both efficient and reliable, ensuring strong performance across different datasets.

CHAPTER 6: EVALUATIONS AND OUTCOMES

This chapter focuses on evaluating the performance of the AI model developed for network anomaly detection and presenting the outcomes of its application. The evaluation involves measuring key performance metrics such as accuracy, precision, recall, and F1-score, and analysing the effectiveness of the model across different attack categories. The outcomes highlight both the technical achievements of the model and the practical implications of these results in real-world cybersecurity scenarios.

6.1 Performance Metrics

To assess the overall performance of the AI model devised for the detection of network anomalies several essential metrics were used. They gave a clear picture of the general performance of the model and areas that could be vulnerable to attacks hence giving us more results on how to improve on the model.

6.1.1 Accuracy

For evaluating the overall percentage of instances correctly classified, including light traffic and actual malware, accuracy was employed. While accuracy gives a good idea about the performance of the model, it seems to shortfall in certain cases when there are some occurrences of imbalanced data.

6.1.2 Precision

The precision determines the accuracy of actual positive instance predictions over all the true positive cases (such as a unique form of attack). This becomes important when the false alarm rate is of importance, for example in systems where any falsified attack alert generates unnecessary reaction and workload.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Figure 12 - True and False Positives/Negatives matrix

6.1.3 Recall

Recall is the number of actual positive samples identified by the model to the total number of actual positives. Thus, recall is crucial in identifying as many actual threats as possible, especially when the attack type differs significantly, for example, worms or exploits. Any low accuracy means the model ignores some attacks hence the system will likely be at risk.

6.1.4 F1-Score

The average F1-score measures the speed of the algorithm at the balance between both false positives and false negatives as calculated from a combination of a precision rate and a particular recall rate. This metric is particularly useful when working with data sets with unequal ratios since it guarantees that the model's precision and recall metrics will be taken into consideration when assessing the output.

These metrics were chosen as they reflect the model's performance in terms of separation of the normal traffic from the different types of anomalous traffic. Whereas accuracy evaluates the overall performance, precision and recall allow for a more in-depth look when it comes to handling imbalance classes, and the F1-score is used for intermediate scenarios where precision and recall conflict. Furthermore, the confusion matrix enables a detailed analysis of specific instances, where errors were made by the model, to better understand what needs to be done to improve the performances, for instance, to better distinguish rare types of attacks.

6.2 Confusion Matrix Analysis And Classification Report

The confusion matrix is the graphical representation of the model's accuracy since it shows the correct and incorrect prediction of each class by the model. It seems optimal for enunciating where the model is going wrong, for instance, completely mistaking one form of attack for something else. The confusion matrix presents information on how samples from specific classes are distributed in other classes; hence, if misclassified, it will be easy to make the necessary adjustments. Confusion matrix is a detailed report of the model's performance through classifying and providing information about correct and incorrect classification of different attack categories. The true labels are given by each row of the matrix whereas the predicted labels are given by each column of the matrix.

The example of how the classification report works is shown in its corresponding table which presents precision, recall, and F1 score for each kind of attack. These metrics provide information about the degree of differentiation between categories of attacks and normally seen traffic.

