A DEEP REINFORCEMENT LEARNING APPROACH TO MODELLING AN INTRUSION DETECTION SYSTEM USING ASYNCHRONOUS ADVANTAGE ACTOR-CRITIC ALGORITHM

BY

JUNIOR KIPLIMO YEGO

A THESIS SUBMITTED TO THE SCHOOL OF INFORMATION SCIENCES, DEPARTMENT OF INFORMATION TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

MOI UNIVERSITY

2023

DECLARATION

Declaration by the Candidate

This thesis is my original work and has not been presented for a degree in any other university. No part of this thesis may be reproduced without the prior written permission of the author and/or Moi University.

Sign:_____ Date:_____

Yego Junior Kiplimo

MS/IT/4514/20

Declaration by Supervisors

This thesis has been submitted for examination with our approval as University Supervisors.

Sign:_____ Date:_____

Dr Nicholas Kiget

Moi University, Eldoret, Kenya

Sign:_____ Date:_____

Dr Irene Moseti

Moi University, Eldoret, Kenya

DEDICATION

This thesis is dedicated to my parents Mr & Mrs Isaac Mettoh, my brothers and sisters.

ACKNOWLEDGEMENT

I thank God for his Unfailing Mercies and Grace. Throughout My study, I have enjoyed Good Health, Sound Mind and Peace. Glory and Honor to God. I would like to thank my Supervisors Dr Nicholas Kiget and Dr Irene Moseti for their support and guidance throughout my study and thesis development. Their wide expertise has provided solutions to the questions about my research and they have also provided valuable comments on the thesis. God bless you. I would also like to thank Mr Daniel Samoei for his expert advice and comments and for steering me in the right direction during the conceptualization of the research. Many thanks also to Mr Bor Dennis for his valuable contribution to the model's development. Finally, I must express my very profound gratitude to my family. They have been consistent in supporting and encouraging me throughout my study and thesis writing. This accomplishment would not have been possible without them. Thank you.

ABSTRACT

With the increasing development and use of the internet, cyber-attacks have evolved and more novel attacks with devastating effects are witnessed. The existing Intrusion Detection System (IDS) has not achieved maximum performance due to high false positives and low detection rates which causes low detection accuracies. The aim of the study was to determine the effectiveness of IDS by using the Asynchronous Advantage Actor-Critic (A3C) algorithm to address the current shortcomings. The objectives of the study were: To determine the effectiveness of using the Asynchronous Advantage Actor-Critic algorithm in anomaly detection; To develop an Intrusion Detection System model, based on Asynchronous Advantage Actor-Critic (A3C) Algorithm; To evaluate the performance of the model developed. The theoretical framework adopted was informed by Computational Learning and Machine Learning theories. The study used a quantitative research approach and experimental research design. The secondary data used for evaluation in this study was the University of New South Wales Network Based 2015 (UNSW-NB15) dataset which was purposively selected as it is a well-established benchmark network intrusion simulation dataset. The dataset contained the UNSW-NB15_TRAIN and UNSW-NB15_TEST sets which were selected and utilized in the study. The records selected were 175,341 records to form the training subset and 82,332 records for the testing subset among the original 2,218,761 records. The UNSW-NB15 dataset was preprocessed to ensure quality results and all features of the dataset were used in the experiment. The method employed in analysis for this study was by using predictive analytics where the model's prediction ability was evaluated and hence the performance rated. The results of this study showed that the capabilities of the A3C algorithm in intrusion detection could perform better as seen in other fields like robotics in automation. From the experiment, the model achieved an accuracy of 93.8%, precision of 92.2% and recall of 95.7% with the compute resource use being average. The experiments showed that the agents quickly learned the optimal policy and maintains the policy until the end of the experiment. The study concludes by pointing out that with the accuracy attained, the learning capabilities of the model can still be increased by fine-tuning it so that it discovers new policies quickly as this is essential to attaining a higher accuracy rate. A recommendation made based on the study was that A3C can be adopted in intrusion detection because of the accuracy of detection and low resource consumption. More research can be done on the adoption of A3C in IDSs by using more training data as this could further improve the model performance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

A2C	Advantage Actor-Critic
A3C	Asynchronous Advantage Actor-Critic
APCA	Advanced Principal Component Algorithm
CSV	Comma Separated Values
DDoS	Distributed Denial of Service
DNN	Deep Neural Network
DOS	Denial of Service
DT	Decision Tree
DTC	Decision Tree Classifier
DR	Detection Rate
DRL	Deep Reinforcement Learning
ETC	Extra Tree Classifier
FMI	Flexible Mutual Information
FP	False Positives
FPR	False Positive Rate
FAR	False Alarm Rate
FM	Fowlkes-Mallows Index
FS	Feature Selection
GA	Genetic Algorithm
GBC	Gradient Boost Classifier
GPU	Graphics Processing Unit
HIDS	Host-based Intrusion Detection System
HMM	Hierarchical Gaussian Mixture Model
IDS	Intrusion Detection System
IDE	Integrated Development Environment
IDES	Intrusion Detection Expert System
IELM	Incremental approach of the Extreme Learning Machine
IoT	Internet of Things
MLP	Multi-Layer Perceptron
NIDS	Network-based Intrusion Detection System
NMF	Non-negative Matrix Factorization
NSA	Negative Selection Algorithm
NSA	National Security Agency
NSL-KDD	Network Security Laboratory – Knowledge Discovery in Databases
PCAP	Packet Capture
RFC	Random Forest Classifier
RL	Reinforcement Learning
RNN	Recurrent Neural Networks
SGD	Stochastic Gradient Descent
SOC	System on Chip
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TD	Temporal Difference
U2R	User to Root
UDP	User Datagram Protocol
UNSW-NB15	University of New South Wales Network Based 2015

CHAPTER ONE

GENERAL INTRODUCTION

1.1 Introduction

This chapter gives an overview of the study and focusses on the background of the study, statement of the problem, aim of the study, objectives, the research questions, justification for the study, significance of the study and operational definition of terms.

1.2 Background of the Study

With increased internet use both in online businesses and also with the wide application of Internet of Things technology and big data technology, an exponential rise in internet activities has been witnessed. This increase in internet use has also led to increased internet-related risks and threats as information is now the target of attacks by malicious hackers. With every vulnerability is the threat of a new attack. A CyberEdge Group (2021) reported 86.2% of organizations that were surveyed were affected by a successful cyber-attack. This number has been on the rise yearly as shown in figure 1.

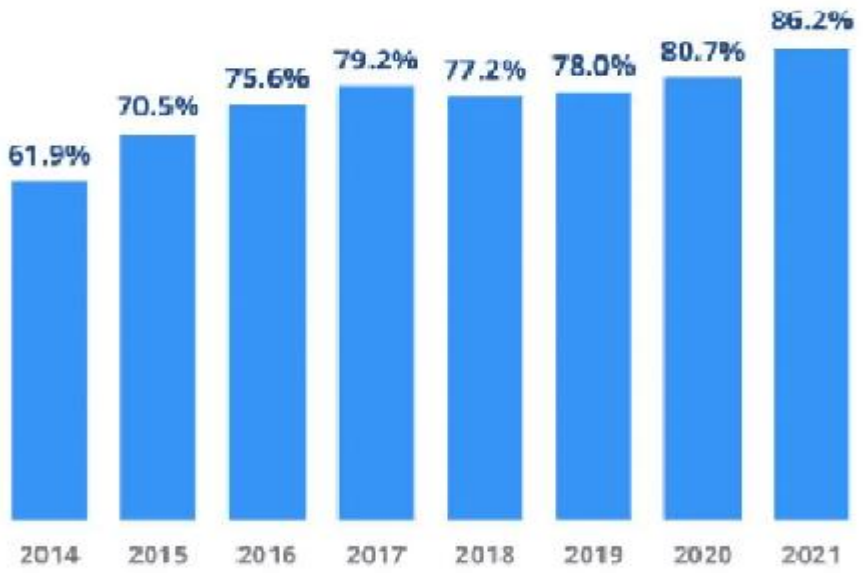


Figure 1: Percentage of Organizations compromised by at least 1 attack
Source: CyberEdge Group

This has led to the development of protective measures like the Intrusion Detection System (IDS) which is elemental in ensuring the security of computer and network systems. There are many risks to network security including intrusions by infiltration from within the network, brute force attacks, and denial of service attacks and with the continuous changes in network behavior, it is important to apply a dynamic approach in the detection and prevention of such.

An anomaly is an isolated behavior in data, whose pattern does not conform to normal behavior. They can be point, which is grouped as single data, they can also be contextual which is based on context, and lastly, collective which is based on a collection of data relationships. Zimek et al. (2017) define anomaly detection, which is also called outlier detection as the process of identifying rarely occurring events or occurrences which are also suspicious in nature because they are inconsistent by a good margin from the rest of the data. IDS on the other hand analyze data traffic and detect behavioral anomalies which pose threats to the network including malicious activity and policy violations by both system insiders and external intruders.

The ideation and conceptualization of IDS was documented by Anderson (1980) at the National Security Agency (NSA) and included tools to assist the security administrators to review audit trails like logs for user access, file access, and system events.

Denning (1986) published an IDS model which is the basis used by many systems. It utilized anomaly detection statistics and was named Intrusion Detection Expert System (IDES). This system was run using Sun workstations and the data that was used was both at user and network-level. The system also had an expert system component that was rule-based to identify common types of intrusion and statistical anomaly detection using user profiles and host and target systems. Teresa (1993) proposed including Artificial Neural Network as an additional component so that all the three components could then be managed by a resolver.

Cohen (1985) noted that it is not possible for an intrusion to be detected all the time. He also noted that intrusion detection resources increase as usage increases. A proposition by Viegas et al. (2017) was for an anomaly-based IDS this targeted application System-on-Chip (SOC) like in the Internet of Things (IoT). Machine learning is applied for the detection of anomalies hence energy efficiency to a decision tree, naïve Bayes, and k-Nearest Neighbors classifiers implementation in an Atom Central Processing Unit (CPU) including that it is hardware friendly to Field Programmable Gate Arrays. This work enabled the measurement of energy consumed for each feature extraction to classify network packets applied in both hardware and software.

For an IDS to be considered effective in terms of performance, it should be accurate in detecting intrusions by having a high level of accuracy in classification and

achieving a low false alarm rate (Endorf et al., 2019). Therefore, when designing an IDS, one of the crucial tasks should be working towards increasing the accuracy in anomaly detection as well as ensuring a reduction in False-Positive Rates (FPR).

Both the supervised and unsupervised techniques of machine learning have been successful in anomaly detection to some point but have been let down by weaknesses as further pointed by Endorf et al. (2019). For Unsupervised learning, due to the lack of existing classified labels, it assumes that normal data is defined by the majority of the data and anomalies are the smaller isolated data. This presents a false picture for data that is high-dimensional because it is sparse by nature and so could have some groups of the data which are sparse and may not be anomalies hence leading to false alarms. Supervised methods on the other hand have normal data definitions and those of known anomalies and its main weakness is that they cannot detect novel anomalies.

Deep Reinforcement Learning (DRL) approach applies the principles of reinforcement and deep learning to be able to create efficient algorithms capable of scaling to previously unsolvable problems by making use of its ability to learn from raw data as input (Abbeel & Schulman, 2021). This is good as it does not require large training data which is tiring and time-consuming work. DRL makes use of a reward function to optimize future rewards which makes it advantageous over other forms like supervised learning. Developing intrusion detection systems using the DRL approach may help solve the shortcomings that are being experienced in the current sintrusion detection systems.

1.3 Statement of the Problem

The IDSs have been used in monitoring systems for suspicious activities that result in policy violations. These systems give an alert so that the activities are contained and

do not cause further damage. The current IDS is useful in identifying anomaly behaviors in a network and guarding against breaches. However, it is still hindered from achieving maximum performance by the false positives reported where a normal activity in the system is classified as an anomaly. It is also hindered by the overall low detections that are caused by huge amounts of data in the networks. The unbalanced distribution of anomaly and normal behaviors also leads to lower accuracies of the intrusion detection systems as supported by Jiadong et al. (2019). This becomes a challenge for the Security Administrators as they need to go through every alert generated by the IDS for action. With these, the system becomes human dependent and hence all alerts must be reviewed.

Signature-based IDS also have difficulty in detecting new attacks in the network since they rely mainly on the library of stored data while anomaly-based, despite having excellent recognition ability, they have a low overall detection rate with false positives. There is, therefore, a need for a more thorough, accurate, and autonomous Intrusion detection System that will strengthen the weaknesses of the current IDSs. This study, therefore, responds to this need by modelling an IDS using Asynchronous Advantage Actor-Critic (A3C) algorithm in an attempt to strengthen the weak areas and address the current needs.

1.4 Aim of the Study

This research aimed to determine how effective the IDS is when the A3C algorithm is used by developing an IDS using A3C and evaluating its performance so as to address the shortcomings currently being experienced.

1.5 Objectives

The objectives of the study were the following:

i. To determine the effectiveness of using the Asynchronous Advantage Actor-Critic algorithm in anomaly detection
ii. To develop an Intrusion Detection System model, based on Asynchronous Advantage Actor-Critic Algorithm
iii. To evaluate the performance of the model developed

1.6 Research Questions

i. How effective is the A3C Algorithm in intrusion detection?
ii. How will the development of the IDS model using the A3C algorithm be achieved?
iii. How is the performance of the IDS model developed using the A3C algorithm?

1.7 Justification

This study sought to improve the accuracy in the detection of anomalous network behavior which will lead to a great reduction of false positives currently experienced by the IDS and hence improve the overall anomaly detection rate. This is because the intrusion detection systems that are currently being used are faced with challenges of false positives in which a legitimate activity is flagged down as an anomaly due to the unbalanced distribution of the anomalous and normal behaviors and high data traffic which in turn brings the overall problem of low detection.

1.8 Significance of the Study

This study was necessary as it sought to establish a foundation using the analysis results to draw a conclusion. This conclusion will then be added to the knowledge

repository which will be built upon by other studies in developing more effective Intrusion Detection Systems to improve network security. This is because more devices are connected and activities are carried out over the internet. The model developed also can be used in enhancing intrusion detection systems and this will enable a decrease in the malicious activities being carried out by attackers. Anomalies will be flagged down with ease and information security mechanisms will be more effective and autonomous and this will reinforce the work of security administrators.

1.9 Scope of the Study

This study focused on an overview of IDS and their importance in cybersecurity including the different types of IDS. An introduction to the A3C algorithm, which is a reinforcement learning algorithm commonly used for training agents in complex environments. It also included a description of the design of the IDS model using the A3C algorithm and the selection and preparation of a suitable dataset for training and evaluation was also discussed. Also, the process of training the IDS model using the A3C algorithm and the evaluation metrics used to assess the model's performance, such as accuracy, precision, recall, and F1 score was covered. A presentation of the results obtained from training and evaluating the IDS model and a discussion of the performance of the model while comparing the results with existing IDS approaches to evaluate the effectiveness of the A3C-based model was finally done

1.10 Limitations of the Study

Because of time and financial limitations, the study limited itself to model development only as proof of concept. The implementation and integration of IDS with the existing infrastructure was not handled in this study. Other limitations were the high resource requirements especially the compute power which limits the scale at which the experiments could be conducted.

1.11 Operational Definition of Terms

1. Data Preprocessing – Process of manipulation or dropping of data before it can be used to ensure and enhance performance.

2. Deep Reinforcement Learning - A subfield of machine learning that combines reinforcement learning algorithms with deep neural networks. It involves training agents to learn optimal actions in an environment by interacting with it and receiving feedback in the form of rewards or punishments.

3. Intrusion Detection System - A security technology designed to monitor network traffic or system events and identify potential intrusions or security breaches.

4. Machine Learning - A subfield of artificial intelligence (AI) that focuses on the development of algorithms and models that allow computers to learn from and make predictions or decisions based on data, without being explicitly programmed.

5. Training data - This data is used in the training of the algorithm or the machine learning model for the prediction of the outcome designed by the model for prediction.

6. Test data – This is the data that is applied in performance measuring, The metrics measured include accuracy, precision, and recall of the algorithm used in training the machine.

1.12 Chapter Summary

This chapter introduced the study by looking at what the research is about, what motivated the researcher to undertake this study and what the study intends to achieve by stating the research questions that will guide the researcher. The terms used in the study have also been defined. The chapter points to the need for a better IDS that can address the main problems of false positives and overall improvement of the detection accuracy. The next chapter looks at the literature review.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter details the review of the literature on the contextual concerns of the study based on the research questions which were: To determine the effectiveness of using Asynchronous Advantage Actor-Critic algorithm in anomaly detection, to develop an Intrusion Detection System model based on Asynchronous Advantage Actor-Critic Algorithm and to validate the performance the model developed. The chapter looks at the background of intrusion Detection Systems and also focusses on the theoretical framework that guides the study. Finally, a discussion the current state of knowledge in the area of study, the conceptual model and validation methods are discussed in this chapter.

2.2 Cybercrime

The main challenge being experienced in this information age is unauthorized access to information which can then be used to harm the owner of the information or control the network security systems to perform illicit actions. Most of the sensitive data illegally accessed are then used by the hackers for their gain like being sold or being used to perform even more serious malicious attacks. This has caused many organizations to ensure the safety of their information and resources from unauthorized access.

Like street crime, with the number of both human and digital targets increasing, cybercrime is set to proportionally increase. This is due to an increase in connection technologies which include the cloud centre data traffic which is increasing exponentially and confirmed by CISCO to represent over 95% of data centre traffic. Another technology is the big data technology fueled by IoT which is the wireless

communication by smart devices and CISCO (2022) projects a growth in number of nodes that are being added to the IP networks to 29.9 billion devices by 2023.

Cybersecurity Ventures (2020) predicts that the damages for committed cybercrime globally will reach $ 6 trillion by 2021. Cyber-attacks continually grow in sophistication, cost, and size. Among the biggest breaches suffered by companies was by Yahoo (2017) which was calculated to have affected 3 billion accounts. Another is Marriot (2018), disclosed in 2018 where 500 million accounts were exposed. There is also the Equifax (2017) reported breach which occurred in 2017 where over 145 million customers were affected.

Apart from unauthorized access to information, other forms of cybercrime include the Distributed Denial of Service (DDoS) attacks where the resources of a targeted system-usually the web server, are flooded by multiple systems and hence making the resource unavailable to its intended users. There is also the zero-day exploit which occurs when a computer software vulnerability is discovered and exploited by the attackers but unknown to the creators until it is mitigated. The ransomware attack is where computers are infected by malware which restricts their access to files by encryption and threatens permanent destruction of the data unless a ransom is paid. They are more complex and are larger in scale with the major examples being the Wannacry and the NotPetya attacks which occurred in 2017.

2.3 Theoretical Framework

A theoretical framework includes proven theories shown by experts in specific areas of research. Swanson (2013) states that the theoretical framework is the structure that can hold or support a theory of a research study. This enables the researcher to have a

supporting foundation that is used to analyze and interpret data, and connect them to existing knowledge while specifying key variables.

These theories studied and applied in this study by the researcher were related to the topic of this study and contributed to informing and guiding the study.

2.3.1 Computational Learning Theory

Computational learning theory is a subfield of artificial intelligence and machine learning that focuses on the theoretical analysis of learning algorithms. This theory is a collective effort from researchers' contributions over the years. Among the most notable contribution to the foundation of this theory is Valiant (1984). His work lays the foundation for understanding the computational complexity of learning problems and provides insights into the theoretical limits and capabilities of learning algorithms. This is supported by Shalev & Ben (2014) who note that Computational Learning Theory is practical and studies the design of computer programs which have the capability to learn, and identifies the computational limits of learning by machines. While the researchers have historically compared learning algorithms empirically by looking at how they perform on sample problems, the problem has been applying the evaluation results in making useful comparisons on the competing learning algorithms (Kononenko & Kukar, 2009). Machine learning has become an integral part of numerous applications, ranging from recommendation systems to image recognition. Kononenko & Kukar (2009) further state that computational learning theory provides a formal framework to analyze the behavior and performance of these learning algorithms, facilitating a deeper understanding of their capabilities and limitations.

Computational learning plays a crucial role in training and optimizing the A3C algorithm, enhancing its performance and enabling efficient learning in complex environments. The application areas of computational learning techniques on the A3C algorithm include:

1. **Neural Network Architecture:** This is a key component of the A3C algorithm and represents the policy and value function. Computational learning techniques, such as deep learning, are utilized in the design and training of the neural network. Convolutional neural networks or recurrent neural networks are commonly used to process the input states and produce the predictions on policy and value (Mnih. V et al., 2016). The parameters of the neural network are updated iteratively using optimization algorithms like stochastic gradient descent (SGD) or variants such as Adam or RMSprop.
2. **Gradient-Based Optimization:** Computational learning techniques heavily rely on gradient-based optimization methods to update the neural network parameters. In A3C, the gradients are computed using the policy gradient theorem or actor-critic methods. The gradients are then used to update the neural network parameters, improving the policy and value estimates (Silver et al., 2014).
3. **Asynchronous Training:** A3C employs asynchronous training to parallelize the learning process, enabling multiple agents to explore the environment concurrently. Computational learning techniques are applied in managing the asynchronous updates of the neural network parameters. The independent agent interacts with the environment collecting experiences and updating the gradients asynchronously. Shared memory and parameter servers are the techniques used for update synchronization and ensuring that there is

consistency across agents. This parallelization accelerates the learning process and enables the exploration of the environment to be efficient.

4. **Exploration and Exploitation:** Computational learning techniques are used to balance the trade-off between exploration and exploitation in the A3C algorithm. Techniques such as epsilon-greedy exploration or Boltzmann exploration can be utilized to make sure that the agents explore new actions and states. This allows for a more in-depth understanding of the environment. Exploration strategies are very important in reinforcement learning, and computational learning provides tools to design and implement effective exploration policies as supported by Sutton & Barto (2018).
5. **Experience Replay:** This is a technique commonly used in deep reinforcement learning and can be combined with the A3C algorithm. Experience replay buffers are used to store and sample experiences (Mnih et al., 2015). This allows the agent to interact and learn from those interactions. Computational learning techniques facilitate the efficient storage and retrieval of experiences from the replay buffer thus enabling the agent to learn from a diverse set of experiences and reduce the impact of correlations between samples.
6. **Transfer Learning and Pre-training:** These computational learning techniques can be applied to enhance the performance of the A3C algorithm. Yosinski et al. (2014) posits that pre-training the neural network on a related task or using transfer learning from a pre-trained model can provide a good initialization and acceleration of the learning process in complex environments. Techniques like fine-tuning or feature extraction can be

employed to leverage pre-trained models and transfer knowledge to the A3C algorithm.

7. **Hyperparameter Optimization:** Computational learning techniques also play a significant role in optimizing the hyperparameters of the A3C algorithm. Techniques like grid search, random search, or more advanced methods like Bayesian optimization or genetic algorithms can be employed to search for optimal hyperparameters (Bergstra & Bengio, 2012). Computational learning allows for efficient exploration of the hyperparameter space, helping to find the best configuration for the A3C algorithm.