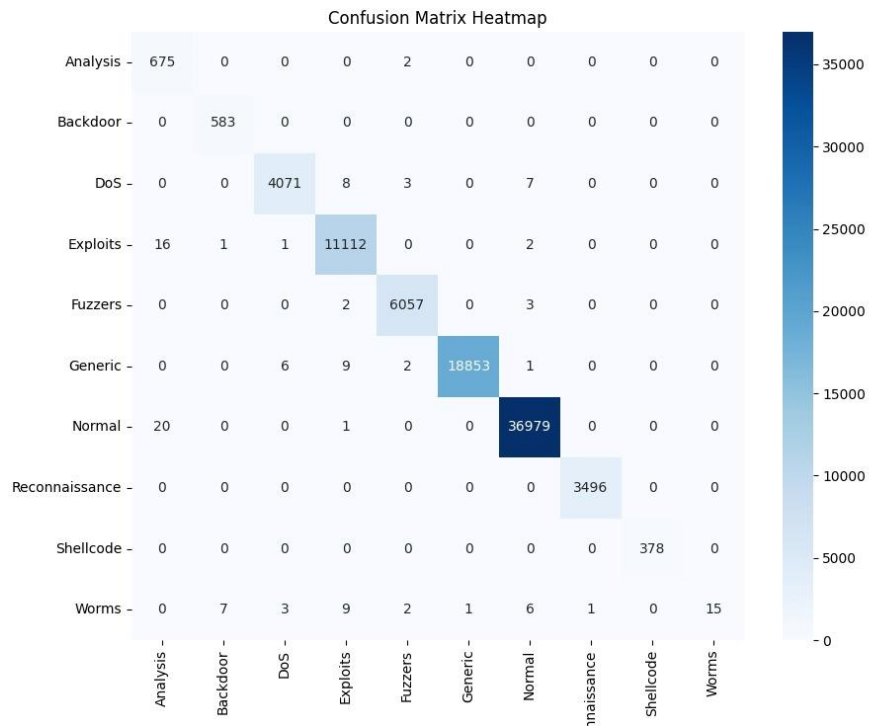


Figure 13 - Confusion Matrix of the final trained model

Category	Precision	Recall	F1-Score	Support
Analysis	0.95	1.0	0.97	677
Backdoor	0.99	1.0	0.99	583
DoS	1.0	1.0	1.0	4089
Exploits	1.0	1.0	1.0	11132
Fuzzers	1.0	1.0	1.0	6062
Generic	1.0	1.0	1.0	18871
Normal	1.0	1.0	1.0	37000
Reconnaissance	1.0	1.0	1.0	3496
Shellcode	1.0	1.0	1.0	378
Worms	1.0	0.34	0.51	44

Figure 14 - Classification Report of the Final Model

6.3 Average Cross-Validation Performance

Cross-validation was performed with a five-fold, which divides the dataset into five distinct parts. Each fold of the 5-fold CV was trained on four subsets of data and validated on the fifth one to reduce biases when evaluating the model. Because of this process, it was able to gauge how well the model performs on unseen data and at the same time avoid the model fitting for the data used to train it. With an average cross-validated accuracy of 99.89%, the model stood out as a good prediction model. with a standard error of ± 0.0003 . The results prove that the model has high robustness in different distribution conditions of training, validation, and testing data sets which guarantees reliable identification of network anomalies.

6.4 Key Insights On Model Performance And Capabilities

The model performed well for most common attack types such as DoS, Exploits, and Reconnaissance with nearly perfect accuracy in terms of precision and recall with a very low false positive and false negative rate. For less frequent kinds of attack, such as Worms and Shellcode, it provided respectable results overall but had lower recall due to the problem with class imbalance. This is a typical problem in the case of imbalanced data and is likely to improve in the future by using oversampling techniques, and data augmentation to identify relatively rare attacks (He & Garcia, 2009).

CHAPTER 7: CHALLENGES FACED

One of the major technical issues that were faced during the project was the skewed dataset problem in the UNSW-NB15 dataset (Moustafa & Slay, 2015). The distribution of attack type was also polarized; simple probable attacks like DoS and Exploits dominated, while infrequent types of attacks like Worms and Shellcode had few instances. This imbalanced distribution of the data led to the model performing very well on the frequent attacks but poorly on the less frequent ones usually with low recall values for the classes. The first steps made to handle this, for example, class-weight corrections and oversampling of minority classes, offered certain added values but failed to eliminate the problem. Other methods such as the Synthetic Minority Over-sampling Technique (SMOTE) were also employed as they have been used successfully in this type of anomaly detection (Haixiang, et al., 2016), yet they could not

solve the issue. Issues like keeping high accuracy for frequently performed attacks and decent recall for less frequent attacks remained relevant during the model development process.

Another problem was overfitting which became more pronounced as the complexity of the neural network was enhanced. This is because, during the training phase, the model is trained to identify specific features within the training data set and hence struggles to perform well on unseen data. To counter these practices such as L2 regularization, dropout layers, and batch normalization were employed where applicable (Srivastava, et al., 2014). These modifications helped stabilize the model and mitigate overfitting problems but fine tuning of parameters such as the dropout rate and the regularisation factor involved a large amount of experimentation. It has been established that the process of parameter tuning in deep learning models may involve some trial and error to arrive at the right level of model intricacy and generalization performance as established in (Goodfellow & Courville, 2016). The concept of early stopping was also used so that the model does not train beyond the stage where it starts to overfit on the validation set (Prechelt, 2000). This made the training and validation process very iterative and elongated the model tuning phase by having to frequently check the performance of the training new training models.

The second set of difficulties was less technical and was identified mainly in the aspects of time and ethical issues. Scheduling the time of the project was challenging, especially when working on periods that involved intensive testing and troubleshooting, like when data processing glitches caused delays. Scheduling is a problem that is familiar to anyone who works with AI models, particularly when the project involves many iterations of hyperparameter optimization and validation. Moreover, it was important for the project to follow ethical considerations, especially regarding the anonymity and the accuracy of the data set involved needed significant planning and attention. Privacy and security issues are important when it comes to ethical considerations of the projects in the cybersecurity domain as pointed out by (Brundage, et al., 2018) on AI ethics initiative. These challenges meant that project management had to constantly be in flux, where the requirements of the numerous tests and ethical standards had to be met alongside the limited time that was available to get the project done.

CHAPTER 8: CONCLUSION

This project has shown how Artificial Intelligence (AI) can improve the field of cybersecurity, with an emphasis on network anomaly detection. In this work, through creating and training a neural network model on the UNSW-NB15 dataset (Moustafa & Slay, 2015) to determine the possibility of deep learning approaches to detect both the known and unknown threats in the flow of traffic in a network. Another interesting feature of the AI model is the way it categorizes the attacks, for instance, DoS and Exploits that demonstrates how it can enhance conventional security frameworks that are quite challenged by advancements of even more complex cyber threats.

During this research, we have not only looked at the technical advantages of AI in cyber security but also discussed the human and ethical uses of AI in cyber security. Regarding ethical concerns mainly regarding data confidentiality and data openness, the project successfully achieved the goals that are ethical while being technical at the same time. Thus, unlike some of the prior work that might oversimplify the problem or focus on what might be seen as features, this paper demonstrates how AI can help cybersecurity teams in their efforts for threat analysis and prevention, help minimize human error, and aid the organizations in becoming more resilient to current and emerging threats.

In conclusion, the current project also stands as a reference to the real-world use of AI in aiding network anomaly detection providing a dictating potential of AI in the integration into current industry cybersecurity frameworks such as Intrusion Detection Systems (IDS), Threat detection, Security Information and Event Management (SIEM) platforms. AI evolves with information parallel to its ability to learn from different datasets showing that it will always be key to the future of cyber security with constant novelty in existing cyber threats. As has been suggested above, this work is an important contribution to the current debate regarding the use of AI in present cybersecurity and the opportunities and risks associated with such a development.

CHAPTER 9: FUTURE SCOPE AND DEVELOPMENT

Looking ahead, it is a fair proposal to consider the perspectives for developing the AI model introduced in this project as far as several up-to-date cybersecurity ideas could be incorporated further concerning its efficiency and utility. Another integration can be with Zero Trust Architecture (ZTA) where this model

can help in always checking for any activity on the network, analyse for any discrepancies, and make sure every internal as well as external user on the network is consistently vouched for. This is arguably a realization of the “never trust, always verify” principle that is an integral principle of Zero Trust (Rose, et al., 2020).

Another advantage identified in the AI model is the ability to evolve with emerging threats. In cases where new forms of attacks are being identified, the model is capable of being trained with new datasets, thus making it suitable to incorporate new patterns in the project. This is beneficial as it allows the model to be constantly updated as we know that the world of cyber security is ever evolving. In the real world, this flexibility becomes critical in terms of sustaining a strong cybersecurity posture. This is because new attacking processes are constantly designed hence a model may become ineffective after some time (Moustafa, et al., 2018). Such retraining allows the model to be trained on newer datasets and be capable of identifying newer threats and safeguarding the clients against them. The model improves the detection and analysis of security events allowing organizations to address threats faster. Also, the utilization of the model to understand anomalies in the traffic pattern of the network makes it appropriate to detect fraud in a financial system, where the system will recommend potentially fraudulent activity (Abu-Nimeh, et al., 2007).