This theory gives a formal framework to accurately formulate and deal with the questions on how the different learning algorithms perform. This, therefore, ensures a good comparison of the ability to predict and the efficiency of computational ability of competing learning algorithms (Mohri & Talwalkar, 2018). The questions sought to be answered by this theory as supported by the authors include:

1. Is it possible to define a general measure of problem difficulty?
2. Can more difficult learning problems, regardless of learning algorithms that have been used and the figure of learning examples that have been given be identified?
3. The lower limit of learning examples (sample complexity) that are needed to ensure successful learning?
4. What is the required computational effort (time complexity) for a given sample size for successful learning?

The authors further outline the key aspects that are formalized as:

1. The way in which the learner interacts with its environment
2. What success is in completing a learning task

3. Formal definition of efficiency of both usage of the data (sample complexity) and processing time (time complexity)

Besides its predictive ability, this model also addresses other features like robustness to variations in the learning scenario, simplicity, and the capacity to make observations on empirically observed phenomena. While the theory provides valuable insights and theoretical foundations for understanding learning algorithms, there are challenges encountered when applying it in research including:

1. Theoretical Complexity: Since this theory involves complex mathematical formalisms and analyses, the frameworks and proofs can be challenging to understand and apply correctly. This then required a solid mathematical background to be able to do this.
2. Interpretability: While computational learning theory primarily focuses on algorithmic performance and generalization guarantees, can lack to address the interpretability or explainability of learning algorithms. This is because of the black box nature of complex machine learning models which remains a challenge. This was mitigated by incorporating human-understandable representations on the results of the model so as to rate its performance.

This theory was therefore significant to this study as the study sought to look at the efficiency of using the A3C algorithm by looking at its computational efficiency and predictive power. It would provide a formal guide in looking at the way the agent interacts with the environment and help set the success factor to be measured during the experimentation. This theory is close to machine learning research theory.

2.3.2 Machine Learning Theory

The machine learning theory has been attributed to various researchers and scientists including Tom Mitchel for their contributions over the years as it is a broad field. This theory complemented the Computational Learning Theory. The machine learning process is as guided in figure 2.

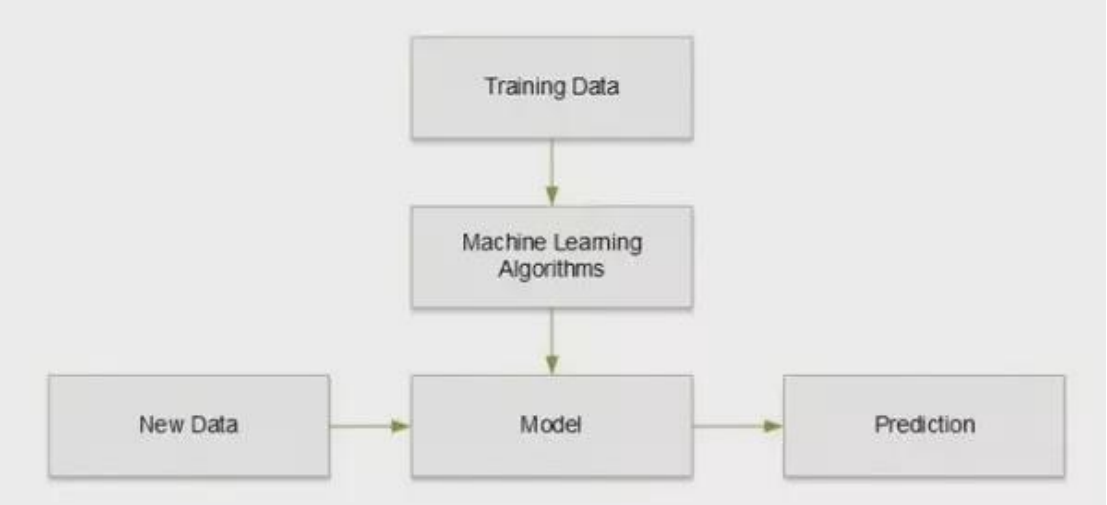


Figure 2: Machine Learning Process
Source: Penchikala (2016)

The training data is fed to the algorithms for training the model and the performance of the model checked by its ability to predict the new data introduced.

Requirements for ML systems

i. **Data**: This is Input data for the prediction of the output.
ii. **Algorithms:** Sequence of well-defined computer implementable instructions to solve a problem
iii. **Automation**: Making systems operate without human intervention
iv. **Iteration:** enables process repetition.
v. **Scalability**: To be able to increase and decrease capacity as per need

vi. **Modeling**: Represents what is learned by the algorithms; it is the output of the algorithm run on data

The machine learning theory looks at elements of both Computation Learning Theory and Statistics. It seeks to understand the capabilities and information needed to successfully learn various tasks (Hastie et al., 2009). It also seeks to comprehend the base algorithm principles used to make computers learn from data and improve how it performs their tasks with the feedback given.

Machine learning theory provides the foundation for understanding and applying the A3C algorithm effectively. The application of machine learning theory on the A3C algorithm looks at the following as supported by Sutton & Barto (2018).

1. **Markov Decision Process:** The A3C algorithm is built upon the theoretical framework of Markov Decision Processes (MDP). Machine learning theory guides on the required concepts and algorithms to model and solve MDPs. AN MDP outlines the dynamics of interactions of an agent with the environment and formalize the decision-making problem. The A3C algorithm makes use of the machine learning techniques to give approximations of the optimal policy and value functions of the MDP.
2. **Reinforcement Learning Theory:** This theory provides the theoretical foundations for training the A3C algorithm. Concepts such as rewards, value functions, policies, and the exploration-exploitation trade-off are at the center of reinforcement learning. The A3C algorithm makes use of value-based and policy-based methods, and utilizes concepts like temporal difference learning, policy gradients, and the Bellman equation to optimize the agent's behavior.

3. **Policy Gradient Methods:** This is a very important component of the A3C algorithm and uses gradients for policy optimization. Machine learning theory provides the mathematical foundations for policy gradient methods. REINFORCE algorithm techniques and its variants, with the A3C advantage function, enable the agent to make the decision to update its policy by using observed rewards and actions (Abbeel & Schulman, 2021).
4. **Value Function Approximation:** There are various techniques given by machine learning theory for value function approximation. This facilitates estimation of the expected rewards in reinforcement learning. The value function approximation is used to get an estimate of the state-value or action-value functions. Machine learning models including neural networks are used for function approximation in A3C, this is achieved by taking advantage of techniques like deep Q-networks or deep value networks.
5. **Exploration and Exploitation:** Machine learning theory gives insights into the trade-offs for exploration-exploitation in reinforcement learning. The A3C algorithm employs epsilon-greedy exploration or Boltzmann exploration techniques and the principles of multi-armed bandits and exploration strategies. Szepesvári (2010) posits that machine learning theory enables the determination of the appropriate balance between exploration and exploitation so as to make sure that efficient learning in the A3C algorithm is achieved
6. **Model-Free Learning:** The A3C algorithm is one of the members under the classification of model-free reinforcement learning. This is where the agent learns directly from environmental interactions without prior knowledge of its dynamics. model-free learning algorithms, such as Q-learning, SARSA, and policy gradients are encompassed in machine learning theory and they form

the basis for the A3C algorithm (Szepesvári, 2010). For its successful application, a clear understanding of the theoretical properties and convergence guarantees of these algorithms is important.

7. **Generalization and Transfer Learning:** Machine learning theory addresses the challenges of generalization and transfer learning, which are relevant to the A3C algorithm. Generalization refers to when the agent applies the knowledge it has learned to similar tasks or states it has not yet interacted with. Transfer learning on the other hand looks at leveraging knowledge acquired from one task so as to make an improvement when learning a related task (Taylor et al., 2016). Machine learning theory provides techniques to generalize and transfer learning. This makes the A3C algorithm adapt to many different environments.
8. **Model-Based Methods:** While A3C is primarily a model-free algorithm, machine learning theory includes model-based methods as well. Model-based reinforcement learning aims to learn an explicit model of the environment and use what it has learnt for planning and decision making (Deisenroth et al., 2013). Theoretical concepts from model-based learning can be applied together with A3C so as to guide the behavior of the agent and increase exploration. This includes model-based value iteration or Monte Carlo Tree Search.
9. **Convergence and Optimality:** Machine learning theory provides an understanding into the convergence and optimality of reinforcement learning algorithms. Understanding a concept like convergence guarantees stability, and a concept like optimality allows researchers to analyze the performance and behavior of the A3C algorithm. Theoretical guarantees make certain

convergence of the learning process to an optimal policy and value estimates over time.

Machine learning theory plays a vital role in the application of the A3C algorithm. The theoretical foundations of reinforcement learning, value function approximation, policy gradients, exploration-exploitation trade-offs, generalization, transfer learning, and convergence provide the necessary tools and understanding to effectively apply the A3C algorithm in various domains. Machine learning theory guides the development, analysis, and optimization of the A3C algorithm, enabling its successful application in real-world problems. The main limitation of this theory apart from the shared one with the computational learning theory encountered was data limitations and bias as machine learning heavily relies on quality and representativeness of the data hence biases present in the data can impact the performance and fairness of the learned models. This is because biased or incomplete data can lead to biased predictions. This was mitigated by using high-quality and representative dataset. Data preprocessing techniques, including data cleaning also helps solve bias in machine learning models.

From computational learning theory, an important factor is coming up with algorithms that can learn faster even when faced with the problem of large amounts of distracting information. The machine learning theory enables the performance of actions like creating mathematical models for machine learning aspects and developing machine learning algorithms. Machine learning theory was utilized in this study as it was used as a reference in the process of developing the A3C model. This process was adopted in creating the model as the key relationships of the different requirements were addressed. The theory was also used to better understand the requirements needed in ensuring successful learning by the model.

Ravitch (2016) instruct that the theoretical framework should be founded on theories that are identifiable and have been published. This study was based on the machine learning theory and computational learning theory which aided in the development of the A3C algorithm in intrusion detection.

2.4 Evaluation of the Machine Learning Techniques Used in IDS

There are many tools and techniques that are being used to secure information from unauthorized access. These include the use of access control which uses security policies to control and ensure the use of the system by only recognized users and devices. Another is using firewalls which use predefined rules and policies to filter incoming and outgoing traffic thereby preventing unauthorized access. There is also the antivirus and antimalware which monitors the network traffic in real-time, scans log files for suspicious behavior, and offer threat mitigation.

Hassan et al. (2018) defines an IDS as a mechanism, whether in the form of hardware or software, that is designed to scan a network or system to identify and mitigate unauthorized access, attacks, and policy violations. It works by analyzing network traffic, system logs, and other data sources so as to detect patterns and anomalies that indicate potential security breaches. IDSs are very important in maintaining the security and integrity of computer networks. They do this by providing real-time monitoring and alerts, enabling timely response and mitigation of security incidents. It is a critical aspect in contributing to organizational success. There have been different approaches applied in IDS in enterprises to ensure their security, availability, and reliability.

The IDS is categorized into 2 depending on where they discover anomalies as host-based and network based according to Endorf et al. (2012). The host-based IDS

analyzes the internal changes of a computer infrastructure on which it is installed e.g., a virus spreading. The network-based IDS operates at the network level. It will analyze the traffic of the device for malicious packets or unusual patterns like flooding attacks. The classification can also be based action as active and passive. Active takes action on the detected malicious activity while passive only detects with no action. Endorf et al. (2012) furthermore, categorizes the IDS in 2 as signature and anomaly based. Signature-based IDS detects intrusion by having a library of the attributes of familiar attacks by which to match network data and use it to pinpoint an intrusion. Although the signatures can be easily developed and understood and have a high detection rate, they only detect attacks stored in the database hence the library must always be updated by the security administrators because new attacks will not be detected (Zhang et al., 2015). Also, a mistake in signature definition will cause a reduction in the detection of the anomalies and the false positive rate will significantly increase. It can also be deceived due to its use of regular expressions and string matching. This means that they will work effectively against the fixed behavioral patterns and fail to deal with the ones with self-modifying characteristics.

In comparison, anomaly-based IDS establishes a normal network behavioral model in terms of ports, bandwidth, protocols, and others and alerts the security administrator of any deviation from normal or potentially malicious activity. Anomaly-based IDS is complex as it needs a training phase for the development of a database of general attacks and carefully setting a threshold level of detection. Detailed knowledge of acceptable network behavior needs to be developed for correct detection and so that novel attacks with no signatures will easily be identified and correctly detected if it is different from the standard acceptable traffic patterns. They have better recognition ability for novel anomalies but it has overall low detections and its false alarm rates

are high according to Kim et al. (2014). IDSs are effective in identifying anomalies in networks.

Mitchel (1997) has the definition of Machine Learning as an examination of computer algorithms that automatically improves through different experiences. This involves looking at new ways to perform various tasks without explicit programming hence they utilize training data to develop models that are used for decision-making and prediction. The learning method is by extracting patterns in data and get insights so as to make accurate predictions based on the examples with the aim being to allow learning without human intervention and adjust actions accordingly.

There is a great deal of promising results exhibited by machine learning in the field of intrusion detection systems. The IDSs are designed to detect and mitigate unauthorized or malicious actions within a computer network. An evaluation of these machine learning techniques are as follows;

1. **Improved Detection Accuracy:** Machine learning techniques have shown the ability to detect previously unencountered or sophisticated attacks that traditional approaches which are rule-based in nature may miss. These malicious actions can be identified with higher accuracy by machine learning models through learning patterns and anomalies from large datasets.
2. **Adaptability by Machine Learning Models to Changing Threat Landscape:** The models can adapt and learn from unencountered attack patterns, making them suitable for dynamic and ever evolving threats in cybersecurity. As attackers continuously change their methods, machine learning algorithms can be trained to detect novel attack patterns and through the adjustment of their detection capabilities, improve their level of detection.

3. **Automated Feature Extraction:** Machine learning techniques always make the feature extraction process automated. This enables the IDS to effectively handle data that is complex and high-dimensional reducing dependence on manual feature engineering, which can be time-consuming and relevant information may not be well captured. Machine learning models can learn discriminative features directly from raw data and this improves the accuracy in detection.
4. **Real-time Detection:** Machine learning algorithms including decision trees, support vector machines (SVM), and deep learning models, can provide real-time intrusion detection. This capability is crucial due to the prompt nature in identifying and responding to attacks and hence reducing the potential damage caused by malicious activities.
5. **False Positive Reduction:** Machine learning algorithms contribute to the reduction of false positives in IDS because they learn from normal network behavior and distinguish it from anomalies. The effective classification of benign activities, leads to minimized false alarms. This in turn enables the security analysts to focus their attention on genuine threats.
6. **Limitations in Handling Imbalanced Data:** Imbalanced datasets is where the number of normal instances is more than the number of malicious instances by a big margin. This is common in IDS. Machine learning techniques may struggle with this kind of data because they can be biased towards the classification with the majority. There are specialized techniques like undersampling, oversampling, or the use of cost-sensitive learning algorithms that are necessary to address this challenge.

7. **Vulnerability to Adversarial Attacks:** Machine learning models fall prey to adversarial attacks This is where inputs are intentionally manipulated by malicious actors to evade detection or mislead the IDS. Adversarial examples can significantly disrupt performance of the IDS leading to a compromised overall security. Adversarial robustness techniques, such as adversarial training and detection, are being developed to mitigate this vulnerability.
8. **Model Interpretation:** Many machine learning models, such as deep neural networks, are termed as black-box models because they lack interpretability and the ability to be explained. This is a challenge presented to security analysts and auditors who need to understand the rationale behind an IDS's decisions. This problem is addressed by techniques like rule extraction, feature importance analysis, or the use of inherently interpretable models.
9. **Training Data Requirements:** Machine learning models majorly make use of labeled training data so as to learn the characteristics of normal activities and anomalies. For IDS, getting training data that is representative and of high-quality can be difficult. This is because it requires access to real-world attack instances and their corresponding labels. To solve this limitation, generating synthetic data generation or making use of datasets that are publicly available datasets are some of the techniques that can be used.

The machine learning techniques have shown significant potential in the quest to improving the effectiveness and efficiency of IDS. However, challenges such as imbalanced data, vulnerability to adversarial attacks, and the need for interpretability and explainability should be wholly dealt with to ensure that the machine learning based IDS deployed is reliable and robust. Continuous research and development are

crucial to advancing the field and enabling more accurate and proactive threat detection.

Machine learning has been classified into three according to Bishop (2006) as: supervised learning where labeled data is used with the goal being to establish a relationship or general rule. The agent is directed on actions to maximize rewards by using examples from the labeled training dataset. Various models have been used in IDS including the Bayes Network, Random Forest Classifier, Multi-Layer Perceptron (MLP), and Decision table. Unsupervised learning uses unlabeled training data to find structure in those collections without prior training. It can detect new attack variants but it has a lot of false positives. The K-Means clustering is a good example where it groups the data, and assigns data points based on the dataset features which are then clustered by looking at the similarity of features.

The third classification is reinforcement learning. This is where interaction is made in a dynamic environment while working towards a goal in the problem space and is provided feedback as rewards, either positive or negative as it navigates the problem space and selects its actions to maximize that reward (Bishop, 2006). For reinforcement learning to be applied in anomaly detection, the key factors critical are the problem definitions: - which include the environment, state, action, and reward, another is the appropriate architectural model and the domain-specific training samples. The Agents are classified as either value-based (no policy), policy-based (no value function), and actor-critic which has both policy and value function. Bishop (2006) further looks at the elements of reinforcement learning as follows:

a) **Agent –** A hypothetical entity that acts in an environment to get a reward
b) **Action –** The possible moves that an agent can take

c) **Environment –** The scenario/outside world where the agent performs actions

d) **State –** The present status of the agent returned by the environment

e) **Reward –** A numerical return signal from the environment to gauge the last action taken by the agent

f) **Policy –** Strategy the agent uses to choose the succeeding action to take based on the current state. Also defined as the method to map agents' state to actions

g) **Value –** Delayed (future) reward received by the agent when it takes an action in a given state

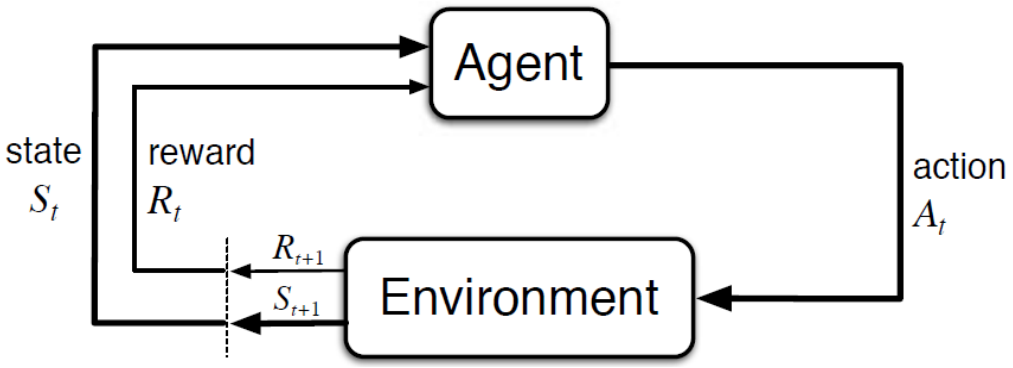


Figure 3: The interaction between the agent and the environment
Source: Sutton & Barto (1998)

The interaction of the agent and the environment is as shown in figure 3 from Sutton & Barto (1998). This interaction occurs at time steps, t = 0, 1, 2, 3…where information about the environment state S_t is received by the agent. the agent decides an action A_t by referring to the state of the environment at point t. After that, the agent will get a signal that is a numerical reward R_{t+1}. This becomes a sequence (S_0, A_0, R_1, S_1, A_1, R_2…). The state is the input used by the Reinforcement Learning agent in policymaking. This means that the state should be as correct as possible.

2.5 Related Work

Many works of literature exist and intrusion detection results are obtained by using different datasets. With that, it is difficult to compare the results of the experiments. This is because differences that are pivotal will depend majorly on testing dataset that has been used to get the results of the detection.

2.5.1 Intrusion Detection Systems

Deokar & Hazarnis (2012) indicated the disadvantages of signature and anomaly-based detection methods. The anomaly-based detection method has a high number of false positives. This is due to its flagging of activities occasionally performed as anomalies. The signature original multi-agent router throttling which detects by applying the divide-and-conquer does not detect novel attacks. This is because it uses stored patterns of attacks in detection. The proposed IDS would discover the attacks that are known and unknown by using the features of both anomaly and signature detection by using log files. The proposed IDS combined the RL method, association rule learning, and log correlation techniques.

Gao et al. (2019) used an Incremental approach of the Extreme Learning Machine (IELM) together with Advanced Principal Component Algorithm (APCA) to come up with an IDS. The APCA selects features required by the IELM adaptively for the prediction of an optimal attack. An evaluation of the IDS framework performance was done using the University of New South Wales Network Based 2015 (UNSW-NB15) dataset. The research concluded with the results of the experiment showing that the IELM-APCA scored an accuracy of 70.51%, Detection Rate (DR) of 77.36%, and obtained a FAR of 35.09%.

Ambusaidi et al. (2016) introduced an IDS put together with a filter-inspired input reduction approach. The Kyoto 2006 dataset, KDDCup99 dataset, and 18 features of the NSL-KDD were used in this experiment. The authors used a Flexible Mutual Information (FMI) technique, which is a non-linear correlation measure so as to maintain the correlation of the different input variables. The Least Square SVM classifier was applied in the study. The result of the experiment using the NSL-KDD dataset showed that, the LS-SVM FMI scored an accuracy of 99.94% and a false alarm rate (FAR) of 0.28%. Using KDD Cup 99 dataset, it was 78.86% accurate. On the 10^{th} iteration using the Kyoto 2006 dataset, it had a detection rate of 97.80% and FAR of 0.43%.

Awad & Alabdallah (2019) proposed a Weighted Extreme Learning Machine (ELM) approach to intrusion detection. The results of the performed experiments gave a precision of around 99%.

Hu et al. (2009) looked at system call-based Hierarchical Gaussian Mixture Model (HMM) training in host-based anomaly intrusion detection. This was later improved by preprocessing the data for recognition and elimination of redundant sub-sequences of system calls, resulting in a smaller number of HMM sub-models. The results of the experiment on the three public databases showed that the cost of training can be cut almost by half without influencing its performance in intrusion detection. There is also a higher but reasonable false alarm rate when checked against the batch method of training which indicated a data reduction of 58%.

A proposal was made by Nakkeeran et al. (2010) for a detection system consisting of layered detection modules. The results of the nodes neighboring the current node are

picked by the present node and the result is passed on to the neighboring nodes. The experiment posted an increase in detection and a reduction in false detection.

Jiong & Mohammad (2006) proposed an unsupervised anomaly Network Intrusion Detection System framework that utilized the outlier detection technique. This was done using a random forest algorithm. The framework builds the patterns of network services over datasets labelled by the services. Using the built patterns and the modified outlier detection algorithm, the framework performs detection of the attacks using the datasets thereby leading to a reduction in calculation complexity. This approach makes an assumption that individual network services have their pattern for normal activities.

Miao Wang, et al. (2006), proposed a method for the Windows Host Anomaly Detection System, which adds to the work of the other security methods in windows. It detects intrusions that invoke an anomaly sequence by programs. General situations such as normal programs that are used without authentication cannot be detected.

Vikash et al. (2020) accomplished developing an IDS system and applied the UNSW-NB15 dataset to validate it. The authors used the Information Gain methodology to come up with a feature reduction method. They used this filter-based feature extraction technique to select 22 of the 42 attributes. The classification process was carried out by using an integrated rule-based model that used multiple Tree-based classifiers on the IDS. The Attack Accuracy (57.01%), F-Measure (90%), and False Alarm Rate (2.01) were the performance measurement results of the system. The recommendation was that it can be improved by using other Machine Learning algorithms to replace the Tree-based methods.