Also, the given model could be enhanced to incorporate the concepts of Zero-Knowledge Proofs (ZKP) – it will help to add the means of information checking without disclosing the said information. It is in such areas that ZKP could help ensure that the anomaly detection model delivers on its mandate without revealing more about the internal data flow of the network or the user population (Ben-sasson, et al., 2014). In addition to these, there is a possibility of integrating the model with behavioural analysis systems to improve on insider threats, given that the model focuses on capturing the actions taken by the users that may signify some sort of threat. This would enhance other tools that are already in use today such as the Security Information and Event Management (SIEM) platforms and increase in capabilities to detect sophisticated threats such as the APTs (Wang, et al., 2004). The further connection with the automated incident response systems can enhance the possibility of not only identifying the anomalies but also the integrated response to the incidents, thereby creating shorter response time and reducing the impact of cyber threats (Cuppens & Cuppens-Boulahia, 2008). Also, the model’s malleability means that it could be applied inside cloud security architectures, the Internet

of Things, and mobile administration, where real-time surveillance of massive data sets is essential (Diro & Chilamkurti, 2017).

These advances prove that the potential of the said AI model has high prospects for widespread usage across diverse specialties of cybersecurity to guarantee continued protection of organizations against new and evolving threats without compromising data confidentiality and accuracy.

CHAPTER 10: BCS PROJECT CRITERIA & SELF-REFLECTION

This chapter reflects on how the project aligns with the BCS project criteria and provides a self-evaluation of the skills gained during the development of the AI model for network anomaly detection. Throughout the process, several key modules from my course, particularly Privacy and Security and Computational Intelligence and Machine Learning proved invaluable in shaping my understanding and guiding the technical and ethical considerations of the project.

The Privacy and Security module was crucial in reinforcing my knowledge of ethical compliance and Cybersecurity, which played a central role in this project. By applying the principles learned in this module, I ensured that the dataset was anonymized and handled by ethical standards. This understanding of data protection helped me implement secure methods during data pre-processing and throughout the project's lifecycle. Furthermore, the module's focus on cybersecure systems provided a solid theoretical foundation, which I could apply practically by enhancing traditional systems with AI-driven security models. This allowed me to bridge the gap between academic learning and practical application in a real-world cybersecurity context.

The Computational Intelligence and Machine Learning module provided the technical backbone for this project. My understanding of neural networks, deep learning, and optimization techniques was directly applied to the design and development of the AI model. The use of dropout, batch normalization, and the Adam optimizer were instrumental in improving the model's performance and preventing overfitting. Additionally, the module's focus on optimization techniques and overfitting prevention guided me in implementing early stopping and cross-validation, ensuring the model was generalized well across various datasets. These concepts were critical in ensuring that the AI model was not only accurate but also robust and adaptable to evolving network threats.

This project has deepened my understanding of how AI can be effectively implemented in cybersecurity, specifically in areas like network anomaly detection. Through the design and development of the AI model, I learned how machine learning techniques, such as neural networks, can be used to identify and classify different types of cyber threats with high accuracy. AI's ability to learn from vast datasets and adapt to new attack patterns, as demonstrated by this project, shows its potential for enhancing traditional security systems, such as Intrusion Detection Systems (IDS) and Security Information and Event Management (SIEM) tools. This experience highlighted how AI can automate threat detection, reduce false positives, and help security teams respond to emerging threats more efficiently.

Finally, this project enhanced my project management and problem-solving skills. Overcoming challenges like class imbalance and ensuring the model's scalability required effective time management and strategic decision-making. The independent nature of the project allowed me to strengthen my communication skills, particularly in articulating complex AI concepts in clear, concise ways essential for explaining technical solutions to diverse audiences in future roles. Overall, the project combined technical knowledge, ethical considerations, and project management skills, providing a well-rounded experience that prepared me for real-world cybersecurity challenges.