Hazem & Nikos (2008) proposed an intrusion detection algorithm and an architecture that includes 2 layers, a global and local layer which perform data collection, analysis, and response functions. The global layer is central. This system applies a data mining approach and by filtering out intrusive behavior and normal, analyzes data that is not known resulting in reduced false alarms.

Vasilis et al. (2010) designed a machine-learning model for anomaly detection through a Bayesian Support Vector Machine (SVM). The use of an SVM with one class to early detect system anomalies is studied along with drift output classification probabilities. Experimentally, when failure training data under one-class SVM is not present, unknown anomalies are recognized quickly. Initially dividing the training data into multiple unrelated lower-dimensional models, an evaluation of the test data is done independently on each model to show outliers in different capacities (as is in the posterior class probabilities evaluation in the Bayesian framework).

Jiang et al. (2020) suggested a Network IDS framework using the One-Side Selection technique for reducing the number of meaningless data records in the majority of the classes. Synthetic Minority Over Sampling was applied to raise the dataset minority examples with Convolutional Neural Networks being used to bring out the dimensional attributes and Bi-directional Long-Short Term Memory models for temporal attributes. This becomes a Deep Learning model for conducting predictive tasks. The UNSW-NB15 and NSL-KDD datasets were used in the assessment. The accuracy obtained from the test data was 77.16% for UNSW-NB15 and 83.58% for NSL-KDD datasets.

Osanaiye et al. (2016) offered a filter-based method that is multiple-filtered for detecting Distributed Denial of Service attacks. The approaches applied include

Information Gain, Gain Ratio, Relief, and Chi-Square. They used the attack detection datasets from the NSL-KDD to show the performance and for classification, they employed the Decision Tree (DT) algorithm. The DT algorithm was subjected to the k-fold cross-validation method where k=10 for training and validation. The results were that using 13 out of 42 features the DT classifier had a detection of 99.67% and a FAR of 0.42%.

The conceptual study reveals the following:

i. Most intrusion detection systems and research carried out use supervised and unsupervised methods of machine learning. These methods have achieved accuracies of up to 98% for selected datasets. Zhou et al. (2021) argue that machine learning algorithms such as Random Forest (RF), Bayes, AdaBoost, achieved around 99% of precision for the anomaly of ‘DDoS’ and ‘Portscan’.

ii. The results of most studies done using machine learning in intrusion detection using the available datasets have been based on selected features of the dataset, thus feature selection has to be done on the datasets so as to get an optimal feature representative of the full feature set (Eesa et al.,2015). The factors affecting the results include the size of the dataset employed, the compute time and the performance of the algorithms (Chandrashekar & Sahin, 2014).

iii. Modern IDS perform well in detecting regular and known intrusions but are weak in adversarial AI attacks where malicious inputs are injected into the A.I training data. This calls for a need for continuous improvement of the performance of intrusion detection systems. This is because accuracy and false alarm rate had not been fully optimized.

2.5.2 Machine Learning Methods

DRL methods from an integration of deep and reinforcement learning can detect and prevent sophisticated types of cyber-attacks. Quah et al. (2002) use Temporal Difference learning on an actor-critic reinforcement learning model and use it to classify by using a fuzzy adaptative learning control network. The model developed is used on the Iris dataset.

Grounds & Kudenko (2008), proposed a parallel version of the Sarsa algorithm. This made use of various separate actor-learners to make training faster. Individual actor-learner learns on its own and at set times sends an update to weights that have experienced significant change compared to other learners using peer-to-peer communication.

Bhosale et al. (2014) came up with a proposition of a multi-agent intelligent system that applied reinforcement learning and an influence diagram to respond fast to attacks that are complex and use rules and procedures to assess the state of captured flow. Every agent uses the local database along with the information from others (decisions and events) to learn its policy.

Khammassi & Krichen (2017) use Genetic Algorithm (GA) and logistic regression for optimal feature subset selection on the UNSW-NB15 and KDDCup99 datasets. This showed that the feature subset chosen using the method is effective for intrusion detection through different decision tree algorithms. After multiple simulations using the Weka simulation tool, on 20 of the 42 features of the UNSW-NB15 dataset, the Logistic Regression and Genetic algorithm together with DT classifier scored an accuracy of 81.42% and a FAR of 6.39%. Also, using 18 features of the KDDCup99 dataset got a detection accuracy of 99.90% and FAR rate of 0.105%.

Khan et al. (2018) was able to develop a technique for feature reduction based on the Random Forest algorithm to give the score on Feature Importance (FI) of every attribute present in the UNSW-NB15 dataset. The FI algorithm uses ranking where the feature that scores the highest FI becomes the most important attribute when classifying in relation to the target variable. A feature subset of 11 attributes was selected and the following Machine Learning approaches were used for classification: kNN, Decision Tree, Bagging Meta Estimator, Extreme Gradient Boost (XGB), and Random Forest. The performance of these indicated that the Random Forest algorithm was the best with an accuracy of 75.56% and Fowlkes-Mallows Index (FM) of 73%.

Muda et al. (2011) proposed a combination of K-Means clustering and Naïve Bayes classification for hybrid learning. The hybrid approach clusters data before classifying. The result of the experiment on the KDD Cup ’99 dataset indicated that this method had an accuracy of 99.5% on DOS and a precision of 40% for User to Root (U2R).

A new Non-negative Matrix Factorization (NMF) based model was proposed by Wei et al. (2004) to characterize both the program and the user practices for anomaly intrusion detection. This method sourced its information from the audit data stream that was derived from progressions of system calls and commands. The audit data is split into limited-length components. The frequencies of individual system calls or commands placed in each piece of data are used to rate how the program and user behave and to bring out the features from the blocks of audit data associated with the normal behaviors, NMF is used. The model that gives a description of the normal program and user behaviors is then built using these features as a basis. Drifting from the expected program and user conduct above a premeditated threshold is then taken as an anomaly. This method differs from other methods which use transition property

because it puts into consideration the constant property of system calls and commands. The results published by the authors from the simulation show that consideration of individual system calls or commands is not necessary, thus it can be deduced that the cost of computing using the NMF method is reasonable and is acceptable for real-time intrusion detection.

Jian et al. (2005), proposed an immunology-inspired method for identifying irregularities in system performance. This approach incorporates the negative selection algorithm (NSA) and the genetic algorithm. This included generating B set of fuzzy rules to distinguish the normal and the abnormal behaviors. NSA filters and gets rid of the baseless detections so as to reduce the search space. The issue with anomaly detection was a two-class classification problem according to the authors, the intention was to group system patterns as self and non-self. The ideas from Fuzzy Logic are used in solving the classification problem. Using the samples; including self and non-self, fuzzy detector rules must be initiated in the non-self-space that will aid in finding the category the new sample belongs to; either self or non-self. There is an improvement in detection as a result of using the fuzzy rules and like any other technique, these rules do not employ hyper-rectangles for representation. The advantages include; The negative selection algorithm acts as a filter in numerous times. It is added to the Genetic Algorithm (GA) especially as a filter operator that gets rid of the bad solutions efficiently so as to have reduced False Positives (FP). A new partial match of a chromosome that is applied in getting compute the distance of two different individuals is adopted by the algorithm. A better depiction of the boundaries for the self and non-self-classes is provided by fuzzy rules and they reduce the search space.

Zhenghong et al. (2009) proposed utilizing Machine Learning for anomaly detection by using a Bayesian classification algorithm for detection of inconsistent nodes to the Wireless Sensor Networks characteristics like restrictions of nodes in terms of power, communication capacity, and computational capabilities. This achieves relatively high accuracy in detection and lowers false positives. After learning of a few samples is complete, it is also possible to establish detection rules. Using the model for simulation the authors attempted to prove that the proposed model could retrieve important association rules and differentiate with high accuracy between normal connection and the intrusion which is unknown.

The most used detection approach used is the anomaly detection as supported by Karami & Guerrero-Zapata (2015) due to its effectiveness in detecting new attacks. There is a wide range of techniques applied and the diversity of the datasets used for evaluation allows for assessing the performance of the proposed approaches across different scenarios. The studies generally report good performance in terms of accuracy, precision, and low false alarm rates. They showcase the potential of machine learning techniques in enhancing the effectiveness of IDS.

Despite the contributions of these research studies, several gaps and limitations can be identified:

1. **Lack of real-world data:** The benchmark datasets relied on by these studies may not fully represent real-world network traffic and attack scenarios as supported by Duan et al. (2020). Incorporating more diverse and up-to-date datasets would enhance the practicality of the proposed IDS approaches.
2. **Limited evaluation metrics:** Alrawashdeh et al. (2019) points out that while accuracy and false alarm rates are commonly reported, studies could benefit

from evaluating the proposed approaches using additional metrics such as recall and F1 score to provide a more comprehensive assessment of their performance.

3. **Scalability and computational complexity:** Some studies do not thoroughly address the scalability of their approaches to handle large-scale networks or the computational complexity required for training and deployment. Considering these factors is crucial for the practical implementation of IDS in complex environments.
4. **Lack of comparative analysis:** The studies often evaluate their proposed approaches in isolation, without making a comparison to existing state-of-the-art IDS methods (Khalid et al., 2018). Comparative analysis is important as it provides insights into the relative strengths and weaknesses of different techniques.
5. **Limited focus on online learning and adaptability:** Continuous monitoring and adaptation are crucial for IDS to handle evolving attack patterns. Many studies do not explicitly address the ability of their approaches to adapt to dynamic network conditions and detect emerging threats.

If these gaps are addressed, it will contribute to the development of more robust and practical IDS solutions that can effectively detect and respond to evolving cyber threats in diverse network environments.

For the gaps identified in these studies, it shows that there is a need to continually build on the intrusion detection systems so as to ensure maximum performance. This is because accuracy and false alarm rates have not been fully optimized in the existing IDSs. Besides, deep reinforcement learning applications on intrusion detection

systems have not been exhaustively tested. This then calls for conducting more research in the area and this is where this study comes in by applying the A3C algorithm to IDS.

2.6 Effectiveness of Using A3C in Anomaly Detection

The Asynchronous Advantage Actor-Critic algorithm is one of the most recent algorithms developed by employing Deep Reinforcement Learning. It makes use of numerous agents each having information on the network and a copy of the environment (Mnih et al., 2016). These agents explore and learn with each interaction with their individual environments asynchronously**.** A3C is good because it does not overutilize the GPU resource hence computationally efficient and also uses less time when training. Its clear structure also makes it easier to debug and optimize.

A3C is not specifically designed for anomaly detection because the algorithm is primarily used in reinforcement learning tasks, particularly for training agents to interact with environments and learn optimal policies. It is possible to adapt the algorithm for this purpose, however, the effectiveness of using A3C for anomaly detection may vary depending on the specific context and requirements.

According to Chandola et al. (2009), anomaly detection typically involves looking for patterns or instances that deviate in a noticeable range from normal behavior. Traditional anomaly detection methods often rely on statistical approaches, clustering techniques, or rule-based systems. These always make an assumption that anomalies rarely occur and differ significantly from normal observations.

Applying A3C to anomaly detection involves training an agent to learn the normal behavior of a system or dataset and then using the learned policy to identify anomalies. The agent would interact with the environment, observe states, take

actions, and receive rewards based on the deviation from normal behavior. By optimizing the policy through reinforcement learning, the agent can learn to distinguish anomalies from normal patterns.

The effectiveness of using A3C for anomaly detection depends on several factors as supported by Abbeel & Schulman (2021) including:

1. Availability of labeled data: Reinforcement learning methods like A3C basically require a significant amount of training data. If labeled data representing both normal and anomalous behavior is available, it can be used to train the agent effectively. However, obtaining labeled anomaly data may be challenging in many real-world scenarios.
2. Complexity of the anomaly patterns: A3C, like other reinforcement learning algorithms, may struggle to detect complex or subtle anomalies that deviate from normal behavior. The agent's performance heavily depends on the representation of the state space and the ability to capture relevant features.
3. Training time and computational resources: Reinforcement learning algorithms, including A3C, can be computationally intensive and time-consuming. Training a robust anomaly detection model using A3C may require substantial computational resources and extended training periods.
4. Generalization to new anomalies: A3C-based anomaly detection models may have limited generalization capabilities. If the agent is trained on a specific set of anomalies, it may struggle to detect novel or previously unseen anomalies. Ensuring the model's ability to generalize to new anomalies would require a diverse and representative training set.

It's important to carefully consider these factors and evaluate the performance of the A3C algorithm in the specific anomaly detection context before drawing conclusions about its effectiveness.

Deep Reinforcement Learning consists of $V(s)$, the long-term expected cumulative reward starting from the state (s) and is useful in appraising the state (s), $Q(s, a)$ refers to the long-term expected cumulative reward starting from executing action (a) in the state (s) and $\Pi(s, a)$, the probability of executing action under state (s) (Sutton & Barto, 2018).

Mnih et al. (2016) explains that A3C algorithm works by executing various copies of the agent in multiple environmental instances in a non-simultaneous manner. When training, the agent interacts each with its own occurrence of the environment, fetches different samples for training, computes the gradient of the network, uploads the gradient to update the global network, and finally downloads the latest network to the local. The global network of the master agent cannot be updated by the numerous copies of the agents' uploaded gradients at once. Figure 4 illustrates the interaction of the Master and Agent and how their interactions are updated.

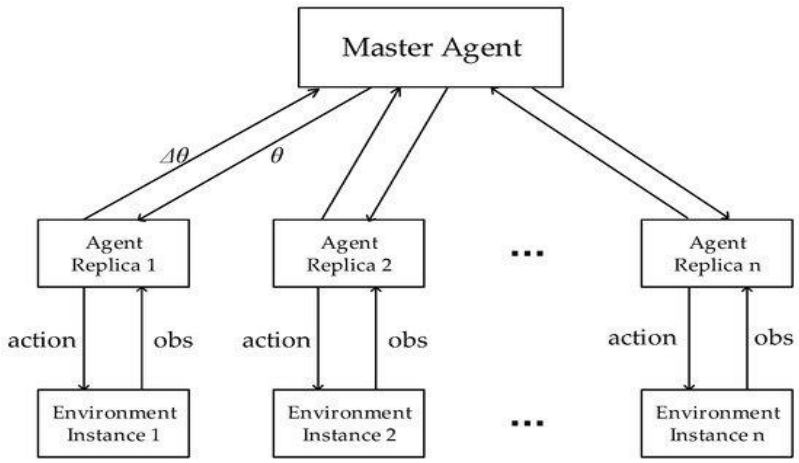


Figure 4: Master-Agent Interaction
Source: Zeng et al. (2019)

Each agent replica interacts with the respective environmental instance by observing the environment and performing an action.

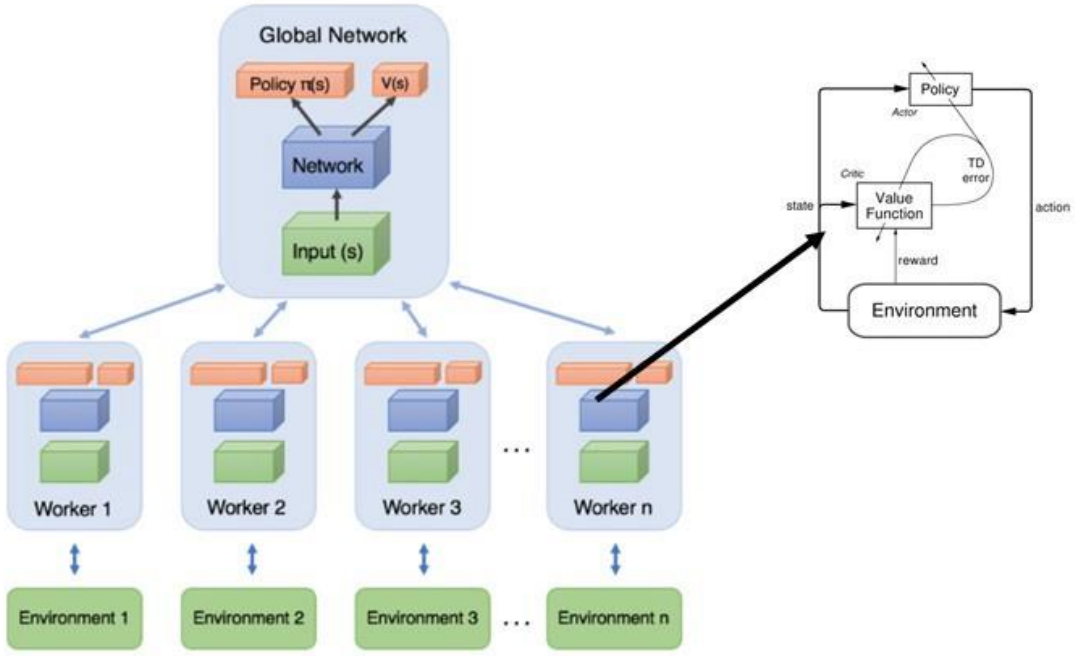


Figure 5: A3C architecture
Source: Karagiannakos, S. (2019).

Figure 5 shows the architecture of A3C. The workers are trained in parallel and the global network is periodically updated which holds shared parameters. Asynchronous comes from the fact that updates are not happening simultaneously. The agents reset their parameters to those of the global network with each update and continue their independent exploration and training for n steps until they update themselves again.

From the figure, information flows not only from the agents to the global network but also between agents as each resets their weights by the global network, which has the information of all the other agents. Actor-Critic takes advantage of both value-based and policy-based while eliminating all their drawbacks. The actor takes the state as input and action as output. The agent is controlled by learning the optimal policy. The critic, on the other hand, looks at the action by computing the value function. The

actor and critic get better in their own role as time passes resulting in a more efficient architecture.

Since the global network controls the agents and each agent sends the gained knowledge from exploration to the global network and the global network also enables the agents to acquire diversified training data, this enables the algorithm to perform better than other reinforcement learning techniques. The workers and the global agent are modeled as Deep Neural Networks each with two outputs: One is for the critic which is scalar and shows the expected reward of a given state, V(s). The other is for the actor which is a vector showing probability distribution over all possible actions (s, a).

Contrary to some straightforward methods which are dependent on either Value-Iteration methods or Policy-Gradient methods, the A3C algorithm merges the best parts of both methods i.e., the algorithm predicts both the value function V(s) as well as the optimal policy function π(s) (actor-critic). Advantage value shows the agent how the rewards were better when contrasted to its expectation. The agent then gains better insight of the environment which improves the learning process. The following expression gives the advantage metric according to Sutton & Barto (2018).

Advantage: A = Q (s, a) – V(s)

The behavior of the agent is defined by the policy and this is seen as a mapping function from states to actions. This policy is the chance of an agent acting under a given state modeled as p(a|s). A3C gives the parameters of the policy function π ($a_t|s_t;\theta$) and the value function $V(st;\theta v)$, using the actor network and the critic network respectively. The algorithm maintains a policy π ($a_t|s_t;\theta$) and an approximation of the value function $V(s_t;\theta_v)$. The actor-critic variation works in the forward view and uses

a mix of n-step returns for updating the policy and value functions. Sutton & Barto (2018) further reiterate the updates of the policy and the value function are done after every t_{max} actions or when a final state is arrived at. The update performed by the algorithm can be seen as $\nabla_{\theta'} \log\pi(a_t|s_t;\theta')A(s_t, a_t;\theta,\theta_v)$ where $A(s_t,a_t;\theta,\theta_v)$ is an estimate of the advantage function given by: $\sum_{i=0}^{k} {}^{1}\gamma^i r_{t+i} + \gamma^k V(s_{t+k};\theta_v) - V(s_t;\theta_v)$, where k varies from one state to another and has the upper bound t_{max}.

Each copy of the agent calculates the actor and critic networks gradients in every fixed step or when the episode terminates and then asynchronously syncs the gradients so that the global network is updated. When calculating the gradient of the critic network, the following formula is used:

$d\theta v=\partial(A(st;\theta v))2/\partial\theta v$ where $A(st;\theta v)$ stands for estimation of the advantage function that is given by the following expression:

$A(st;\theta v)=\sum k-1i=0\gamma irt+i+\gamma kV(st+k;\theta v)-V(st;\theta v)$

where γ is the discount factor and k is the variation of the sample sequence length and the index of the current sample (coded from zero).

The expression for calculating the gradient of the actor-network is as shown:

$d\theta p=\nabla\theta plog\Pi(at|st;\theta p)A(st;\theta v)+\beta\nabla\theta pH(\Pi(st;\theta p))$

where H represents the entropy of the policy Π. The hyper-parameter β keeps in check the strength of the entropy regularization term. The exploration ability of the agent is aided by adding entropy to the activity of updating the gradient of the policy function. It also stops the convergence of policies to the local optimal solution (Mnih. V et al., 2016). The concept of entropy measures the amount of uncertainty or randomness in a set of data (Gao et al., 2020). It gives an understanding of how much information is needed to describe or predict the outcomes within that data. Entropy is applied in various fields, such as information theory, statistics, and machine learning. In

machine learning, entropy is often used as a criterion to measure the impurity or disorder in decision trees or to guide the training process of models.

A3C is different from A2C in that it has workers that are independent and they have their weights. These workers interrelate with a mimic of the environment in parallel. This means that the state - action space that they can traverse is bigger and takes less time in theory. The global network will update itself and update the worker as soon as the worker sends an update. The parameter's information reaches the global network from the workers and also between workers as each worker resets weights that agree with the global network.

Advantages of A3C

Mnih et al. (2016) gives the advantages of A3C as follows;

i. The A3C algorithm is speedier and more vigorous when compared to the regular Reinforcement Learning Algorithms.
ii. The algorithm has better performance than the other Reinforcement learning algorithm since it utilizes knowledge diversification.
iii. The algorithm can be utilized in both the discrete and continuous action spaces.

The A3C algorithm will be effective in intrusion detection as envisioned in the conceptual model in figure 6 below.

2.7 Conceptual Model

A conceptual model is a framework that has been utilized by researchers to define the trajectories available for taking an action or present an idea or thought (Elangovan & Rajendran, 2015). It provides a visual representation of the target theoretical constructs and variables.

In using reinforcement learning, the agent actions include dataset selection, prediction, and validation and coming up with a policy that maximizes rewards to attain higher accuracy as a result of the intrusion detection. The reward given is measured by correct recognition by the agent. If the agent correctly predicts, a reward is given. The objective the agent has is to get as many rewards as possible. The environment state variables the traffic date from the UNSW-NB15 dataset. The conceptual model shows the systematic approach to the processes based on the A3C algorithm. Figure 6 shows the reinforcement learning conceptual model for intrusion detection.

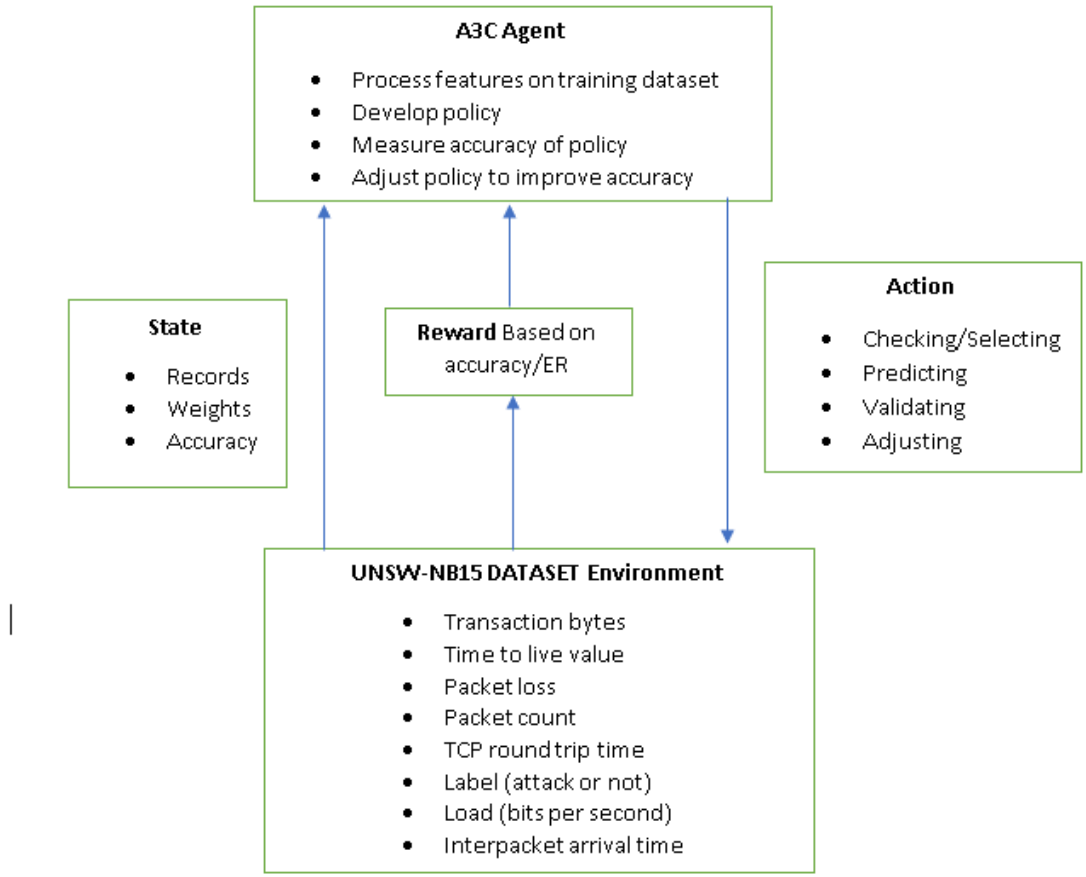


Figure 6: A Reinforcement Learning Conceptual Model for Intrusion Detection
Source: Author

A Markov Decision Process (MDP) is a mathematical framework used to model decision-making problems and has states, actions, transition probabilities, and

rewards. Puterman (2014) supports that the agent interacts with an environment by taking actions in specific states. The outcome of an action is determined probabilistically based on the current state and the chosen action. The state-action pair is associated with a reward value, which shows how desirable the action is in that state.

The goal of the agent is to learn a policy, that maximizes the cumulative expected reward over time. This is achieved by estimating the value function, which represents the expected cumulative reward from a particular state following a certain policy.

The Markov decision process and the conceptual model both look at the agent's interaction with the environment and the choosing of the optimal action so as to maximize rewards at the states.

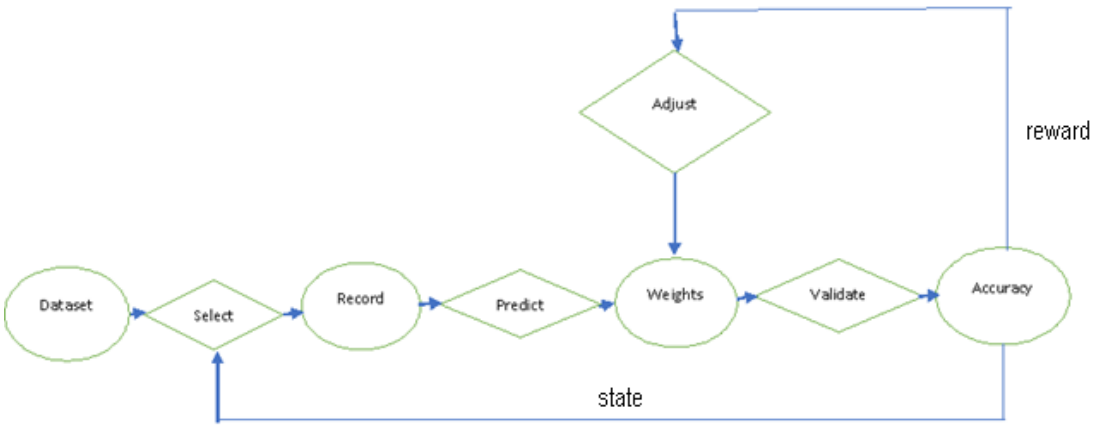


Figure 7: A Markov Decision Process
Source: Author

The agent processes the state variables and performs actions. The reward is calculated using the actions and the feedback is given to the agent and finally the agent updates the policy based on the reward and states and the process is repeated.

2.8 Appropriate Dataset for Use on A3C

There are many datasets available for use in the study including the UNSW-NB15 dataset, KDD CUP ’99 dataset and the NSL-KDD dataset. The UNSW-NB15 dataset was selected for the study and shall be discussed in detail.

2.8.1 The UNSW-NB15 Dataset

There have been challenges in finding datasets that give a true reflection of the network traffic in the modern world and also has various low footprint attacks according to Garcia-Teodoro et al. (2009). This is because most of them are private because of security and privacy concerns. Publicly available datasets have also been largely anonymized and fail to validate that they exhibit real-world network traffic behavior. This makes it difficult to get a comprehensive and up to date data. It becomes difficult for researchers to develop effective intrusion detection systems and evaluate their performance. This is caused by the challenge of assessing the robustness and generalizability of proposed approaches in detecting low-footprint attacks and accurately modeling network behavior Barocas & Selbst (2019). This requires a collaborative effort between researchers, industry professionals, and organizations to develop and share more diverse and realistic datasets. Having a balance between ensuring data privacy and security while still providing access to datasets that reflect the complexities of modern network traffic and encompass a wide range of attack scenarios is important.

The KDD98, KDDCUP99, and NSLKDD datasets were generated years ago. These datasets have been shown by many studies including Tavallaee et al. (2009) and McHugh (2000) not to be reflective of current network traffic and lack modern low footprint attacks found in the current network threat environment. Many studies have

highlighted the importance of using up-to-date datasets reflecting network threats' evolving nature (García-Teodoro et al., 2012; Sperotto et al., 2010). This is made with an emphasis that the dataset's quality significantly impacts detection systems' reliability and performance as posited by Gogoi et al. (2012).

The UNSW-NB15 is a full-scale dataset for network intrusion detection systems and was created and benchmarked by Moustafa et al. (2015) and the University of New South Wales, Australia. The IXIA PerfectStorm tool was used in this dataset. This was created for producing a combination of real modern behaviors that are considered as regular and synthetic contemporary attacks using new and existing processes for feature generation. The dataset traffic was captured in two days, on the 22nd of January and 15 hours on 17th February 2015 (Moustafa et al., 2015) in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). The preprocessed data is in packet capture (PCAP) files, (Comma Separated Value) CSV files, and the output of flow information extraction tools Argus and Bro-IDS. The dataset contains 45 features that are both numeric and categorical. These features are grouped as:

i. Flow features include transport layer features like IP addresses, ports, and protocol type
ii. Basic contains packet-based and flow-based features, such as duration, number of bytes, and number of packets.
iii. Content features contains exploration of connection content and window advertisements or sequence numbers.
iv. Time features contain time related aspects like jitter, start & end time
v. Others are categorized as general purpose and connection features.

Table 1 below shows the table of all the features in the UNSW-NB15 dataset.

Table 1: The dataset features, described in Moustafa and Slay (2016)

No.	Name	Type	Description
1	Srcip	nominal	Source IP address
2	Sport	integer	Source port number
3	Dstip	nominal	Destination IP address
4	Dsport	integer	Destination port number
5	Proto	nominal	Transaction protocol
6	State	nominal	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)
7	Dur	Float	Record total duration
8	Sbytes	Integer	Source to destination transaction bytes
9	Dbytes	Integer	Destination to source transaction bytes
10	Sttl	Integer	Source to destination time to live value
11	Dttl	Integer	Destination to source time to live value
12	Sloss	Integer	Source packets retransmitted or dropped
13	Dloss	Integer	Destination packets retransmitted or dropped
14	Service	nominal	http, ftp, smtp, ssh, dns, ftp-data ,irc and (-) if not much used service
15	Sload	Float	Source bits per second
16	Dload	Float	Destination bits per second
17	Spkts	integer	Source to destination packet count
18	Dpkts	integer	Destination to source packet count
19	Swin	integer	Source TCP window advertisement value
20	Dwin	integer	Destination TCP window advertisement value
21	Stcpb	integer	Source TCP base sequence number
22	Dtcpb	integer	Destination TCP base sequence number
23	Smeansz	integer	Mean of the ?ow packet size transmitted by the src
24	Dmeansz	integer	Mean of the ?ow packet size transmitted by the dst
25	trans_depth	integer	Represents the pipelined depth into the connection of http request/response transaction
26	res_bdy_len	integer	Actual uncompressed content size of the data transferred from the server's http service.
27	Sjit	Float	Source jitter (mSec)
28	Djit	Float	Destination jitter (mSec)
29	Stime	Timestamp	record start time
30	Ltime	Timestamp	record last time
31	Sintpkt	Float	Source interpacket arrival time (mSec)
32	Dintpkt	Float	Destination interpacket arrival time (mSec)
33	Tcprtt	Float	TCP connection setup round-trip time, the sum of 'synack' and 'ackdat'.
34	Synack	Float	TCP connection setup time, the time between the SYN and the SYN_ACK packets.
35	Ackdat	Float	TCP connection setup time, the time between the SYN_ACK and the ACK packets.
36	is_sm_ips_ports	Binary	If source (1) and destination (3)IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0

37	ct_state_ttl	Integer	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).
38	ct_flw_http_mthd	Integer	No. of flows that has methods such as Get and Post in http service.
39	is_ftp_login	Binary	If the ftp session is accessed by user and password then 1 else 0.
40	ct_ftp_cmd	integer	No of flows that has a command in ftp session.
41	ct_srv_src	integer	No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).
42	ct_srv_dst	integer	No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).
43	ct_dst_ltm	integer	No. of connections of the same destination address (3) in 100 connections according to the last time (26).
44	ct_src_ ltm	integer	No. of connections of the same source address (1) in 100 connections according to the last time (26).
45	ct_src_dport_ltm	integer	No of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26).
46	ct_dst_sport_ltm	integer	No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26).
47	ct_dst_src_ltm	integer	No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).
48	attack_cat	nominal	The name of each attack category. In this data set , nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms
49	Label	binary	0 for normal and 1 for attack records

The training set which is employed for the training of models has 175,341 records and then the set employed for testing trained models is the testing set and has 82,332 records from both the attack and normal.

Table 2: The 9 families of attacks, described in Moustafa and Slay (2016)

CATEGORY	TRAIN	TEST
Normal	56000	37005
Backdoor	1746	583
Analysis	2000	677
Fuzzers	18185	6062
Shellcode	1133	378
Reconnaisance	10492	3496
Exploits	33393	11132
DoS	12264	4089
Worms	130	44
Generic	40000	18871
Total	175343	82337

The UNSW-NB15 dataset includes 9 families of attacks, described in Moustafa & Slay (2016) as shown in Table 2. They are outlined as follows depending on the nature of the attack.

i. **Fuzzers:** This attack utilizes huge amounts of data that are random known as "Fuzz". This causes network failure or attempts to crash crucial network servers.

ii. **Analysis:** Contains attacks based on port scans, vulnerability scans, spam files, and footprinting. This type of attack is also called Active Reconnaissance and is where the network is scanned in some particular way but is not exploited.

iii. **Backdoors**: It uses malicious software to position themselves in a computer system and to provide remote access to attackers as a part of an exploit.

iv. **Denial of service (DoS):** This is where a lot of illegal connection requests deny the use of network resources by the intended users leading the machine to a state of compromise. They can be hard to differentiate from legitimate network activity; however, there are some indicators to detect such intrusive activities in progress.

v. **Exploits:** They target known vulnerabilities in operating systems hence compromising them. Once detection of a potential vulnerability in a network is made, these attacks can be automated.

vi. **Generic:** This is a cipher-based attack, which is essentially a collision attack on the secret key generated by the cryptographic principles. This analysis can be applied to block, stream and message authentication code cyphers. It is often referred to as a collision attack as most of these families are susceptible to the birthday attack. The attack depends on the higher likelihood of collisions found between random attack attempts.

vii. **Reconnaissance:** This type of attack collects exploratory information on a public network or target host and uses exploit techniques to intrude on target hosts or networks by maximizing the use of the gathered information. Reconnaissance uses the freely available information in public “Whois” service, ARIN records, and Shodan. Passive reconnaissance includes searches in the social media which also assist in reconnaissance attacks.

viii. **Shellcode:** This is a subgroup of the exploit class. This attack utilizes a small piece of code as a payload of an exploit. The malicious code is injected into an active application to compromise and gain remote access to a victim's computer. It typically starts a command shell from which the attacker can control the compromised machine.

ix. **Worms**: A worm is a malicious attack that spreads through network propagation and infects a much larger network rather quickly. The worm infects computers and makes zombies or bots, intending to use them in distributed attacks by forming botnets.

2.9 Model Validation

Splitting the data when training the model is the basis of validation techniques. This is done so as to understand what happens if the model is faced with data that it has not interacted with before. There are several validation methods used. The train/test split method will be used in this study. This is selected for computational efficiency and the size of the dataset (Brownlee, 2016).

2.9.1 Train/Test Split

Goodfellow et al. (2016) emphasize that it is important to separate data into training and testing sets for model evaluation. The train/test split is as a crucial step in building and validating deep learning models. The train/test split is a technique to evaluate how a machine learning algorithm performs. This technique involves splitting the dataset into training and testing sets. The model is then trained using the training set and applied to the testing set (Goodfellow et al., 2016). The question when validating the models is how well the validation datasets will capture the representation of contemporary low-rate attacks. An evaluation of how the IDS performs is usually done by running it over data that has malicious and legitimate traffic.

The figure of malicious traffic detected by the IDS (True Positives) and one misclassified as a legitimate instance (False Positives) are then counted. For a more satisfactory result, the evaluation data used in the method must be a representation of the real world. Since attacks are constantly evolving, the evaluation of an IDS with old datasets may yield a good performance but the outcome may not accurately represent its potential to detect contemporary attacks. The experiments conducted by Hadžiosmanovic et al. (2012) and Wressnegger et al. (2013) showed that the performance of the IDS is low when faced with more recent attacks.

2.10 Chapter Summary

This chapter looked at the previous research on the study, computation and machine learning theories as the theoretical frameworks informing the study and a conceptualization and validation of the study. The theoretical frameworks provide robust Learning Framework. Machine learning theory provides a solid foundation for understanding the principles and algorithms behind learning from data. Computational learning theory, in particular, focuses on the theoretical analysis of learning algorithms and their performance guarantees. Application of these theories to the A3C algorithm helps ensure that the IDS model learns effectively and provides reliable results. The conceptualization and use of MDP gives an in depth understanding of the working of the IDS while the process flow gives an implementation roadmap of the model. Chapter three looks at the Research Methodology.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

Creswell (2014) refers to research methodology as the systematic approach or framework adopted by researchers to conduct a study and gather relevant data or evidence to answer research questions or test hypotheses. It involves the design, planning, and implementation of the research process, including the methods, techniques, tools, and procedures used to collect, analyze, and interpret data. This covers the systematic approach and techniques used for planning, conduct, and evaluation of research study to ensure the reliability, validity, and ethicality of the research findings. It also provides a framework for data collection and analysis for addressing research objectives and contribute to the existing body of knowledge. Research Methodology guides on how the research study will be conducted, ensuring that it is rigorous, reliable, and valid as supported by Creswell (2014). It involves making decisions about various aspects of the research, such as the research design, sampling strategy, data collection methods, data analysis techniques, and ethical considerations. The choice depends on the nature of the research objectives, the research questions, and the type of data required.

3.2 Research Philosophy

Walliman & Walliman, (2011) define research as “an activity that involves finding out, in a more or less systematic way, things you did not know” while Brown (2006) posits that “Methodology is the philosophical framework within which the research is conducted or the foundation upon which the research is based”.

O’Leary (2004) also notes that it is a framework associated with paradigmatic assumptions that are to be used when conducting research. Allan & Randy (2005) give the criteria which should be met when conducting a research methodology. First is its suitability to achieve the objectives of the research and it should be made possible to reproduce the methodology used in other research that are of the same nature. Research methodology provides the principles for organizing, planning, designing, and conducting research.

Saunders et al. (2019) define research philosophy as a set of beliefs, assumptions, and principles guiding the researcher's research approach. It provides a framework for understanding the nature of knowledge, reality, and the relationship between the researcher and the research subject. Research philosophy influences the choice of research methods, data collection techniques, and data analysis approaches used in a study as Saunders et al. (2019) posits. There are different research philosophies, each with its own underlying assumptions and implications. Some common research philosophies include positivism, interpretivism, and critical realism.

Research philosophy is associated with the nature of knowledge and the development of that knowledge. This is important as the philosophy adopted contains the worldview of the researcher and is therefore important. The Positivism Philosophy is an epistemology adopted in this research and is that of a natural scientist. Remenyi et al. (1998) points out that the researcher will use observable social reality which at the end of the research becomes a law-like generalization related to those produced by the natural and physical scientists. Positivism is based on the belief that knowledge can be obtained through objective observation and measurement. It emphasizes the use of

quantitative data, the application of scientific methods, and the pursuit of causal explanations.

When using a positivist philosophy for modeling an IDS using the A3C algorithm, the focus is on employing objective and empirical data collection and quantitative analysis, and deriving conclusions that can be generalized. Positivism emphasizes the use of scientific methods, observation, and experimentation to understand phenomena. A positivist approach involves:

1. Formulation of clear research questions by looking at objectives and goals of the research, which is improving the detection accuracy and reducing false positives in intrusion detection.
2. Gathering data in a systematic manner which involves collecting data related to network traffic, attack patterns, and system behavior by using publicly available datasets.
3. Using statistical and computational techniques to analyze the collected data. This includes data preprocessing, feature extraction, and employing A3C for intrusion detection.
4. Evaluating the performance of the IDS model by measuring its accuracy, precision, recall and F-Measure and comparing with existing IDS approaches or benchmarks so as to validate the effectiveness of the A3C algorithm. This is done objectively.
5. Generalization and replication: Using the findings to draw conclusions and determining how generalizable the results are to other contexts.

Creswell (2014) points out that positivist philosophy assumes an objective reality that is looked at using empirical methods. It is primarily focused on quantitative data, causal relationships, and generalizable findings.

Saunders et al. (2019) advices that it is important for researchers to clearly articulate their research philosophy as it shapes the overall approach and methodology of the study. The choice of research philosophy should align with the research objectives, research questions, and the nature of the phenomenon under investigation. This research was undertaken, in a value-free way with the assumption that 'the researcher is independent of and neither affects nor is affected by the subject of the research' (Remenyi et al., 1998). This was particularly applicable to this study as the research was independent of the researcher and at the end of the observation, a conclusion as a result of the findings of the study was discussed and given as a generalization.

3.3 Research Design

Creswell (2012) refers to research design as the specific procedure on how the researcher intends to conduct a study. It is about deciding on the methods, data collection procedures, sampling techniques, and data analysis approaches to be used for addressing the research questions or objectives. Research design is a reference for the research process to ensure that it is well-structured, systematic, and capable of generating reliable and valid results. It guides the research from defining the problem and formulating research questions to collecting and analyzing data and drawing conclusions.

Experimental research design was used in this study. Experimental research is a quantitative research method that involves manipulating one or more independent variables to observe their effect on a dependent variable while controlling for

extraneous factors (Creswell, 2014). Experimental research is one study that is commonly applied in scientific research designs, and it is the base approach for scrutinizing the cause/effect relationships and explore the association of one variable to the other by systematically manipulating and measuring them under controlled conditions. This kind of research is known as hypothesis testing or a deductive research method because it is used in hypotheses formulated by the researcher by supporting or disproving them to answer a specific question or issue.

A variable, according to Fraser Health Authority (2011) is “a property or characteristic of things and people that vary in quality and quantity”. It is measurable and can be manipulated and controlled. An independent variable, also called an experimental or predictor variable is a variable to be manipulated during the experiment and observed for its effect on the dependent or outcome variable. The dependent variable is dependent on the independent variable.

The independent variables were the raw data packet training datasets and the dependent variables will be the accuracy, precision and recall matrices. Machine learning models were split up into two as training and testing parts. The training data samples were used in the learning algorithm where the training took place. In testing, the learning algorithm makes use of an execution engine to predict the unknown test data used (classified data) to detect novel attacks.

The researcher intended to model an Intrusion Detection System using the Asynchronous Advantage Actor-Critic Algorithm which learns both policy and state value functions simultaneously. The reward function included, motivates the agent to go for as many rewards as possible. DNN is applied in the reward function to see if a new observation has a normal pattern. Exploration and exploitation approach was

adopted in this model by using entropy to avoid the common problem of a policy early converging to some inescapable points.

Once the development of the model was completed, training of the agent using already available data followed. On the data collection, cleaning, and benchmark part, the data to be used in the experiment were the secondary data from the UNSW-NB15 dataset. This data was loaded and cleaned in preparation for analysis in the preprocessing stage. Data preprocessing is done since such datasets contain some discrepancies. The preprocessing functions included ensuring no null values and consistent data and transformation by normalization and one hot encoding. Normalization was performed due to the heterogenous nature of network traffic logs. Categorical features were standardized by one-hot encoding and the raw values of numerical features were rescaled to the proper range.

Preparing a visualization of the dataset was important in enabling a better understanding of the dataset being used in the experiment. A benchmark was performed using traditional machine learning algorithms for prediction on the datasets with the aim of a prediction accuracy score of above 90%.

After dataset benchmarking, the model of the IDS using A3C was developed using Python language. The environment was built, while the agent (actor and critic) was developed using TensorFlow – an open-source software library for machine learning. Training of the agent on the dataset using the algorithm, testing, and analysis was done on the effectiveness of the algorithm.

3.4 Research Approach

This study employed a quantitative research approach. Bryman (2012) states that it is a research strategy that focuses on quantifying the collection and analysis of data.

This is supported by Aliaga & Gunderson (2002), who describe quantitative research as the method of gathering data in numerical form so as to explain a phenomenon and performing analysis using mathematical methods. Williams (2011) points out that quantitative research uses inquiry strategies like experiments and surveys, and performs data collection on predetermined instruments to give statistical data.

This approach is preferred by the researcher because it is objective, scientific and focused. The quantitative approach is attractive due to its speed and efficiency. This is because of using data computing equipment and software to process and analyze data, even with a large sample size. The objective of using the quantitative approach was to develop and employ a model using the A3C algorithm in the intrusion detection system.