REFERENCES

1. Abu-Nimeh, S., Nappa, D., Wang, X. & Nair, S., 2007. A comparison of machine learning techniques for phishing detection. *ACM International Conference Proceeding Series*, Volume 269. DOI: 10.1145/1299015.1299021.
2. Aldhaferi, A., Alwahedi, F., Ferrag, M. A. & Battah, A., 2023. Deep learning for cyber threat detection in IoT networks: A review. *Internet of Things and Cyber-Physical Systems*, Volume 4. DOI: 10.1016/j.iotcps.2023.09.003.
3. Alshamrani, A., Myneni, S., Chowdhary, A. & Huang, D., 2019. A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities. *IEEE Communications Surveys & Tutorials*, Volume 21, pp. 1851-1877. DOI: 10.1109/COMST.2019.2891891.
4. Apruzzese, G. et al., 2022. The Role of Machine Learning in Cybersecurity. *Digital Threats: Research and Practice*, Volume 4. DOI: 10.1145/3545574.
5. Ben-sasson, E. et al., 2014. *Zerocash: Decentralized Anonymous Payments from Bitcoin*. DOI: 10.1109/SP.2014.36.
6. Biggio, B. & Roli, F., 2018. *Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning*. DOI: 10.1145/3243734.3264418.
7. Breiman, L., 2001. Random Forests. *Machine Learning*, Volume 45, pp. 5-32. DOI: 10.1023/A:1010950718922.
8. Brundage, M. et al., 2018. *The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation*, DOI: 10.48550/arXiv.1802.07228.
9. Buczak, A. & Guven, E., 2015. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials*, Volume 18. DOI: 10.1109/COMST.2015.2494502.
10. Cortes, C. & Vapnik, V., 1995. Support-vector networks. *Machine Learning*, Volume 20, p. 273–297. DOI: 10.1007/BF00994018.

11. Cuppens, F. & Cuppens-Boulahia, N., 2008. Modeling contextual security policies.
International Journal of Information Security, Volume 7, pp. 285-305. DOI: 10.1007/s10207-007-0051-9.
12. Cybersecurity & Infrastructure Security Agency, c., 2024. *CISA Artificial Intelligence Use Cases*. [Online]
Available at: <https://www.cisa.gov/ai/cisa-use-cases> (Accessed: 11 September 2024).
13. Department of Homeland Security, d., 2024. *Artificial Intelligence Use Case Inventory*. [Online]
Available at: https://www.dhs.gov/data/AI_inventory (Accessed: 12 September 2024).
14. Diro, A. & Chilamkurti, N., 2017. Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems*. DOI: 10.1016/j.future.2017.08.043.
15. Engstrom, D. F., Ho, D. E., Sharkey, C. M. & Cuéllar, M.-F., 2020. Government by Algorithm: Artificial Intelligence in Federal Administrative Agencies. *NYU School of Law, Public Law Research Papers*, Issue No. 20-54. DOI: 10.2139/ssrn.3551505.
16. Fier, Justin, 2017. *Detecting Metamorphic Malware with Darktrace's AI Technology*. [Online]
Available at: <https://darktrace.com/blog/how-darktraces-ai-detects-metamorphic-malware> (Accessed: 13 September 2024).
17. Garcia-Teodoro, P., Diaz-verdego, J., Macia-Fernandez, G. & Vazquez, E., 2009. Anomaly-based network intrusion detection:. *Computers & Security*, Volume 28, p. 18–28. DOI: 10.1016/j.cose.2008.08.003.
18. Goodfellow, I, Bengio Y & Courville A ., 2016 : Deep learning: The MIT Press, 800 pp, *Genetic Programming and Evolvable Machines*, Volume 19. ISBN: 0262035618. DOI: 10.1007/s10710-017-9314-z.
19. Haixiang, G. et al., 2016. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, Volume 73. DOI: 10.1016/j.eswa.2016.12.035.
20. He, H. & Garcia, E., 2009. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, Volume 21. DOI: 10.1109/TKDE.2008.239.