The quantitative approach has advantages relevant to the study including objective results and minimum bias of results that can be generalizable. The quantitative data was from the values assigned to the records of the dataset and the resulting analysis done from the experiment. Quantitative data was analyzed by having frequency tables and bar charts generated so as to give a visual representation of the dataset composition and an objective and generalizable representation. This type of analysis enabled the researcher to make viable conclusions after assessing the data critically.

The study included selecting the data from the UNSW-NB15 dataset, optimizing the data and using the training and test data in analyzing the effectiveness of the algorithm. To evaluate the performance of the A3C algorithm, the main test case was classifying the record as either an attack or as normal using the features.

3.5 Deductive Reasoning

This research employed the deductive approach also called deductive reasoning in which a hypothesis is developed, and a research strategy is designed to test the hypothesis (Wilson, 2010). This is because the Deductive approach offers benefits such as:

i. It is possible to explain causal relationships between concepts and variables
ii. It is possible to quantitatively measure the concepts
iii. Research findings may be generalized.

In modelling an IDS using deep reinforcement learning, the premise was that if the performance of deep reinforcement learning is high in other fields such as robotics, finance, games, and autonomous driving, then there would be better performance in intrusion detection. The data of the UNSW-NB15 dataset was the dependent variable while the prediction matrix including accuracy, precision, recall, F-score and false positive rate was the independent variable.

Performance evaluations for the algorithm were to be done. Once the algorithm had been trained on the training data set, the test data set would be useful in calculating performance indicators for the evaluation of the algorithm's performance and quality. Important metrics to evaluate the performance of the algorithm include:

3.6 Development of an IDS model using A3C

The steps taken in the development of the IDS must be guided and hence carried out systematically for realization of results. The process flow highlights the steps to be followed.

3.6.1 Process flow

This study begins with obtaining the UNSW-NB15 dataset and performing pre-processing for the data to be optimized for training. This will be followed by modelling of the IDS using A3C then agent training and testing and finally evaluating and analysis on how the model performs. Figure 8 shows the process flow diagram

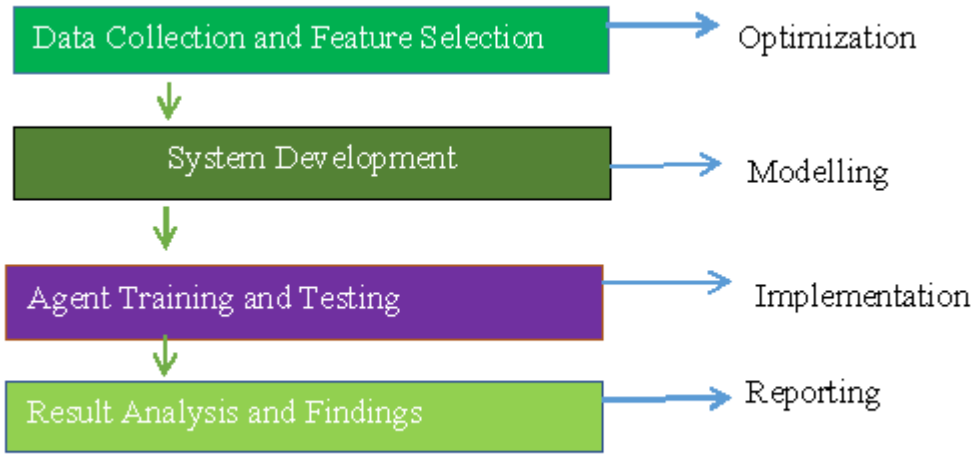


Figure 8: Process Flow Diagram
Source: Author

1. Dataset preprocessing: The UNSW-NB15 dataset needs to be preprocessed. Preprocessing made it suitable for training the agent. This process typically involves converting the dataset into a format that can be fed into the A3C algorithm and include encoding categorical variables and normalizing numerical features.
2. Environment design: In reinforcement learning, an environment is a representation of the problem or task that the agent interacts with. This case required an environment that emulates network traffic scenarios. The preprocessed dataset is taken as input by the environment and a simulation of the network environment enables the agent to make decisions.

3. Agent setup: A3C algorithm is set up by defining the neural network architectures for the actor and critic, which will learn to make decisions and evaluate the actions, respectively. The agent's policy which describes how it selects actions is updated based on the observations from the environment and feedback from the critic.
4. Training and Testing process: Multiple parallel agents are used to interact with the environment and collect experiences. These experiences, including states, actions, rewards, and next states, are used to update the agent's policy and value function. Training involves running the agents in parallel, exploring different actions, collecting trajectories, estimating advantages, and updating the network weights through backpropagation. Testing involves using unseen network traffic data from the UNSW-NB15 dataset for the agent to apply its learned policy to detect network attacks and provide responses accordingly.
5. Evaluation: Evaluation metrics such as accuracy, precision, recall, or F1 score can be used to assess the agent's performance.

3.6.2 Model Development Cycle

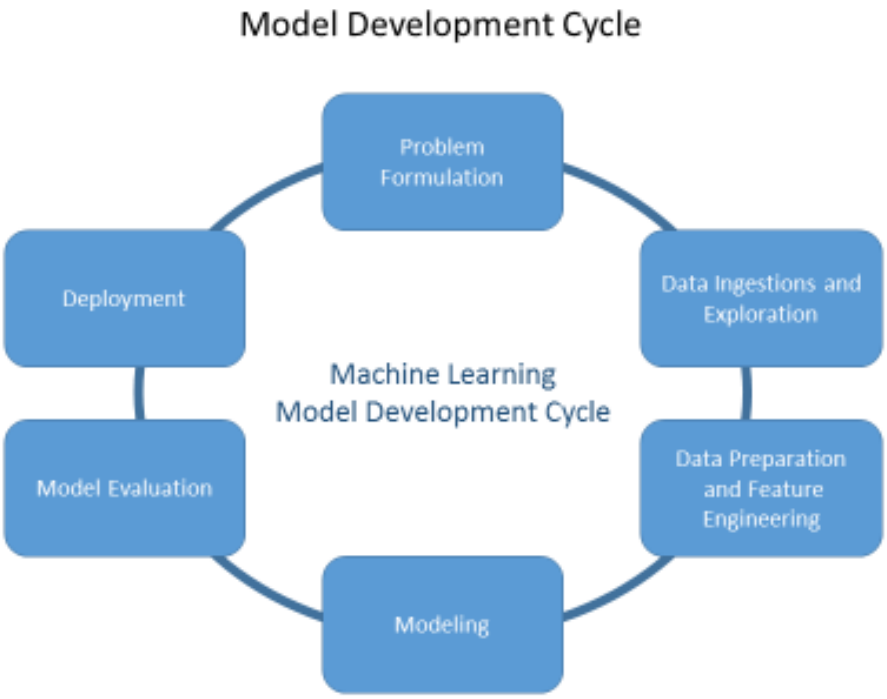


Figure 9: Model Development Cycle
Source: Packt subscription. (n.d.))

The model development cycle in figure 9 shows the steps that will be taken in modelling using A3C. After the problem is understood, data selection and preparation follow then modelling of the model using A3C. The model will then be evaluated.

Model development will follow the evolutionary prototyping methodology. This is selected as the prototype is robust and structured and more refining is done as needed. Blackwell (2015) defines a prototype as an early sample, model, or release of a product built to test a concept or process. The final prototype will be the proof of principle as the aim of the study is to use the A3C algorithm in intrusion detection. The advantage of this method is that since continuous feedback is received because of continuous testing, system completion is speeded up and it is easier to meet the requirements of the system. Figure 10 shows the steps taken in the evolutionary prototyping.

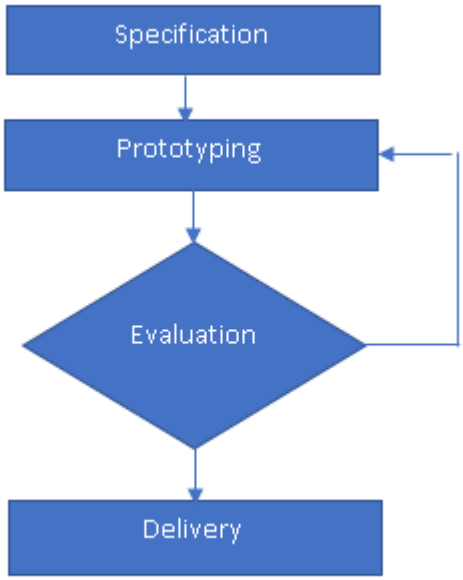


Figure 10: Prototyping Process
Source: Henderson-Sellers & Loucopoulos (2001)

In evolutionary prototyping, the initial version of the software system which is a prototype is developed to ensure key requirements have been covered. This prototype is then refined and enhanced in subsequent iterations based on feedback and changing requirements. The process continues until the final version of the software system is achieved.

Key characteristics of evolutionary prototyping as supported by Erder (2020) include:

1. Iterative Development: The software system is developed in a series of iterations, with each iteration improving upon the previous version.
2. Feedback-Driven: User feedback, testing, and evaluation play a crucial role in guiding the evolution and refinement of the prototype.
3. Rapid Iterations: The emphasis is on quick turnaround times between iterations to gather feedback and incorporate changes.
4. Incremental Development: Each iteration adds new features, functionalities, or enhancements to the existing prototype.

5. Flexible and Adaptive: Evolutionary prototyping accommodates changing requirements and allows for flexibility in design and development decisions.

This approach is advantageous in situations where requirements are not clear or can change, as it allows for a more iterative and adaptive development process leading to a software system that better aligns with user needs and expectations.

3.7 Basic metrics Terms and Formula

True Positive (TP) - Both the original and predicted data instances are true

True Negative (TN) – Both the original and predicted data instances are false

False Positive (FP) – The original data instances are false but the predicted data instances are true

False Negative (FN) – The original data instances are true but the predicted data instances are false

Accuracy – Represents the number of correct over the total data instances.

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

Precision – It is also called the positive predictive value and is the relative number of relevant (true) instances among the ones retrieved.

$$Precision = \frac{TP}{TP + FP}$$

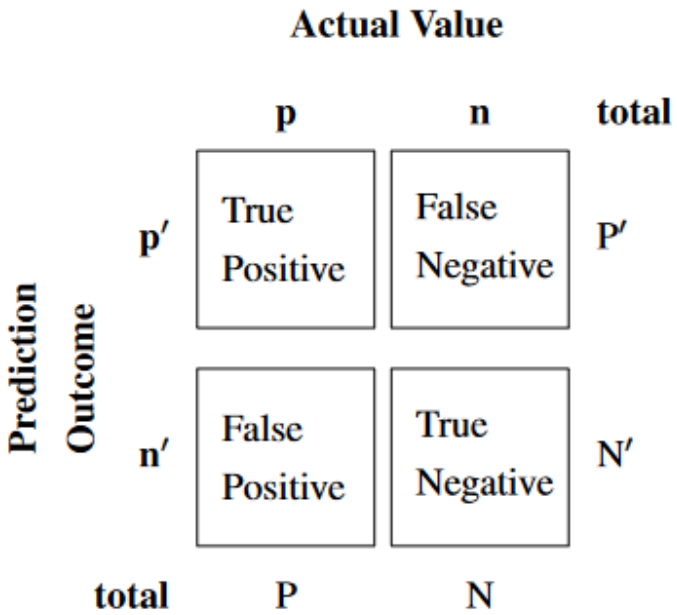
Recall – Also known as sensitivity or true positive rate, is the relative number of true (relevant) instances that were retrieved. A good IDS should have a high recall.

$$Recall = \frac{TP}{TP + FP}$$

F- Score – Also called F-Measure aims to use both recall and precision using harmonic mean to seek a balance between the two.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Confusion Matrix – This is also called a contingency table and is an N*N matrix to evaluate classification model performance. It differentiates the true positive, false positive, true negative, and false-negative predictions by comparing actual target values and the ones predicted by the machine learning model. The confusion matrix is as shown:



All these measures were used to assess how the A3C model performs in intrusion detection. The false positive, where the IDS records normal traffic as an attack, and

the False-negative where the IDS fails to record a malicious record as an attack. These should be kept as low as possible to ensure the maximum performance of the IDS.

Another measure is the false positive rate also known as False Alarm Rate and is the correspondence between the number of False Positives and the total number of actual negative events. A good IDS should have a low false-positive rate.

$$FPR = \frac{FP}{FP + TN}$$

3.8 Population and Sampling Technique

Kombo & Tromp (2006) state that the purpose of collecting data in research is to further how the researcher understands an issue. A population is a complete set of objects with a specialized set of characteristics. It refers to the entire group of individuals, objects, or events that the researcher is interested in studying. It is the larger target group to which the research findings are intended to be generalized.

A sample on the other hand is a subset of the population. Cooper & Schindler (2019) state that sampling involves selecting a subset of individuals or units from the population to be included in the research study. Since studying the entire population is often impractical or impossible due to factors such as time, cost, and accessibility, researchers use sampling techniques to gather data from a smaller, more manageable group known as the sample. The sample is expected to be representative of the larger population and should allow the researcher to make valid inferences about the population.

The choice of sampling method depends on various factors, including the research objectives, available resources, time constraints, and the characteristics of the

population. It is crucial for researchers to carefully consider their sampling strategy to ensure the validity and generalizability of their findings.

The UNSW-NB15 dataset is the population of the study. This dataset was created by Moustafa et al. (2015) at the University of New South Wales, Australia for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors. The dataset is a widely used network intrusion detection dataset developed to facilitate research and evaluation of intrusion detection systems using network traffic data. The dataset contains a large-scale and diverse set of network traffic features, including normal and malicious activities.

This is secondary data and is available on the internet and has characteristics relevant to the study including the various modern and contemporary attacks as covered in Chapter 2.8.1. The dataset was also a good choice for research due to the reliability of the dataset when compared to other datasets like the KDD99 dataset as supported by Moustafa & Slay (2016). As it is expensive to experiment on the whole population, a sample that constitutes the whole population was required.

3.8.1 Sampling Technique

Sampling is the selection of a subset of the population of interest in research. These subsets are often used to address challenges such as class imbalance, reduce computational complexity, or create a manageable sample size for analysis. Kemper et al. (2003) identify principles that govern all forms of sampling and must be adhered to including the strategy selected in sampling should sensibly come from the conceptual framework as well as the research questions being addressed by the study; thorough database generation on the type of phenomenon under study; ability to make clear conclusions and acceptable explanations from the data; ethical sampling

strategy; feasible sampling plan; generalizability; efficient and practical sampling scheme.

Purposive sampling was the method of sampling employed in this study. Purposive sampling, also known as subjective or judgement sampling uses expert judgement on selecting cases which will serve specific purposes. Researchers can obtain using sound judgement a representative sample resulting to cost and time saving (Black, 2010). This is supported by Morse & Niehaus (2009) who point out that whether the methodology being used is quantitative or qualitative, the sampling methods should point to ensure efficiency and validity.

The researcher used purposive sampling for this study as it was best suited for this experiment. This was informed by the layout of the UNSW-NB15 dataset. This dataset was grouped into four CSV files each having attack and normal records. These include UNSW- NB15_1.csv, UNSW-NB15_2.csv, UNSW-NB15_3.csv and UNSW-NB15_4.csv. The records are ordered in each CSV file according to the last time attribute. Each file of the first three CSV files contains 700000 records and the fourth file contains 440044 records. There is the ground truth table named UNSW-NB15_GT.csv and the list of event file is labelled UNSW-NB15_LIST_EVENTS which contains the attack category and subcategory. There are also the UNSW-NB15_TRAIN and UNSW-NB15_TEST which were selected and utilized in the study. These training and testing subsets of UNSW-NB15 are also provided in Moustafa et al. (2015) and the records selected were 175,341 records to form the training subset and 82,332 records for the testing subset among the original 2,218,761 records. This sample was selected since it was contained the targeted representation of the whole population and would yield the desired results. The sample was also

selected based on expert knowledge by relying on the judgment of experts who have deep domain expertise as the UNSW_TRAIN and UNSW_TEST datasets were generated alongside the whole dataset. This ensured the inclusion of samples that are representative of real-world scenarios and challenges. Purposive sampling also allows efficient use of resources by focusing on the most relevant samples. Unlike random sampling, which may require analyzing a large volume of data, purposive sampling enabled the researcher to concentrate on specific subsets of the UNSW-NB15 dataset that were more likely to contribute to the development and evaluation of the IDS using A3C.

Feature selection also called attribute or variable selection, involves selecting more relevant attributes, and removing irrelevant or less relevant attributes or noisy data or features that don't add value to a machine learning algorithm (Kumar et al., 2014). Although making use of all the dataset features is not a guarantee for maximum performance due to increase in system error rates and computational cost, the reduction of features also causes data loss. The use of a whole dataset lends more information about the dataset and is thus more valuable. This is because the probability of getting more useful information is increased and hence advantageous and relevant.

Yang & Zhu (2011) posit that when relevant features for machine learning algorithms are used, it makes the computation faster and prediction becomes more accurate. This, therefore, means that data has to be processed before being used on machine learning algorithms. Data optimization was done on the sample data to ensure good training data set and reduce the outliers which causes a low detection performance of the IDS.

3.9 Ethical Considerations

Blumberg et al. (2005) define ethics as the 'moral principles, norms or standards of behavior that guide moral choices about our behavior and our relationships with others. Research ethics involves the process of formulation and clarification of the research topic, design of the research access gaining, data collection, storage and processing of data, analysis of data and writing up of research findings in a moral and responsible manner. This study endeavored to abide by all the regulations laid out when conducting research.

Research studies using qualitative and quantitative approaches must adhere to sound ethical principles as supported by (Ong'ondo, 2009). First, permission was sought by the researcher from National Commission for Science, Technology and Innovation to conduct the study. The research permit is attached in appendix I. The researcher also ensured the originality of the work by submitting the work to antiplagiarism software and a similarity index of 5% was achieved as attached in Appendix II. The freedom to use the UNSW-NB15 dataset for academic research is expressly given, hence the researcher need not seek further permissions. Integrity was preserved as the results presented were the exact results as generated from the IDS model.

3.10 Reliability and Validity

The raw network packets of the UNSW-NB 15 dataset were created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) by Moustafa et al. (2015) for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors. The Tcpdump tool is utilized to capture up to 100 GB of the raw traffic (e.g., Pcap files). This dataset is among the most used by scientific researchers alongside the NSL-KDD dataset. It has been

deemed reliable for use in experiments and has been experimented on and used in research. The train/test split approach of validation was used where 33% was the test data while 67% was the train data. This is among the common split percentages used as pointed out by Brownlee (2016).

The validity of the A3C model was also ensured by performing a prior benchmark on the UNSW-NB15 dataset using traditional machine learning algorithms so as to check their performance. This was relevant for comparing how the traditional machine learning algorithms perform against the A3C model. The reliability of the model was ensured by the consistency of the results obtained from the experiment which was performed more than once. The experiment was conducted 4 different times and the episodes of the learning and testing process were evaluated to ensure the consistency of the results obtained.

3.11 Chapter Summary

This chapter has described in detail the methodology employed in carrying out this research. Research methodology is highly relevant to this study as it provides a systematic and structured approach to conducting research. It enabled the researcher select appropriate methods and techniques and also in collecting and analyzing data, and drawing valid and reliable conclusions. Overally, research methodology provides a systematic framework for conducting research, ensuring rigor, validity, and reliability. It guides in making informed decisions at every stage of the research process, ultimately contributing to the credibility and significance of the study's findings. Chapter four discusses the analysis and interpretation of results obtained.

CHAPTER FOUR

DATA PRESENTATION, ANALYSIS AND INTERPRETATION

4.1 Introduction

Analysis, presentation, interpretation, and discussion of the findings will be done in this chapter.

The study focused on answering the following research questions:

i. How effective is the A3C Algorithm in anomaly detection?
ii. How will the development of the IDS model using the A3C algorithm be achieved?
iii. How is the performance of the IDS model developed using the A3C algorithm?

Andrienko & Andrienko (2006) define data analysis as the process of computing various summaries and derived values from given data by studying and examining data so as to generate conclusions about the phenomenon under study using some analytic techniques.

The data analysis, presentation, and interpretation of the study based on the research questions were outlined.

i. Determining the effectiveness of using Asynchronous Advantage Actor-Critic algorithm in anomaly detection
ii. Developing an Intrusion Detection System model, based on Asynchronous Advantage Actor-Critic (A3C) Algorithm
iii. Evaluation of the performance of the model developed

4.2 Determining the effectiveness of using A3C algorithm in anomaly detection

This aimed at providing answers to the research question which sought to find how effective the A3C Algorithm is in anomaly detection. This was an area where a conceptual study was done. This was necessary since it was ideal not to engage in models that do not work since it could have amounted to a waste of limited time. However, a model could fail due to errors made, but with sufficient time it can be tried several times to ascertain its effectiveness.

In relation to this study, Sewak (2019) posits that Multiple agents of the actor-critic model could even work in parallel to interact with their individual instances of the environment thereby not only making the training faster but also removing a lot of bias and obviating large memory requirements. The parallel approach could be implemented in both synchronous and asynchronous manners. The Asynchronous Advantage Actor-Critic (A3C) implementation has been quite successful in surpassing many best scores of previous models across the Atari2600 games. On comparison with other RL techniques like DQN on the Atari 2600 games, A3C performs better as it achieves same performance as DQN with half the time and also using CPUs instead of GPUs.

Deepanshu (2020) states that Reinforcement Learning is applicable in almost every field for its automation and advancement. With the A3C having a fast-training speed and acting on both discrete and continuous action spaces resulting to an advantage when comparing the different RL algorithms.

These observations make the A3C algorithm better suited and more effective when used in anomaly detection due to its ability to perform faster and consume fewer compute resources as supported by Abbeel & Schulman (2021).

4.3 Developing an Intrusion Detection model, based on the A3C Algorithm

This section aimed at providing answers to the research question: How will the development of the IDS model using the A3C algorithm be achieved? The UNSW-NB15 dataset was selected for this study. The dataset has 45 features and is divided into UNSW-NB15-TRAIN used for model training and UNSW-NB15-TEST for testing at 67% and 33% respectively. All 45 features of the dataset were used in this study. The dataset was preprocessed and this included cleaning by ensuring no null values and consistent data and transformation by normalization and one hot encoding. The nominal features were converted to numerical for use by the machine learning models. This was implemented using sci-kit learn in Python.

The model should not train on the test data to avoid data leakage which occurs when training and a model sees information it is not supposed to hence introducing bias to the final model, leading to poor performance on the model for previously unseen data (Shabtai et.al, 2012).