21. Ioffe, S. & Szegedy, C., 2015. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. DOI: 10.48550/arXiv.1502.0316.
22. Jada, I. & Mayayise, T., 2023. The impact of artificial intelligence on organisational cyber security: An outcome of a systematic literature review. *Data and Information Management*. DOI: 10.1016/j.dim.2023.100063.
23. Kaur, R., Gabrijelčič, D. & Klobučar, T., 2023. Artificial Intelligence for Cybersecurity: Literature Review and Future Research Directions. *Information Fusion*, Volume 97. DOI: 10.1016/j.inffus.2023.101804.
24. Kingma, D. & Ba, J., 2014. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*. DOI: 10.48550/arXiv.1412.6980.
25. Kohavi, R., 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *IJCAI'95: Proceedings of the 14th international joint conference on Artificial intelligence*, Volume 2, pp. 1137 - 1143. DOI: 10.5555/1643031.1643047.
26. Lansky, J. et al., 2021. Deep Learning-Based Intrusion Detection Systems: A Systematic Review. *IEEE Access*, Volume 9. DOI: 10.1109/ACCESS.2021.3097247.
27. LeCun, Y., Bengio, Y. & Hinton, G., 2015. Deep Learning. *Nature*, Volume 521. DOI: 10.1038/nature14539.
28. Mathews, M. S. M., Shetty, S. & McKenzie, R., 2007. *Detecting Compromised Nodes in Wireless Sensor Networks*, ISBN: 978-0-7695-2909-7. DOI: 10.1109/SNPD.2007.538.
29. Moustafa, N. & Slay, J., 2015. *UNSW-NB15: a comprehensive data set for network intrusion detection (UNSW-NB15 network data set)*. DOI: 10.1109/MilCIS.2015.7348942.
30. Moustafa, N., Turnbull, B. & Choo, K.-K. R., 2018. An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things. *IEEE Internet of Things Journal*, Volume PP. DOI: 10.1109/JIOT.2018.2871719.
31. Nair, V. & Hinton, G., 2010. *Rectified Linear Units Improve Restricted Boltzmann Machines*. s.l., Proceedings of ICML. DOI: 10.5555/3104322.3104425.
32. National Cyber Security Centre, n., 2023. *NCSC Annual Review*. [Online] Available at: www.ncsc.gov.uk/collection/annual-review-2023/technology/case-study-cyber-security-ai (Accessed: 12 September 2024).

33. Nazmul Alam, M. et al., 2020. Phishing Attacks Detection using Machine Learning Approach. *Third International Conference on Smart Systems and Inventive Technology*. DOI: 10.1109/ICSSIT48917.2020.9214225.
34. Ng, A., 2004. Feature selection, L 1 vs. L 2 regularization, and rotational invariance. *Proceedings of the Twenty-First International Conference on Machine Learning*. DOI: 10.1145/1015330.1015435.
35. Prechelt, L., 2000. *Early Stopping - But When?* ISBN: 978-3-540-65311-0. DOI: 10.1007/3-540-49430-8_3.
36. Rose, S., Borchert, O., Mitchell, S. & Connelly, S., 2020. *Zero Trust Architecture*. DOI: 10.6028/NIST.SP.800-207.
37. Saied, M., Guirguis, S. & Madbouly, M., 2024. Review of artificial intelligence for enhancing intrusion detection in the internet of things. *Engineering Applications of Artificial Intelligence*. DOI: 10.1016/j.engappai.2023.107231.
38. Samunnisa, K., Gaddam, S. & Madhavi, K., 2022. Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods. *Measurement: Sensors*, Volume 25. DOI: 10.1016/j.measen.2022.100612.
39. Sharafaldin, I., Habibi Lashkari, A. & Ghorbani, A., 2018 . *Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization*. s.l.:s.n. DOI: 10.5220/0006639801080116
40. Srivastava, N., Hinton, G., Krizhevsky, A. a. S. I. & Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, Volume 15. DOI: 10.5555/2627435.2670313.
41. Wang, W., Guan, X. & Zhang, X., 2004. *A Novel Intrusion Detection Method Based on Principle Component Analysis in Computer Security*, ISBN: 978-3-540-22843-1. DOI: 10.1007/978-3-540-28648-6_105.
42. Watson, R., Bergman, R. & Ciepiela, P., 2024. *How can cybersecurity transform to accelerate value from AI?*. [Online]
Available at: https://www.ey.com/en_no/consulting/transform-cybersecurity-to-accelerate-value-from-ai (Accessed: 14 September 2024).