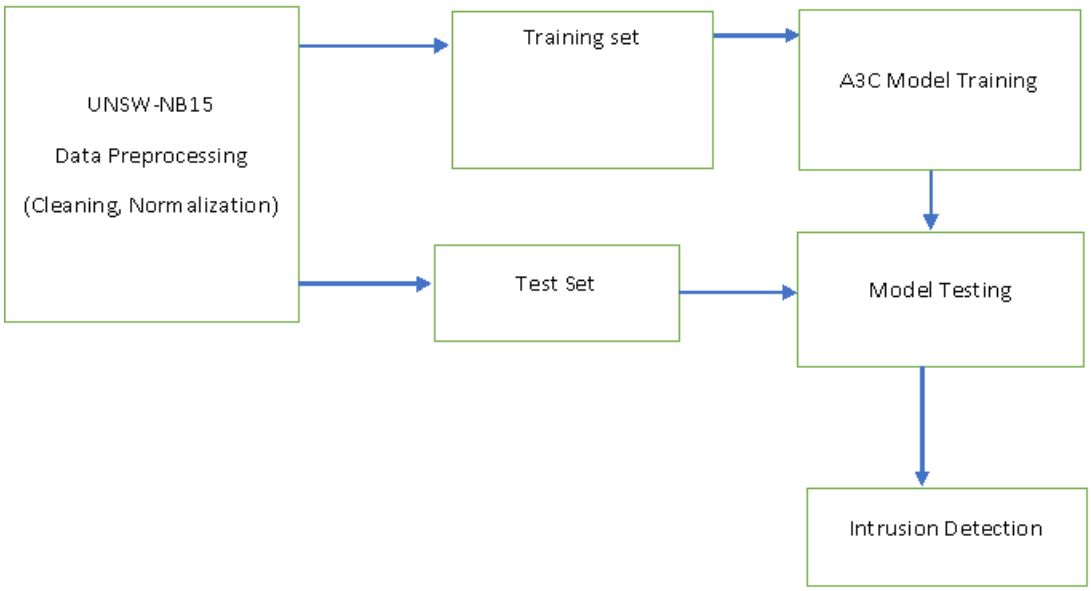


Figure 11: IDS Architecture using UNSW-NB15 dataset
Source: Author

In the proposed architectural design show in figure 11, it involves several steps, including dataset preprocessing, A3C model development, training, testing, and evaluation as explained in detail below:

1. Dataset Preprocessing - This is first done by obtaining the UNSW-NB15 dataset, which is a popular benchmark dataset for network intrusion detection. This was downloaded from the internet. Next was performing data preprocessing, which involved cleaning the dataset, handling missing values, and converting categorical variables into numerical representations for ease of use.
2. A3C Model Development - Design and implement the A3C model architecture. The A3C algorithm combines reinforcement learning and deep learning techniques. The model generally consists of two main components: the actor and the critic. The actor is responsible for making decisions (actions) based on the observed states. In IDS, this involves classifying network activities as normal or intrusive. The critic evaluates the value of each state and provides feedback to update the model's policy.
3. The A3C model was implemented using deep neural networks, as recurrent neural networks (RNNs). Necessary libraries such as TensorFlow or PyTorch, for implementing the A3C algorithm were imported and the actor and critic components of the A3C model, including their respective neural network architectures defined This also included implementing the necessary functions for model initialization, forward pass, and parameter updates.
4. Training the A3C Model - Initialize the model's parameters and hyperparameters, such as learning rate, discount factor, and exploration rate. The training subset of the preprocessed dataset was used to train the A3C

model. A3C algorithm's asynchronous training process was then employed, where multiple agents interact with the environment, compute gradients, and updates the model's parameters independently. During training, the A3C algorithm explores the environment, collects experiences, and updates the model's parameters using a combination of policy gradient methods and value function approximation. Multiple worker threads interact with the environment and update the model concurrently achieved using TensorFlow.

5. Testing and Evaluation - The model is used on the testing dataset to classify network activities as normal or intrusive. The performance metrics such as accuracy, precision, recall, and F1 score, are then calculated to assess the IDS's effectiveness in detecting intrusions. An analysis of the false positive and false negative rates is done to understand the IDS's performance in terms of misclassifying legitimate activities or failing to detect intrusions. A comparative analysis with other IDS approaches or baseline models is done to determine the effectiveness of the A3C-based IDS. Other additional criteria such as computational efficiency and response time are evaluated.

Dataset visualization is the graphical representation of information and data. This is done by visual elements like charts and graphs. Tools used in data visualization provide an accessible way to see and understand trends, outliers, and patterns in data. The most important features of the dataset featured is shown in the figures below.

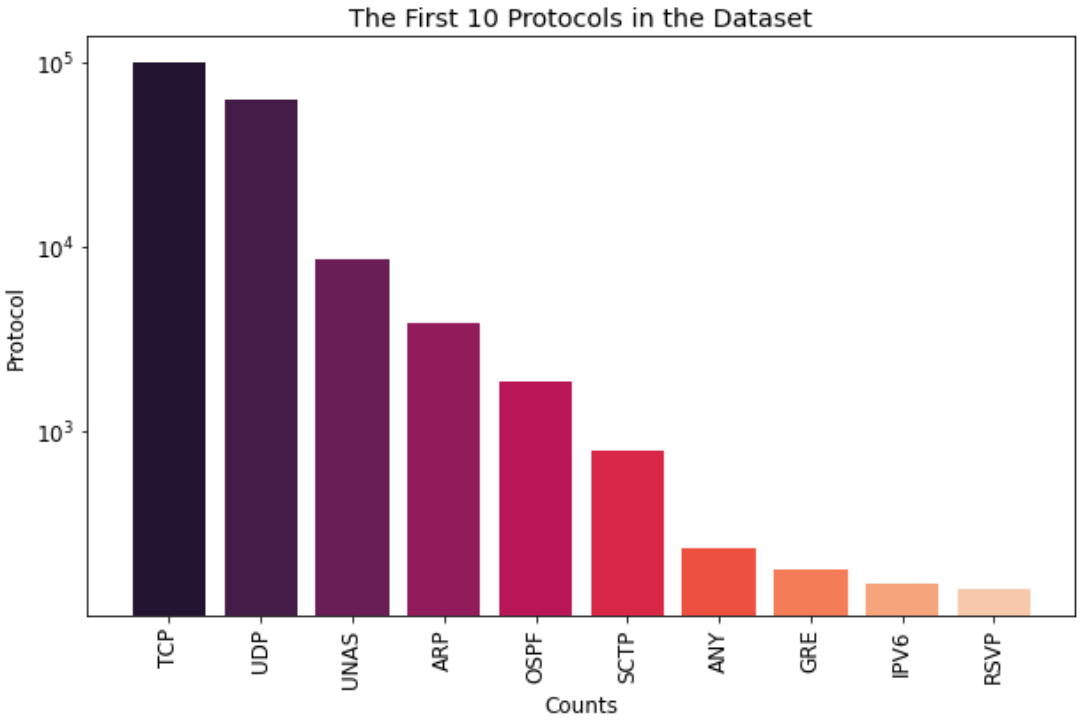


Figure 12: Protocols in the UNSW-NB15 dataset

The first two protocols; Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are the most common and cover over 90% of the total protocols included in the dataset the rest follow by distribution as shown in figure 12.

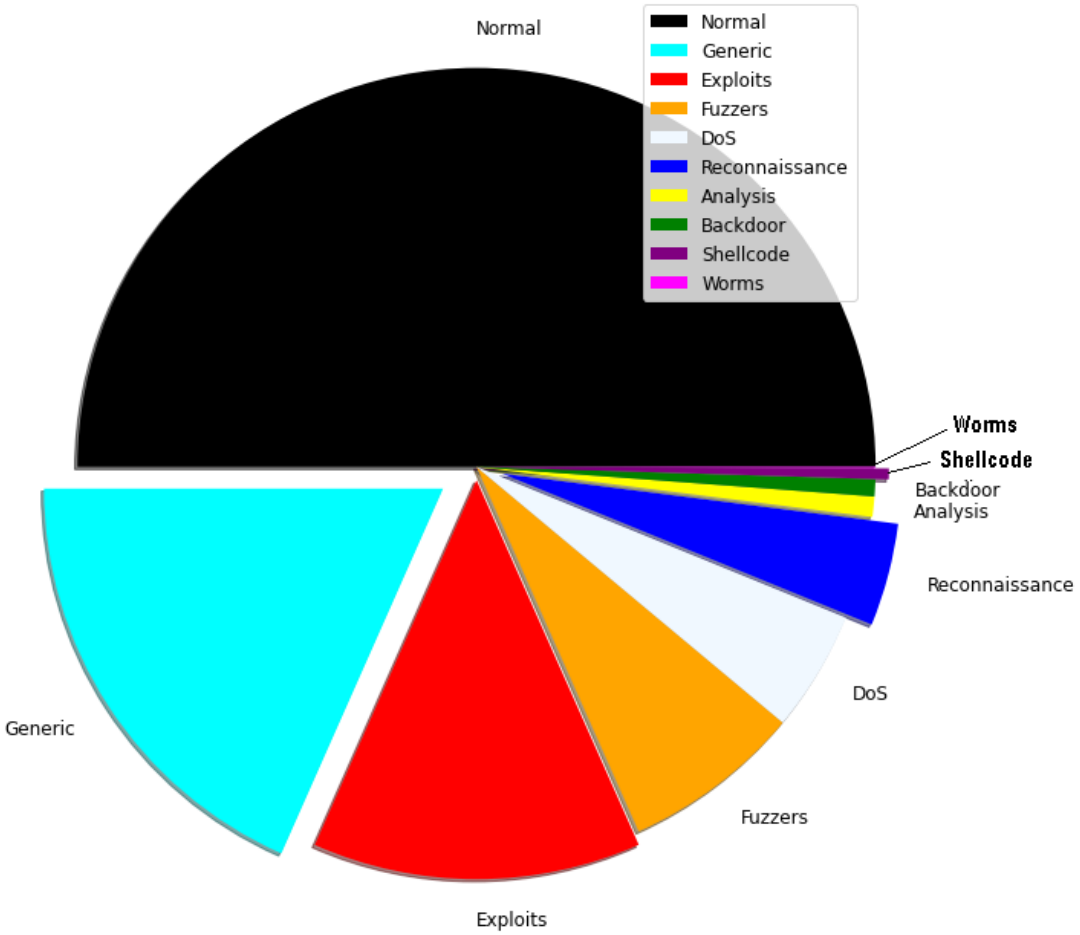


Figure 13: Distribution of attacks in the UNSW-NB15 dataset

The UNSW-NB15 dataset has 9 families of attacks and normal. The normal takes around a half while the rest is distributed among the different attack types with Generic attacks, exploits and fuzzers taking the biggest share in the distribution as shown in figure 13 above.

Model development was done using Python programming language with the help of Pycharm Intergrated Development Environment (IDE) and Jupiter Notebook which is an online interactive computing platform. The requirements for the model development are attached in Appendix II. The model included using 3 layers. An entropy was added to the loss to encourage exploration. The environment state and reward were looked at with the reward being +1 for the agent. In the agent, 2 workers were running in parallel, running episodes locally and updating globally. A reward

discount factor of 0.99 was also included to encourage exploration and exploitation since a lower discount factor results in higher reward variance encouraging greedy policy and hence less exploration while a higher discount gives importance to future rewards and this results in a balance of exploration and exploitation.

4.4 Evaluation of the performance of developed model

This step aimed at providing answers to the research question stating: How is the performance of the IDS model developed using the A3C algorithm? This was achieved by using the following in evaluating the model's performance;

1. Performance Metrics: The performance of the IDS was measured using standard evaluation metrics such as accuracy, precision, recall, and F1 score to assess the ability of the IDS to correctly classify normal and intrusive network activities.
2. False Positive and False Negative Rates: Analysis is done using the rates of false positives and false negatives produced by the IDS. The lower the rates, the more effective the IDS.
3. Benchmark Datasets: Testing the IDS using publicly available benchmark datasets commonly used in intrusion detection research and compare the performance of the A3C-based IDS against other existing IDS approaches.
4. Comparison with Baseline Models: The performance of the A3C-based IDS was compared against baseline models. This was so as to evaluate the improvement achieved using the A3C algorithm.

4.4.1 Traditional Reinforcement Learning Algorithms

The traditional reinforcement learning algorithms were used as a benchmark. These were the Decision Tree Classifier (DTC), Random Forest Classifier (RFC), Extra

Trees Classifier (ETC), and Gradient Boosting Classifier (GBC). Table 3 shows the performance of each algorithm.

Table 3: Traditional Machine Learning Algorithms Results

Method			**Normal**			**Attack**		
	Acc	**FPR**	**TPR/ Recall**	**Precision**	**F-Score**	**TPR/ Recall**	**Precision**	**F-Score**
DTC	93.6	0.065	0.94	0.94	0.94	0.94	0.94	0.94
RFC	95.3	0.063	0.97	0.94	0.95	0.94	0.97	0.95
ETC	95.2	0.065	0.97	0.94	0.95	0.94	0.97	0.95
GBC	94.5	0.067	0.96	0.93	0.95	0.93	0.96	0.94

All traditional reinforcement learning algorithms performed well as their accuracies were above 90% and had a low false-positive rate. The traditional reinforcement algorithms are however limited in their performance as they may have a bias towards the majority class, leading to reduced detection of rare intrusions. The algorithms can also be sensitive to noisy or irrelevant features and this negatively impacts their performance. The datasets can also have noisy or redundant features, making it difficult to separate relevant patterns from noise. Deep reinforcement learning (DRL) algorithms can automatically learn relevant features and handle noisy data more effectively. These baseline algorithms also have limitations in capturing complex relationships and non-linear dependencies present in a dataset while deep learning algorithms have the capability of modeling patterns that are crucial in detecting sophisticated and evolving intrusion patterns.

4.4.2 Proposed Model using A3c Algorithm

The dataset was randomly shuffled to avoid the similarity of the samples in the groups. The datasets were used as environments for reinforcement learning and are divided into train and test sets and have 42 features. Datasets were grouped into

27900 samples each to be run episodically. The dataset includes instances of both normal and malicious activities to train and evaluate the IDS. The dataset was then divided into data sequences of 27900 each representing a discrete event window. The IDS was then evaluated on each episode as it receives the sample as input and makes decisions on whether the activities within the sample are classified as normal or malicious. The IDS then gives output scores for the classification.

Based on the IDS's predictions and the ground truth labels of the samples, evaluation metrics including accuracy, precision, recall, and F1 score were then calculated to assess the performance of the IDS. The process is an iterative process and is repeated for each sample in the dataset for a comprehensive evaluation of the IDS across various scenarios and time periods. This approach provides insights into how well the IDS can generalize to new and unseen data.

Results of accuracy, precision and recall were obtained from the episodes. For all four experiments conducted, there was similarity in the progression of results with improvement with each trial. The graph of accuracy against the number of episodes represents the performance of the system over the course of training or testing. The accuracy is used to evaluate the effectiveness of an IDS, indicating the proportion of correctly classified instances out of the total instances.

When interpreting the graph of accuracy against the number of episodes the following aspects are considered as supported by Murphy (2012).

1. The progress of training: In the initial stages of training, the accuracy may be relatively low as the model is still learning to understand the patterns and characteristics of normal and anomalous behavior. As the training progresses,

the accuracy is expected to increase gradually, reflecting the model's ability to make more accurate predictions.

2. Learning Rate: The rate increase in accuracy and stabilization shows the learning capacity of the model. A steep increase in accuracy indicates that the model is learning quickly, while a slower increase shows gradual learning.

The accuracy against the number of episodes graph provides a visual representation of the IDS model's performance during training or testing. It enables the evaluation of the learning progress and assess the capability of the model to detect intrusions accurately.

4.4.3 Experimental Results

Test 1

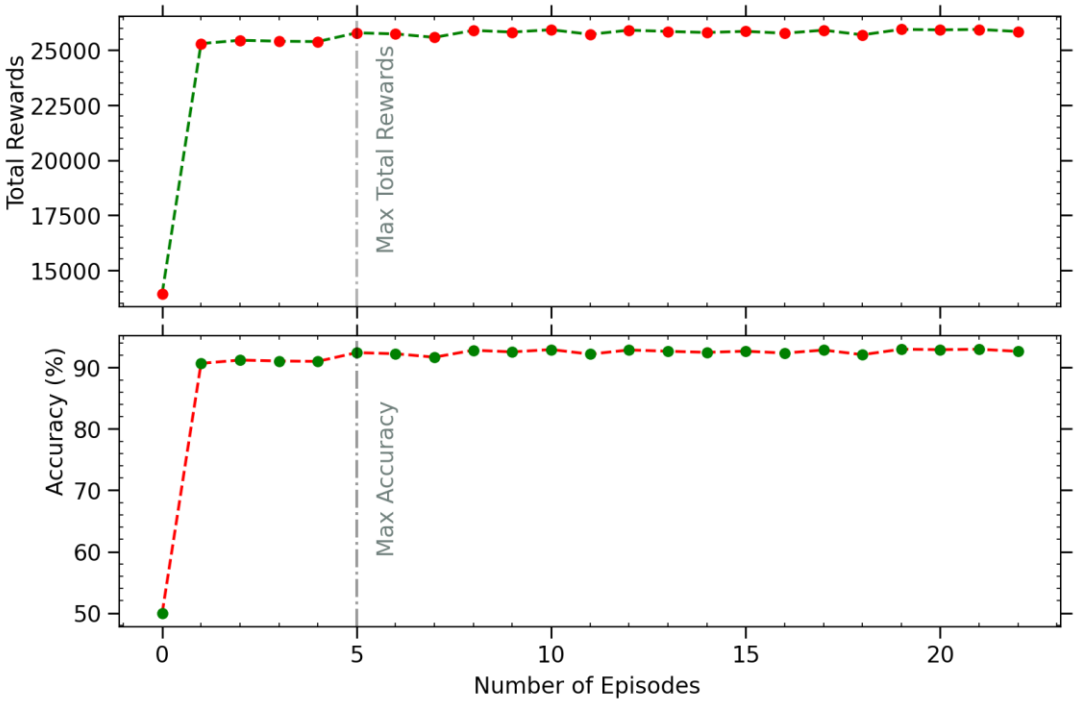


Figure 14: First run accuracy and reward distribution using collab

```
Total reward: 25789 | Number of samples: 27900 |
       Estimated Correct  Total F1_Score
normal     14879   13312  13856  92.6536
DoS        13021   12477  14044  92.2003
The PostScript backend does not support transpar
The PostScript backend does not support transpar
Performance measures on Test data
Accuracy =  0.92
```

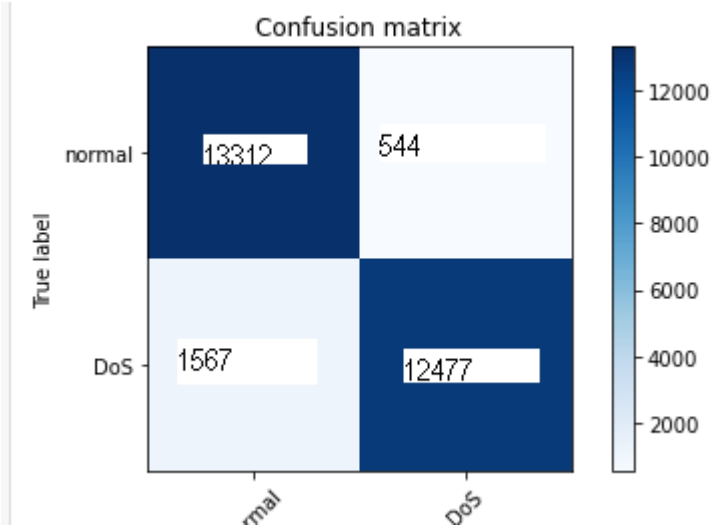


Figure 15: Test 1 result and confusion matrix

Observation

The experiment was run virtually on google collab, with assigned 16GB of RAM and processing power. There was no major effect on other computer operations while the experiment was running. The number of episodes that ran in the first test was small taking only 22 episodes before being stopped. This was because the researcher was looking at the performance and the requirements for conducting the experiment. An accuracy of 92% was achieved as seen in figure 14. Figure 14 shows the performance of the agent using accuracy as the agent progressed with the episodes. A distribution of the rewards given to the agent as per the performance of the agent is also visualized. Figure 15 displays the confusion matrix for the test.

Test 2

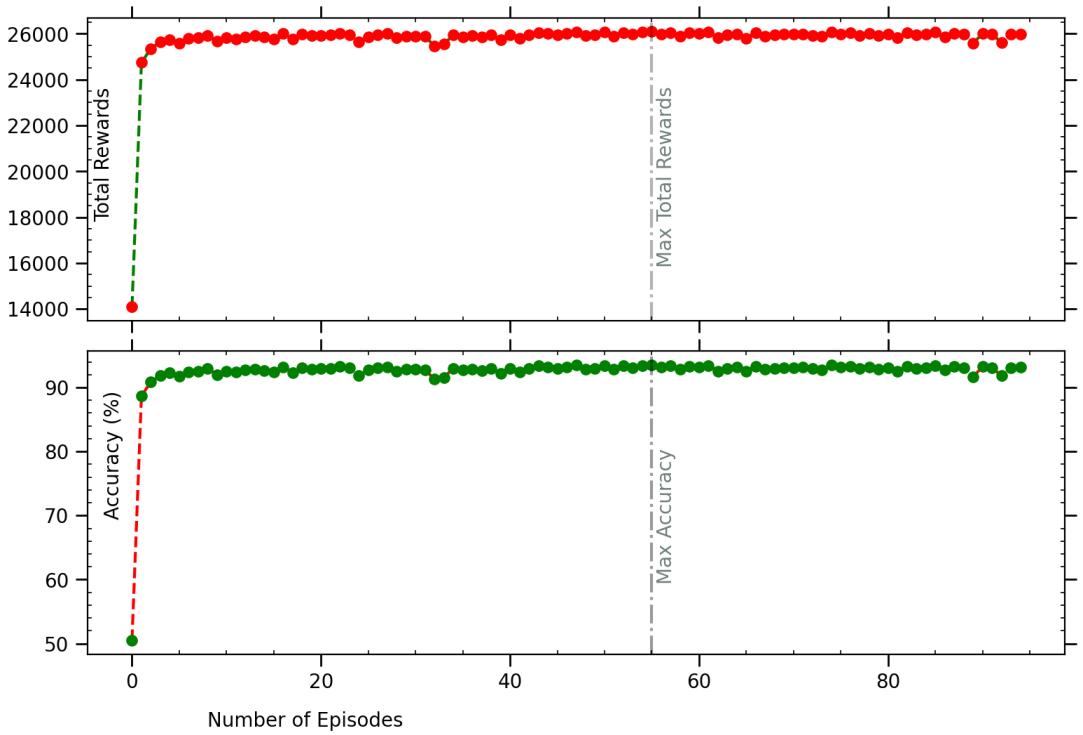


Figure 16: The Second run results on accuracy and rewards

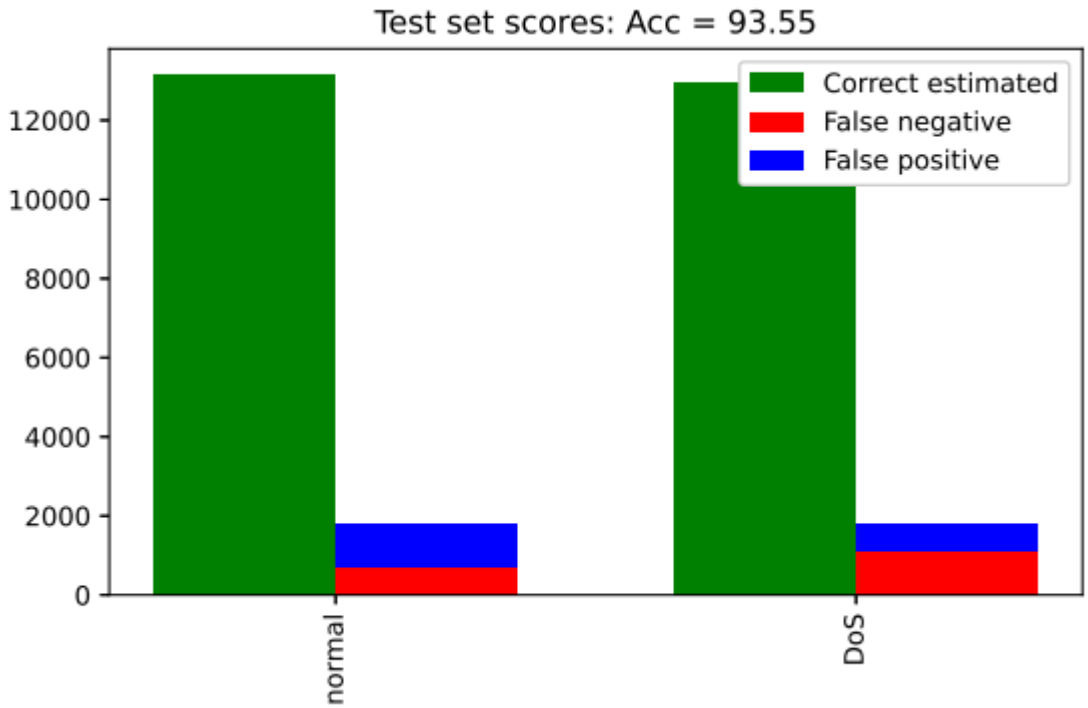


Figure 17: Test set Score

```
Total reward: 26100 | Number of samples: 27900
       Estimated Correct  Total F1_Score
normal     14244   13150  13856  93.5943
DoS        13656   12950  14044  93.5018
The PostScript backend does not support transpa
The PostScript backend does not support transpa
Performance measures on Test data
Accuracy =  0.94
F1 =  0.94
Precision_score =  0.94
recall_score =  0.94
Confusion matrix
[[13150.   706.]
 [ 1094. 12950.]]
```

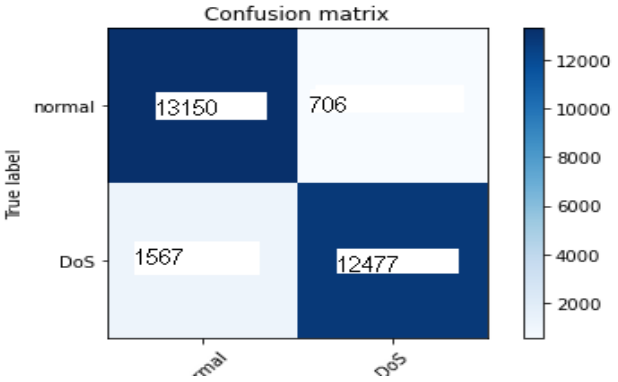


Figure 18: Test 2 result and confusion matrix

Observations

The researcher noticed that with the increased number of episodes, accuracy also increased. This experiment was done using Google collab. The number of episodes run was 94 giving an accuracy of 93.55%. This experiment was run over a period of 12 hours. Figure 16 also shows the distribution of accuracy over the 94 episodes allowed to run. From the figure a consistency with the rewards of the agent can be seen. Figure 17 shows the test set score for the accuracy of 93.55% attained by the agent and the matrices distribution for the positively identified, false positive and false negative. Figure 18 shows the confusion matrix for the performance.