43. Xin, Y. et al., 2018. Machine Learning and Deep Learning Methods for Cybersecurity. *IEEE Access*, Volume PP. DOI: 10.1109/ACCESS.2018.2836950.
44. Zhang, Z. et al., 2022. *Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research*. s.l.:s.n. DOI: 10.48550/arXiv.2208.14937.

APPENDIX

```
import pandas as pd # Importing the pandas library for data manipulation and analysis
import numpy as np # Importing the numpy library for numerical operations and array manipulation
from sklearn.model_selection import train_test_split # Importing train_test_split for cross-validation splitting
from sklearn.preprocessing import StandardScaler, OneHotEncoder, LabelEncoder # Importing preprocessing tools: StandardScaler for scaling features, OneHotEncoder for one-hot encoding categorical variables, and LabelEncoder for encoding labels
from sklearn.compose import ColumnTransformer # Importing ColumnTransformer to apply different preprocessing steps to different columns
from tensorflow.keras import layers, models, optimizers, regularizers # Importing Keras modules for building and training neural networks
from tensorflow.keras.callbacks import EarlyStopping # Importing EarlyStopping to stop training early if the validation loss stops improving
from sklearn.metrics import classification_report, confusion_matrix # Importing metrics for evaluating the performance of the model
from scipy.sparse import csr_matrix # Importing csr_matrix to handle sparse matrices (used in feature transformation)
import joblib # Importing joblib for saving and loading the preprocessor and model objects
```

Figure 15 - Code Snippet of Python packages used

```
# Load the datasets
train_data = pd.read_csv('UNSW_NB15_training-set.csv') # Load training dataset
test_data = pd.read_csv('UNSW_NB15_testing-set.csv') # Load testing dataset

# Drop unnecessary columns (id) and encode attack categories
X_train = train_data.drop(columns=['id', 'label']) # Drop 'id' and 'label' columns from training data
y_train = train_data['attack_cat'] # Extract the 'attack_cat' column as labels for training

X_test = test_data.drop(columns=['id', 'label']) # Drop 'id' and 'label' columns from testing data
y_test = test_data['attack_cat'] # Extract the 'attack_cat' column as labels for testing

# Encode attack categories as integers
label_encoder = LabelEncoder() # Initialize LabelEncoder to convert categorical labels into integers
y_train = label_encoder.fit_transform(y_train) # Fit and transform the labels for the training data
y_test = label_encoder.transform(y_test) # Transform the labels for the testing data using the same encoder

# One-hot encode the labels for multi-class classification
y_train = np.eye(len(label_encoder.classes_))[y_train] # Convert integer labels to one-hot encoded format for training data
y_test = np.eye(len(label_encoder.classes_))[y_test] # Convert integer labels to one-hot encoded format for testing data

# Identify categorical and numerical features
categorical_features = ['proto', 'service', 'state'] # Specify categorical features that need encoding

# Ensure that no numeric feature has non-numeric data
numeric_features = [] # Initialize a list to hold numeric features
for col in X_train.columns:
    if col not in categorical_features: # Check if the column is not categorical
        if X_train[col].dtype == 'object': # If the column has non-numeric data
            categorical_features.append(col) # Add it to the list of categorical features
        else:
            numeric_features.append(col) # Otherwise, add it to the list of numeric features
```

Figure 16 - - Code Snippet of loading and handling data

```
# Preprocessing pipeline
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numeric_features), # Standardize numeric features
        ('cat', OneHotEncoder(handle_unknown='ignore', sparse_output=False), categorical_features) # One-hot encode categorical features
    ])

# Apply preprocessing
X_train_preprocessed = preprocessor.fit_transform(X_train) # Fit and transform the training data
X_test_preprocessed = preprocessor.transform(X_test) # Transform the testing data using the same preprocessor

# Convert the sparse matrix to a dense matrix if not done already
if isinstance(X_train_preprocessed, csr_matrix):
    X_train_preprocessed = X_train_preprocessed.toarray() # Convert training data to a dense matrix if it's sparse

if isinstance(X_test_preprocessed, csr_matrix):
    X_test_preprocessed = X_test_preprocessed.toarray() # Convert testing data to a dense matrix if it's sparse

# Verify preprocessing
print(f"Processed training data shape: {X_train_preprocessed.shape}")
print(f"Processed testing data shape: {X_test_preprocessed.shape}")

# Save the preprocessor
joblib.dump(preprocessor, 'preprocessor_anm.joblib') # Save the preprocessor object to a file for later use
print('Preprocessor saved as preprocessor_anm.joblib')
```