Test 3

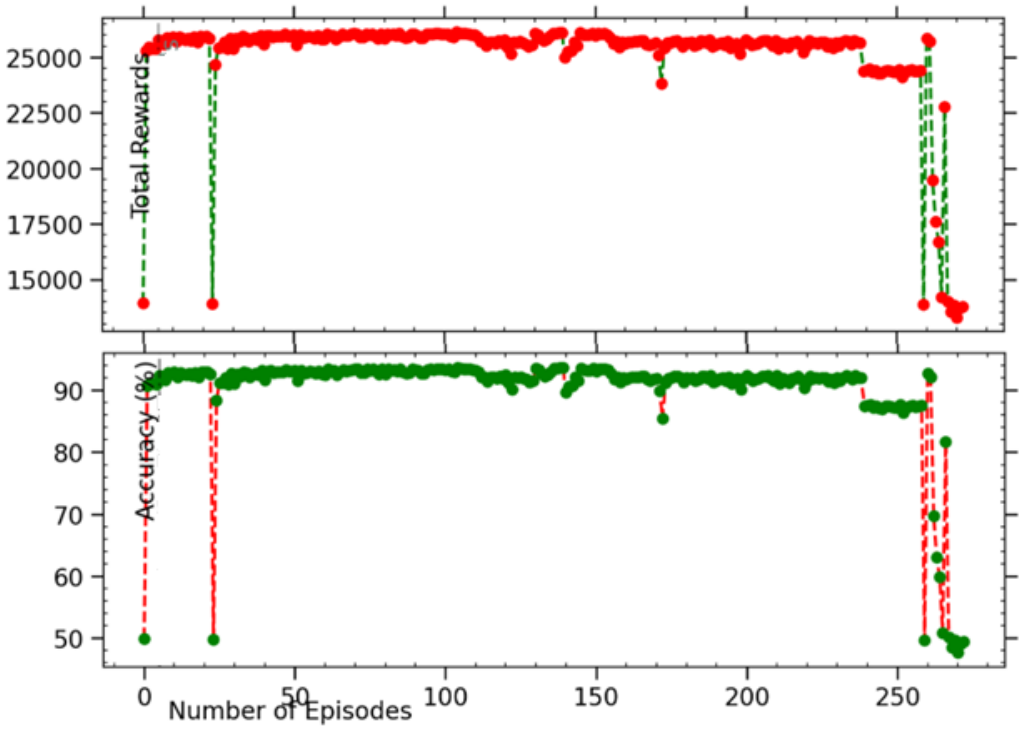


Figure 19: 3rd Run Reward and Accuracy Distribution (Time Taken- 36h)

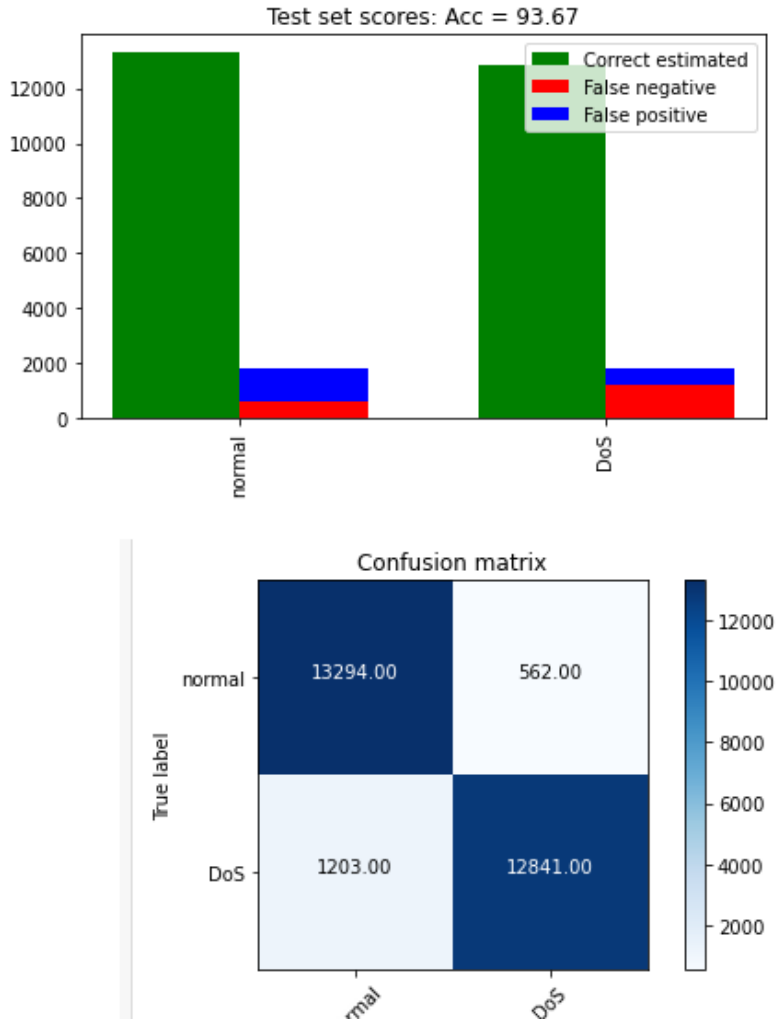


Figure 20: Result and Confusion Matrix

Observation

In this part of the experiment, the time taken for the run was approximately 36 hours which was the longest time of all the experiments. The number of episodes was 274 with the distribution of episodes shown in figure 19. Figure 20 shows the test set score for the accuracy attained by the agent and the matrices distribution for the positively identified, false positive and false negative. The figure also describes the confusion matrix for the result obtained.

This experiment was done on the local machine with 8Gb RAM,4GBGPU, 500GB HDD, i5@2.2GHz using Anaconda running Notebook. The accuracy achieved was 93.67% which was higher than the previous experiments done. The overall consumption of computer resources during this experiment was average as the researcher could use the computer for lighter tasks like document processing but heavier tasks like video editing using Adobe which uses more RAM and GPU could not be done.

The problem of overfitting was encountered due to the excess running of the algorithm on the dataset. This is as seen by the accuracy plateau where there were no more substantive improvements in accuracy and the accuracy fluctuation which caused significant variations on the episodes as seen in Figure 21 which eventually led to convergence at the end. This problem was handled by using the early stopping approach as shown in the final experiment performed in figure 23 below. There was is an option of increasing the size of the training dataset as this often mitigates overfitting. This is because with a larger and more diverse dataset, the model has a better chance of capturing the underlying patterns without memorizing noise. This option was however limited because the larger dataset required also increases the model requirements in terms of resources which was limited in this study.

Test 4

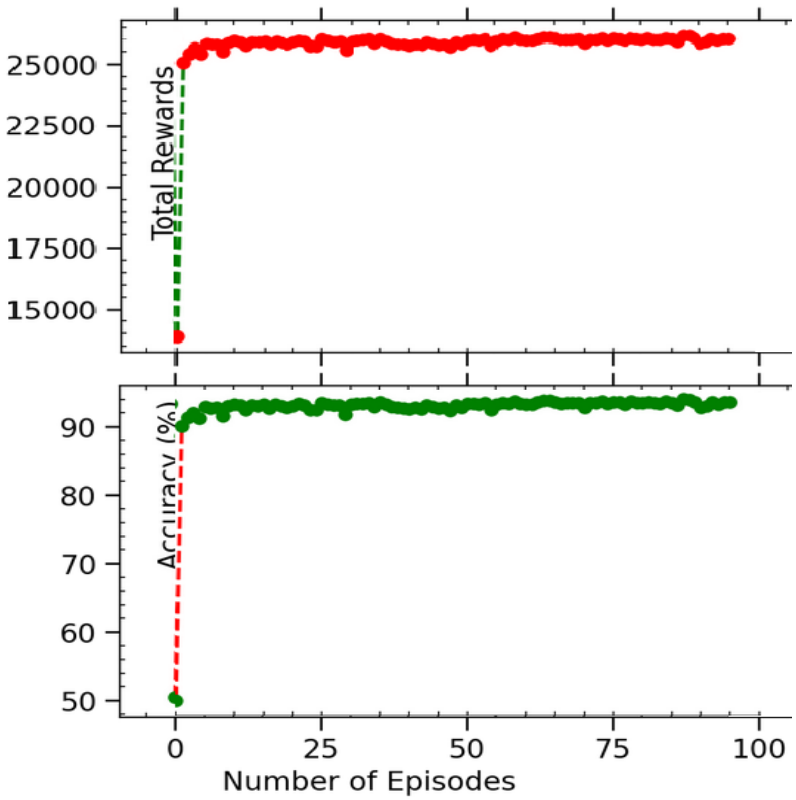


Figure 21: 4th experimental result with no dataset balancing (6hrs 13 mins)

Observations

Overfitting problem was dealt with by early stoppage as one of the regularization techniques. The final accuracy result achieved was 93.8%. This was the highest accuracy achieved in this experiment with the distribution shown in figure 21.

```
Total reward: 26171 | Number of samples: 27900
       Estimated Correct  Total F1_Score
normal     14381   13254  13856  93.8768
DoS        13519   12917  14044  93.7271
/content/drive/My Drive/Anomaly-ReactionRL/est:
  fig, ax = plt.subplots()
The PostScript backend does not support transpa
The PostScript backend does not support transpa
Performance measures on Test data
Accuracy =  0.94
```

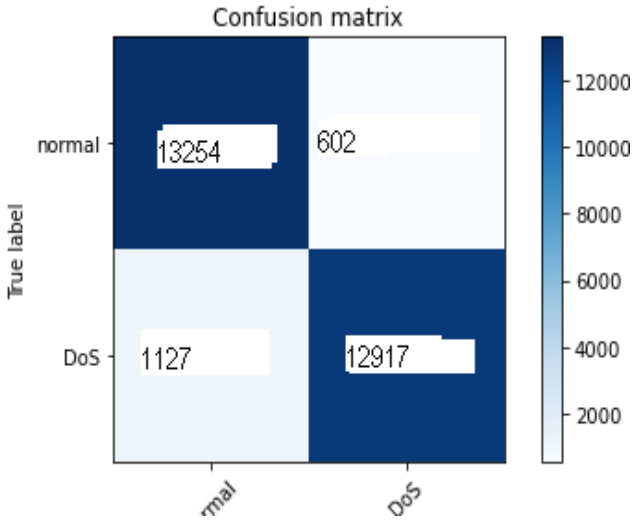


Figure 22: Best episode performance and confusion matric

Figure 22 above shows the confusion matrix for the performance of the last experiment done. In the first episode, the agent initially begins with an accuracy of 50.51% as shown in figure 23 below.

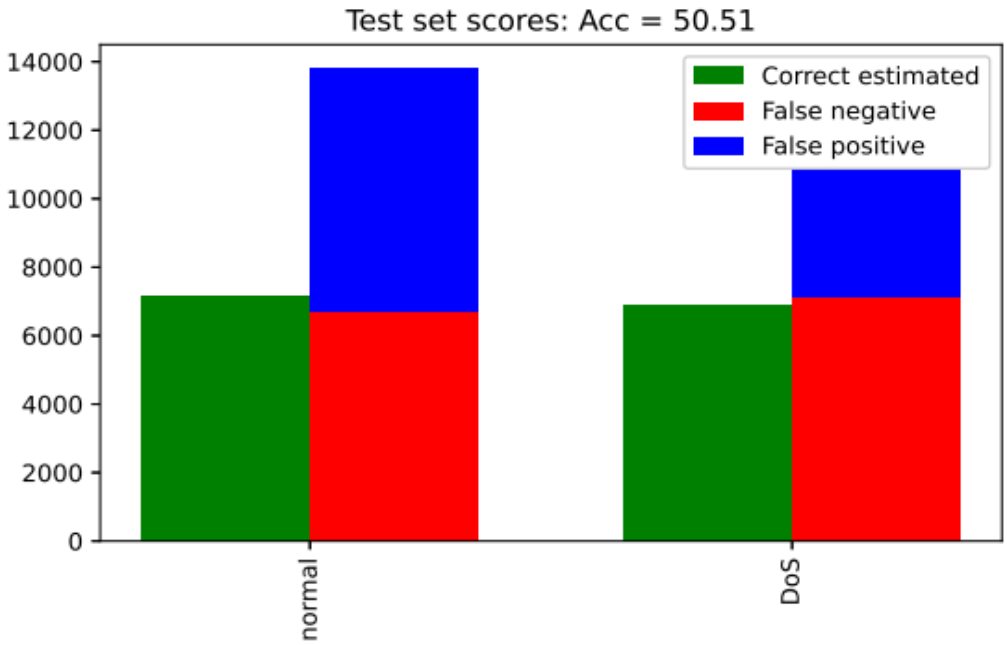


Figure 23: Test result first run accuracy

As the agent continued in the subsequent episodes, the accuracy increased to above 90% with the highest being 93.8%. This shows how quick the agent learns the policy.

Table 4 below shows the performance of the model across all the 4 experiments by presenting all the matrices including Accuracy, False positive rate, Precision, Recall and the F-Score.

Table 4: Performance of A3C in the Experiments

A3C MODEL	Acc	FPR	Precision	Recall	F-Score
Test 1	92.4	0.04	96.1	89.5	92.67
Test 2	93.55	0.05	94.9	92.3	93.6
Test 3	93.67	0.04	95.9	91.7	93.8
Test 4	93.8	0.04	95.7	92.2	93.9

The agents quickly learn the optimal policy and maintain the policy to the end of the experiment hence the performance was not improved. Discovering new policies quickly is essential to attaining a higher accuracy rate. The method is an alert type and does not stop traffic from going through hence false positives will not be dropped. This means that the false-positive impact is less than that of the false negative.

Use of deep neural networks helps in learning feature representations directly from raw data, eliminating the need for extensive feature engineering. They have the potential to capture complex relationships and dependencies in the data more effectively, resulting in improved detection performance. DRL algorithms can combine the power of deep learning and reinforcement learning to learn effective policies for intrusion detection.

4.5 Comparison of the Model with Similar Works

Table 5: Model comparison with published works

	Author	Method	Dataset	Accuracy	Precision	Recall
1	Hosseini et al. (2017)	Ramp KSVCR	UNSW-NB15	93.52	N/A	N/A
2	Hsu & Matsuoka (2020)	DQN	UNSW-NB15	91.8	93.2	91.7
3	Baig et.al (2017)	Cascade ANN	UNSW-NB15	86.4	86.74	93.4
4	Niyaz et al. (2015)	Self-Taught Learning	UNSW-NB15	88.2	83.1	98.7
5	Alavizadeh et.al (2022)	DQL	NSL-KDD	78	N/A	N/A
6	**Proposed**	**A3C**	**UNSW-NB15**	**93.8**	**95.7**	**92.2**

A comparison of the A3C model developed was made with other published works in the same field as shown in Table 5. The comparison result showed the proposed model to have better accuracy and precision scores than the rest. However, Niyaz et al. (2015) model on Self Taught Learning performed better on recall.

4.6 Performance of the Model on Computing Resources

The first experiment was done on google compute, with an allocation of 12GB of Ram as there was some uncertainty on the performance of the algorithm on the PC and fear of overheating. After monitoring the performance on Google Compute, the statistics showed that the amount of RAM consumption was at most 4-6GB. The experiment was then repeated on the laptop PC to look at the resource use of the

algorithm. The result was that the memory usage of the PC throughout the experiment gave an allowance to continue with other lighter tasks like Microsoft Office with ease. Only when software that required more RAM of above 6GB was run that the computer heat up and became slow. The processing power of the CPU affects the training and inference speed of the A3C model hence a more powerful CPU will generally result in faster computations. Sufficient RAM is also necessary to store and manipulate the dataset and model parameters. Insufficient RAM can lead to performance bottlenecks or out-of-memory errors as supported by Geron (2019).

Other key factors that affect the performance of the model include;

1. Software and code Optimization: optimized libraries and frameworks are used to train the A3C model like TensorFlow to speed up computations and ensure that the software and libraries used in the A3C model development were compatible with the laptop's operating system and hardware. Optimized code for the A3C algorithm is important for the efficient use of hardware resources.
2. Model Architecture and Hyperparameters: The complexity of the A3C model architecture affects both training time and memory usage. More complex architectures require more computational resources to train and may result in longer training times. The choice of hyperparameters, such as learning rate, batch size, and network size, can significantly impact the performance of the A3C model. Proper tuning and optimization of hyperparameters are crucial for achieving the best performance on a given hardware setup.

4.7 Limitations of the Model and Potential improvement

When developing an A3C IDS model using the UNSW-NB15 dataset, there were several limitations encountered, both from the model and on the dataset used including;

1. Lack of Real-World Diversity: Though the dataset contains real-world and synthetic attacks, like many other publicly available IDS datasets the UNSW-NB15 dataset, may not fully represent the diversity and complexity of real-world network traffic. This is because it was generated in a controlled environment, which may not capture the full range of network behaviors and attack scenarios encountered in actual network environments.
2. Generalization to Other Datasets: The performance of an A3C IDS model trained on the UNSW-NB15 dataset may not generalize well to other datasets or real-world scenarios. The characteristics of different network environments, network protocols, and attack patterns can vary significantly, and a model trained on one specific dataset may struggle to adapt to new and unseen data.
3. Scalability: A3C models, in general, can face challenges with scalability when applied to large-scale IDS deployments. The training period may increase significantly as the dataset size or complexity of the network environment grows, which can be impractical for real-time IDS applications.

The following areas of improvement are suggested for better performance of the model;

1. Augmenting the UNSW-NB15 dataset with additional real-world network environment data or other similar datasets to enhance the diversity of network

traffic and attack scenarios. Multiple datasets can be combined to create a more comprehensive and diverse training set.

2. Optimizing the implementation of the A3C algorithm by utilizing distributed computing frameworks or parallel processing techniques to improve its scalability.

4.8 Chapter Summary

This chapter presented the findings of the study. The analysis and interpretation of the results were covered based on the objectives of the study. The result was that A3C algorithm was applied in Intrusion Detection System and its performance in detection was good even when compared to other algorithms applied in the Intrusion Detection Systems. Although the objectives of the research were realized, there were other factors that limited the model's efficiency and some areas that needed improvement suggested in the implementation of the model suggested. The next chapter presents the summary of findings, conclusion and recommendations of the study.

CHAPTER FIVE

SUMMARY OF FINDINGS, CONCLUSION AND RECOMMENDATION

5.1 Introduction

This chapter presents the summary of the research work that was done, the conclusions drawn and the recommendations made as a result of this study.

5.2 Summary of Findings

The findings of this study were as follows:

- With more connected devices on the internet and more threats being witnessed more attention is being focused on deep reinforcement learning as a solution in intrusion detection systems. The A3C algorithm if successfully implemented in IDS can post an improvement in the accuracy and precision rates in anomaly detection.
- Using A3C in Intrusion detection systems is advantageous as the computing resources requirements are minimal and this enables flexibility and affordability of use since heavy processing power is not required. This is because the laptop PC used was able to realize the objective of the study. However, a laptop PC may have limitations due to hardware constraints compared to using dedicated high-performance computing resources. Other factors like software optimization, code optimization an hyperparameters must be adhered to.
- The accuracy and precision rates achieved show that A3C has the potential to be applied in intrusion detection systems and can achieve good results. Although higher levels of accuracy were not attained, this study has proved its wide-area applications.

5.3 Conclusion

IDSs are important in protecting computer networks from security threats which are becoming more complex. There is a need for advanced IDS models. The A3C algorithm, originally developed for reinforcement learning, has gained attention for its potential application in IDS.

The quantitative research approach and experimental research design were employed in this research in training and evaluating an A3C IDS model using the UNSW-NB15 dataset. The dataset was preprocessed to handle missing values and have a suitable dataset. The A3C model's hyperparameters, including learning rate, discount factor, and exploration rate, were to ensure optimum performance.

In evaluating the performance of the A3C IDS model, various metrics were considered including accuracy, false positive rate, precision, and F1 score. A comparison was performed against other models to assess the performance of the A3C algorithm. The experimental results of the study with an accuracy of 93.8% demonstrated that the proposed A3C model can learn effectively from the environment in an autonomous manner and it has good detection accuracy.

The application of A3C algorithm in IDS shows a promising direction for IDS research as it has the potential for improved intrusion detection capabilities. More research can be advanced in the field of IDS so as to enhance network security in the face of evolving cyber threats.