Figure 17 - Code Snippet of Preprocessing data and Data pipeline

```
# Define a function to create the model
def create_model():
    model = models.Sequential() # Initialize a Sequential model
    model.add(layers.Dense(128, activation='relu', input_shape=(X_train_preprocessed.shape[1],),
                           kernel_regularizer=regularizers.l2(0.001))) # Add a dense layer with L2 regularization
    model.add(layers.BatchNormalization()) # Add batch normalization to stabilize and speed up training
    model.add(layers.Dropout(0.3)) # Add dropout to prevent overfitting
    model.add(layers.Dense(64, activation='relu', kernel_regularizer=regularizers.l2(0.001))) # Add another dense layer with L2 regularization
    model.add(layers.BatchNormalization()) # Add batch normalization
    model.add(layers.Dropout(0.3)) # Add dropout
    model.add(layers.Dense(32, activation='relu', kernel_regularizer=regularizers.l2(0.001))) # Add a third dense layer
    model.add(layers.Dense(len(label_encoder.classes_), activation='softmax')) # Output layer with softmax for multi-class classification

    optimizer = optimizers.Adam(learning_rate=0.0005) # Initialize Adam optimizer with a learning rate of 0.0005
    model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy']) # Compile the model with categorical crossentropy loss
    return model # Return the compiled model
```

Figure 18 - Code Snippet of creating the model

```
# Implement Early Stopping
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True) # Initialize early stopping to prevent overfitting

# Cross-Validation
kf = KFold(n_splits=5, shuffle=True, random_state=42) # Set up 5-fold cross-validation with shuffling

cv_scores = [] # List to store cross-validation scores
fold = 1 # Counter for folds
best_val_accuracy = 0.0 # Track the best validation accuracy
best_model = None # Placeholder for the best model

for train_index, val_index in kf.split(X_train_preprocessed):
    print(f"\nTraining fold {fold}...")
    fold += 1

    X_train_fold, X_val_fold = X_train_preprocessed[train_index], X_train_preprocessed[val_index] # Split data into training and validation sets for this fold
    y_train_fold, y_val_fold = y_train[train_index], y_train[val_index] # Split labels into training and validation sets

    # Create a new model instance for each fold
    model = create_model()

    # Train the model
    history = model.fit(X_train_fold, y_train_fold,
                       epochs=50, # Set the number of epochs
                       batch_size=128, # Set the batch size
                       validation_data=(X_val_fold, y_val_fold), # Set validation data for this fold
                       callbacks=[early_stopping], # Use early stopping to avoid overfitting
                       verbose=1) # Print progress during training

    # Evaluate the model on the validation set
    val_loss, val_accuracy = model.evaluate(X_val_fold, y_val_fold, verbose=0) # Evaluate on the validation set
    cv_scores.append(val_accuracy) # Store the validation accuracy
    print(f"Validation accuracy for this fold: {val_accuracy:.4f}")

    # Save the best model during cross-validation
    if val_accuracy > best_val_accuracy:
        best_val_accuracy = val_accuracy # Update the best validation accuracy
        best_model = model # Keep the reference to the best model

# Save the best model during cross-validation
if best_model is not None:
    best_model.save('best_cross_validated_model_ana.keras') # Save the best model found during cross-validation
    print(f"Best model saved with validation accuracy: {best_val_accuracy:.4f}")

# Report the average cross-validated accuracy
average_cv_accuracy = np.mean(cv_scores) # Calculate the mean accuracy across all folds
std_cv_accuracy = np.std(cv_scores) # Calculate the standard deviation of the accuracy
print(f"\nAverage cross-validated accuracy: {average_cv_accuracy:.4f} ± {std_cv_accuracy:.4f}")
```

Figure 19 - Code Snippet of Fine tuning the model