5.4 Recommendations

Based on the findings and conclusions of the study, the researcher found out that Intrusion detection systems using Deep Reinforcement Learning methods is a field that requires more research due to its potential to solve the current problems faced.

Adoption of the A3C algorithm in developing intrusion detection systems is encouraged as it presents the accuracy of detection and consumes less compute resources.

The study recommends using larger sets of data which are more diverse and a reflective of real network traffic and allow the agent to run more steps. In order to realize more optimal results, more agents should be used and the agents should learn various optimal policies so as to choose the best. This is important in enhancing scalability in large and complex networks.

The study also recommends looking at online learning techniques that enables updating the models in real time while still prioritizing accuracy. This will enable quick adaptation to evolving threats.

The security of the model itself should also be protected and hence more techniques should be explored so as to ensure that the integrity of the IDS is not compromised

5.5 Suggestions for Further Research

Future research should look at ways of addressing the challenges in practical implementation of IDS in real-world systems which include integration with existing network infrastructure, sensor placement, and deployment strategies. The deployment aspects and associated challenges should be extensively addressed. The algorithm's practical application also depends on investigating its scalability and how it performs in real-time network situations.

Future studies should also look at implementation of A3C based IDS on IoT devices by optimizing computational and memory consumption while maintaining accuracy of the systems

REFERENCES

Abbeel, P., & Schulman, J. (2021). *Deep Reinforcement Learning*. MIT Press.

Alavizadeh, H., Alavizadeh, H., & Jang-Jaccard, J. (2022, March 11). *Deep Q-Learning Based Reinforcement Learning Approach for Network Intrusion Detection*. MDPI. Retrieved December 19, 2022, from https://www.mdpi.com/2073-431X/11/3/41

Aliaga, M., & Gunderson, B. (2002, April 1). *Interactive Statistics*. Prentice Hall.

Allan, AJ, Randy, LJ, 2005, *Writing the Winning Thesis or Dissertation. A Step-by-Step Guide*, Corwin Press, California

Alrawashdeh, M., Aldwairi, M., Alhasanat, M. B., & Alarifi, A. (2019). Intrusion Detection Systems in Computer Networks: Techniques and Challenges. *Journal of Information Security and Applications*, 46, 157-175

Ambusaidi, M. A., He, X., Nanda, P., & Tan, Z. (2016, October 1). Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm. *IEEE Transactions on Computers*, *65*(10), 2986–2998. https://doi.org/10.1109/tc.2016.2519914

Anderson, James P., "*Computer Security Threat Monitoring and Surveillance*," Washing, PA, James P. Anderson Co., 1980.

Andrienko, N., & Andrienko, G. (2006, March 28). Exploratory Analysis of Spatial and Temporal Data. In *A Systematic Approach*. https://doi.org/10.1604/9783540311904

Awad, M., & Alabdallah, A. (2019, September 30). Addressing Imbalanced Classes Problem of Intrusion Detection System Using Weighted Extreme Learning Machine. *International Journal of Computer Networks & Communications*, *11*(5), 39–58. https://doi.org/10.5121/ijcnc.2019.11503

Baig, M. M., Awais, M. M., & El-Alfy, E. S. M. (2017, March 29). A multiclass cascade of artificial neural network for network intrusion detection. *Journal of Intelligent & Fuzzy Systems*, *32*(4), 2875–2883. https://doi.org/10.3233/jifs-169230

Barocas, S., & Selbst, A. D. (2019). Datasets Over Algorithms. *University of Pennsylvania Law Review, 167*(3), 1-59.

Bergstra, J., & Bengio, Y. (2012). *Random search for hyper-parameter optimization. Journal of Machine Learning Research,* 13(Feb), 281-305.

Bhosale, R., Mahajan, S., and Kulkarni, P. (2014). Cooperative machine learning for intrusion detection system. *International Journal of Scientific and Engineering Research*, 5(1), 1780-1785.

Bishop, C. M. (2006, August 17). *Pattern Recognition and Machine Learning*. https://doi.org/10.1007/b9479810.1007/978-0-387-45528-0

Black, K. (2010, January 11). *Business Statistics - Contemporary Decision Making*.

Blackwell, A. H.; Manar, E., *eds. (2015). "Prototype". UXL Encyclopedia of Science (3rd ed.)*

Blumberg, B., Cooper, D.R. And Schindler, P.S. (2005) *Business Research Methods*, Maidenhead, McGraw-Hill.

Brown, R. B. (2006, February 15). Doing Your Dissertation in Business and Management. In *The Reality of Researching and Writing*. https://doi.org/10.1604/9781412903516

Brownlee, J. (2016). *Machine learning mastery with python: Understand your data, create accurate models, and work projects end-to-end.*

Bryman, A. (2012, January 19). *Social Research Methods*.

Chandrashekar, G., & Sahin, F. (2014, January). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024

Chandola, V., Banerjee, A., & Kumar, V. (2009). *Anomaly detection: A survey*. ACM Computing Surveys (CSUR), 41(3), 15.

Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper. (2020, March 9). Cisco. Retrieved March 11, 2022, from https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

Cohen, F. (1985). *Computer viruses*. University of Southern California.

Cooper, D. R., & Schindler, P. S. (2019). *Business research methods* (13th ed.). McGraw-Hill Education.

Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications.

Creswell, J. W. (2012*). Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research*. Pearson.

Cyberthreat Defense Report 2021 - CyberEdge Group. (n.d.). CyberEdge Group. Retrieved December 20, 2021, from https://cyber-edge.com/cdr/

Deepanshu, M. (2020). State-of-the-art reinforcement learning algorithms. *International Journal of Engineering Research And*, *V8*(12). https://doi.org/10.17577/ijertv8is120332

Deisenroth, M. P., Rasmussen, C. E., & Peters, J. (2013). Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2), 408-423.

Denning, D. E., "*An Intrusion Detection Model*," Proceedings of the Seventh IEEE Symposium on Security and Privacy, May 1986, pages 119–131

Deokar, B., & Hazarnis, A. (2012). Intrusion detection system using log files and reinforcement learning. *International Journal of Computer Applications*, 45(19), 28-35.

Duan, K., Li, X., & Gao, L. (2020). *A Deep Learning Framework for Network Intrusion Detection System Based on the Cloud Computing Environment*. IEEE Access, 8, 6123-6133.

Eesa AS, Orman Z, Brifcani AMA. *A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems*. Expert Syst Appl 2015;42(5):2670–9

Elangovan, N., & Rajendran, R. (2015, April 25). Conceptual Model: A Framework for Institutionalizing the Vigor in Business Research. IBM 2015 Proceedings - SRIT ISBN: 9-38423-409-X. Retrieved from SSRN: https://ssrn.com/abstract=3515944

Endorf, C., Schultz, E., & Mellander, J. (2012). *Intrusion Detection Systems: A Comprehensive Guide*. Jones & Bartlett Learning.

Endorf, C., Schultz, E., & Mellander, J. (2019). *Intrusion Detection and Prevention Systems: Concepts and Techniques*. Auerbach Publications. ISBN: 978-1138485963.

Equifax Inc. (2017, September 7). Equifax announces cybersecurity incident involving consumer information. Equifax. https://investor.equifax.com/news-and-events/news/2017/09-07-2017-213000628

Erder, M. (2020). *Continuous Architecture: Sustainable Architecture in an Agile and Cloud-Centric World*. CRC Press. ISBN: 978-0367331390.

Fraser Health Authority, 2011. *Quantitative Research Methods and Tools.* http://www.fraserhealth.ca/media/2011_11_14_Quantitative-Research-Methods-and-Tools.pdf

Freeze, D. (2020, November 9). *Global Cybercrime Damages Predicted To Reach $6 Trillion Annually By 2021*. Cybercrime Magazine. Retrieved January 20, 2022, from https://cybersecurityventures.com/annual-cybercrime-report-2019-to-2020/

Gao J, Chai S, Zhang B, Xia Y. (2019*) Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis*. Energies;12(7):1223.

Gao, J., Han, W., Li, X., & Liu, X. (2020*). Entropy-Based Machine Learning: An Overview and Opportunities.* IEEE Access, 8, 101542-101557. doi: 10.1109/ACCESS.2020.2991764

García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems, and challenges. *Computers & Security*, 28(1-2), 18-28.

García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2012). Anomaly-based network intrusion detection: Techniques, systems, and challenges. *Computers & Security*, 31(3), 362-371.

Geron, A. (2019). *Hands-on machine learning with scikit-learn, Keras, and tensorflow: Concepts, tools and techniques to build Intelligent Systems*. O'Reilly.

Grounds, M. & Kudenko, D. (2008). *Parallel reinforcement learning with linear function approximation*. In Proceedings of the 5th, 6th, and 7th European Conference on Adaptive and Learning Agents and Multi-agent Systems: Adaptation and Multi-agent Learning, pp. 60–74. Springer-Verlag,

Gogoi, P., Bhuyan, M.H., Bhattacharyya, D.K., & Kalita, J.K. (2012). Packet and flow-based network intrusion dataset. "Contemporary Computing". Springer Berlin Heidelberg, P 322-334.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT Press.

Hadžiosmanovi´c, D., Simionato, L., Bolzoni, D., Zambon, E., and Etalle, S. (2012*) N-gram against the machine: On the feasibility of the n-gram network analysis for binary protocols*. In International Workshop on Recent Advances in Intrusion Detection, pages 354–373. Springer.

Hassan, M., Rehman, S. U., & Shah, M. A. (2018). *Intrusion Detection Systems (IDS): A comprehensive study*. In 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. 1-6). IEEE.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction. Springer Science & Business Media.

Hazem, M.E, & Nikos, M., "*Real-Time Intrusion Detection Algorithm for Network Security"*, WSEAS Transactions on Communications, Issue 12, Volume 7, December 2008.

Hosseini Bamakan, S. M., Wang, H., & Shi, Y. (2017). Ramp loss K-support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem. *Knowledge-Based Systems*, *126*, 113–126. https://doi.org/10.1016/j.knosys.2017.03.012

Hsu, Y. F., & Matsuoka, M. (2020, November 9). A Deep Reinforcement Learning Approach for Anomaly Network Intrusion Detection System. *2020 IEEE 9th International Conference on Cloud Networking (CloudNet)*. https://doi.org/10.1109/cloudnet51028.2020.9335796

H., Ren, J., Guo, J., Qian, W., Yuan, H., Hao, X., & Jingjing, H. (2019, June 16). *Building an Effective Intrusion Detection System by Using Hybrid Data Optimization Based on Machine Learning Algorithms*. Building an Effective Intrusion Detection System by Using Hybrid Data Optimization Based on Machine Learning Algorithms. Retrieved May 20, 2021, from https://www.hindawi.com/journals/scn/2019/7130868/

https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781838550356/1/ch01lvl1sec05/ml-development-life-cycle. (n.d.). Retrieved September 30, 2021, from https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781838550356/1/ch01lvl1sec05/ml-development-life-cycle.

Hu, J., Yu, X., Qiu, D., & Chen, H. H. (2009). A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection. *IEEE Network*, *23*(1), 42–47. https://doi.org/10.1109/mnet.2009.4804323

Jiadong, R.; Xinqian, L.; Qian, W.; Haitao, H.; Xiaolin, Z. *A multi-level intrusion detection method based on KNN outlier detection and random forests*. J. Comput. Res. Dev. 2019,56, 566

Jian, X., Jing, Y. & Fengyu Liu, "*A fuzzy rules-based approach for performance anomaly detection*" IEEE 2005

Jiang, K., Wang, W., Wang, A., & Wu, H. (2020). Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network. *IEEE Access*, *8*, 32464–32476. https://doi.org/10.1109/access.2020.2973730

Jiong, Z., & Mohammad, Z. "*Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection*" IEEE International Conference on Communications, 2006

Karagiannakos, S. (2019, May 3). *The idea behind actor-critics and how A2c and A3C improve them*. Medium. https://towardsdatascience.com/the-idea-behind-actor-critics-and-how-a2c-and-a3c-improve-them-6dd7dfd0acb8

Karami, A., & Guerrero-Zapata, M. (2015, February). A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks. *Neurocomputing*, *149*, 1253–1269. https://doi.org/10.1016/j.neucom.2014.08.070

Kemper, E.A., Stringfield, S., Teddlie, C. Mixed methods sampling strategies in social science research. In: Tashakkori, A., Teddlie, C., editors. *Handbook of mixed methods in the social and behavioral sciences*. Sage; Thousand Oaks, CA: 2003. pp. 273–296.

Khalid, M. S., Arshad, H., & Siddique, K. (2018). Network Intrusion Detection Systems: A Comprehensive Review. *Journal of Network and Computer Applications*, 116, 1-27.

Khammassi, C., & Krichen, S. (2017, September). A GA-LR wrapper approach for feature selection in network intrusion detection. *Computers & Security*, *70*, 255–277. https://doi.org/10.1016/j.cose.2017.06.005

Khan, N.M., Negi, A., Thaseen, I.S., et al. “Analysis on improving the performance of machine learning models using feature selection technique.” In*: International conference on intelligent systems design and applications. Springer*; 2018. pp. 69–77.

Kim, G., Lee, S., & Kim, S. (2014, March). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems With Applications*, *41*(4), 1690–1700. https://doi.org/10.1016/j.eswa.2013.08.066

Kombo, D. K. & Tromp, D. L.A. (2006). *Proposal and Thesis Writing: An introduction*. Nairobi: Paulines Publications Africa.

Kononenko, I., & Kukar, M. (2007). *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Horwood Publishing.

Kumar, V. (2014, June 30). Feature Selection: A literature Review. *The Smart Computing Review*, *4*(3). https://doi.org/10.6029/smartcr.2014.03.007

Kumar, V., Sinha, D., Das, A. K., Pandey, S. C., & Goswami, R. T. (2019, October 29). An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset. *Cluster Computing*, *23*(2), 1397–1418. https://doi.org/10.1007/s10586-019-03008-x

Lunt, T. F., "Detecting Intruders in Computer Systems,*" 1993 Conference on Auditing and Computer Technology*, SRI International

Marriott International. (2018, November 30). Marriott International announces breach of Starwood guest reservation database. Marriott News Center. https://marriott.gcs-web.com/news-releases/news-release-details/marriott-international-announces-breach-starwood-guest

McHugh, J. (2000, November). Testing Intrusion detection systems. *ACM Transactions on Information and System Security*, *3*(4), 262–294. https://doi.org/10.1145/382912.382923

Miao, W., Cheng, Z. and Jingjing (2006), “Native API Based Windows Anomaly Intrusion Detection Method Using SVM*” IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*,

Mitchell, T. (1997). *Machine Learning.* New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., and Kavukcuoglu, K. (2016, June). *Asynchronous methods for deep reinforcement learning*. In International Conference on Machine Learning (pp. 1928-1937).

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529-533.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine* Learning. MIT Press.

Moustafa, N., & Slay, J. (2016, January 11). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective*, *25*(1–3), 18–31. https://doi.org/10.1080/19393555.2015.1125974

Moustafa, N. & Slay, J. et al. (2015). "*UNSW-NB15: a comprehensive data set for network intrusion detection systems* (UNSW-NB15). Retrieved from https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/

Morse, J.M. & Niehaus, L. *Mixed method design: Principles and procedures*. Left Coast Press; Walnut Creek, CA: 2009.

Muda, Z., Yassin, W., Sulaiman, M.N. and Udzir, N.I." *Intrusion Detection based on K-Means Clustering and Naïve Bayes Classification*", 7th International Conference on IT in Asia (CITA), 2011.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Nakkeeran, R., Aruldoss Albert, T., & Ezumalai, R. (2010). Agent Based Efficient Anomaly Intrusion Detection System in Adhoc networks. *International Journal of Engineering and Technology*, *2*(1), 52–56. https://doi.org/10.7763/ijet.2010.v2.99

N., D. E., & R., R. (2020, January 31). *Conceptual Model: A Framework for Institutionalizing the Vigor in Business Research*. Conceptual Model: A Framework for Institutionalizing the Vigor in Business Research by Dr. Elangovan N., Rajendran R. :: SSRN. https://ssrn.com/abstract=3515944

Niyaz, Q., Sun, W., Javaid, A. and Alam, A. "A Deep Learning Approach for Network Intrusion Detection System," *in Proceedings of the 9th EAI International Conference on Bio Inspired Information and Communication Technologies* (BICT), 2015

O'Leary, Z. (2004). *The essential guide to doing research*. Sage.

Ong'ondo C.O. (2009). *Pedagogical practice and support of English language student teachers during the practicum in Kenya* (PhD Thesis). University of Leeds, UK.

Osanaiye, O., Cai, H., Choo, K. K. R., Dehghantanha, A., Xu, Z., & Dlodlo, M. (2016, May 10). Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. *EURASIP Journal on Wireless Communications and Networking*, *2016*(1). https://doi.org/10.1186/s13638-016-0623-3

Penchikala, S. (2016, May 15). *Big Data Processing with apache spark - part 4: Spark machine learning*. InfoQ. https://www.infoq.com/articles/apache-spark-machine-learning

Puterman, M. L. (2014). *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons.

Quah, K. H., Quek, C. & Leedham, G (2002). "*Pattern classification using fuzzy adaptive learning control network and reinforcement learning*," Proceedings of the 9th International Conference on Neural Information Processing. ICONIP '02., Singapore pp. 1439-1443 vol.3. 2002.

Ravitch, S. M. (2016, January 20). Reason & Rigor. In *How Conceptual Frameworks Guide Research*.

Remenyi, D., Williams, B., Money, A., & Swartz, E. (1998, August 1). Doing Research in Business and Management. In *An Introduction to Process and Method*. https://doi.org/10.1604/9780761959496

Saunders, M., Lewis, P., & Thornhill, A. (2019). *Research methods for business students* (8th ed.). Pearson.

Servin, A., & Kudenko, D. (n.d.). Multi-agent Reinforcement Learning for Intrusion Detection. *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, 211–223. https://doi.org/10.1007/978-3-540-77949-0_15

Sewak, M. (2019). Actor-Critic Models and the A3C. *Deep Reinforcement Learning*, 141–152. https://doi.org/10.1007/978-981-13-8285-7_11

Shabtai, A., Elovici, Y., Rokach, L (2012). *A survey of data leakage detection and prevention solutions*. Berlin: Springer;

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). *Deterministic policy gradient algorithms.* In Proceedings of the 31st International Conference on Machine Learning (Vol. 32, pp. 387-395). PMLR.

Sotiris, V. A., Tse, P. W., & Pecht, M. G. (2010, June). Anomaly Detection Through a Bayesian Support Vector Machine. *IEEE Transactions on Reliability*, *59*(2), 277–286. https://doi.org/10.1109/tr.2010.2048740

Sperotto, A., Baier, H., Pras, A., & Stiller, B. (2010). *Evaluation of netflow-based intrusion detection for operational networks*. Journal of Network and Systems Management, 18(4), 452-478.

Sutton, R. S., & Barto, A. G. *Reinforcement learning: An introduction*. MIT press Boston, MA, USA, 2018.

Swanson, R. A. (2013*). Theory building in applied disciplines.* San Francisco, CA: Berrett-Koehle

Szepesvári, C. (2010). Algorithms for Reinforcement Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 4(1), 1-103.

Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A. (2009). "*A detailed analysis of the KDD CUP 99 data set*". Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications.

Taylor, M. E., Stone, P., & Liu, Y. (2016). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research,* 17(1), 1-40.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11), 1134-1142.

Viegas, E., Santin, A. O., Fran?a, A., Jasinski, R., Pedroni, V. A. and Oliveira, L. S. (2017-01-01). "Towards an Energy-Efficient Anomaly-based Intrusion Detection Engine for Embedded Systems*". IEEE Transactions on Computers.* 66 (1): 163–177. doi:10.1109/TC.2016.2560839. ISSN 0018-9340.

Vikash, C., Singh, S. K., Khamparia, A., Gupta, D., Tiwari, P., Moreira, C., & et al. (2020). *A Novel Transfer Learning Based Approach for Pneumonia Detection in Chest X-ray Images.* Applied Sciences, 10, 559.

Walliman, N. S. & Walliman N. (2011) "*Research methods: the basics*" Taylor and Francis

Wei, W., Xiaohong, G., Xiangliang, Z., "Profiling program and user behaviors for anomaly intrusion detection based on non-negative matrix factorization" *43rd IEEE Conference on Decision and Control*, 2004. Issue Date: 14-17 Dec. 2004, On page(s): 99 - 104 Vol.1

Williams, C. (2011, February 7). Research Methods. *Journal of Business & Economics Research (JBER)*, *5*(3). https://doi.org/10.19030/jber.v5i3.2532

Wikipedia (2017, May 14). WannaCry ransomware attack. Retrieved from https://en.wikipedia.org/wiki/WannaCry_ransomware_attack

Wikipedia (2017, June 27). Petya (malware). Retrieved from https://en.wikipedia.org/wiki/Petya_(malware)

Wilson, J. (2010, May 5). Essentials of Business Research. In *A Guide to Doing Your Research Project*.

Wressnegger, C., Schwenk, G., Arp, D., and Rieck, K. (2013). A close look on n-grams in intrusion detection: anomaly detection vs. classification. *In Proceedings of the 63rd 2013 ACM workshop on Artificial intelligence and security*, pages 67–76. ACM, 2013.

Yahoo. (2017, October 3). Yahoo provides notice to additional user accounts affected by previously disclosed August 2013 data theft. Yahoo. https://yahoo.com/security-update

Yang, P., & Zhu, Q. (2011, March). Finding key attribute subset in dataset for outlier detection. *Knowledge-Based Systems*, *24*(2), 269–274. https://doi.org/10.1016/j.knosys.2010.09.003

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). *How transferable are features in deep neural networks? In Advances in neural information processing systems* (pp. 3320-3328).

Zeng, J., Ju, R., Qin, L., Hu, Y., Yin, Q., & Hu, C. (2019). *Navigation in unknown dynamic environments based on Deep Reinforcement Learning*. *Sensors*, *19*(18), 3837. https://doi.org/10.3390/s19183837

Zhang, J., Li, H., Gao, Q., Wang, H., & Luo, Y. (2015, October). Detecting anomalies from big network traffic data using an adaptive detection approach. *Information Sciences*, *318*, 91–110. https://doi.org/10.1016/j.ins.2014.07.044

Zhenghong, X., Chuling, L. and Chaotian, C. (2009), "An Anomaly Detection Scheme Based on Machine Learning for WSN" *IEEE International Conference on Information Science and Engineering.*

Zhou, K., Wang, W., Hu, T., & Deng, K. (2021, February 25). Application of Improved Asynchronous Advantage Actor Critic Reinforcement Learning Model on Anomaly Detection. *Entropy*, *23*(3), 274. https://doi.org/10.3390/e23030274

Zimek, A., & Schubert, E. (2017). Outlier Detection. *Encyclopedia of Database Systems*, 1–5. https://doi.org/10.1007/978-1-4899-7993-3_80719-1

APPENDICES

Appendix I: NACOSTI Research Licence

REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: 192490

Date of Issue: 16/May/2022

RESEARCH LICENSE

This is to Certify that Mr.. JUNIOR YEGO of Moi University, has been licensed to conduct research in Nandi on the topic: A DEEP REINFORCEMENT LEARNING APPROACH TO MODELLING AN INTRUSION DETECTION SYSTEM USING A3C ALGORITHM for the period ending : 16/May/2023.

License No: NACOSTI/P/22/17528

192490

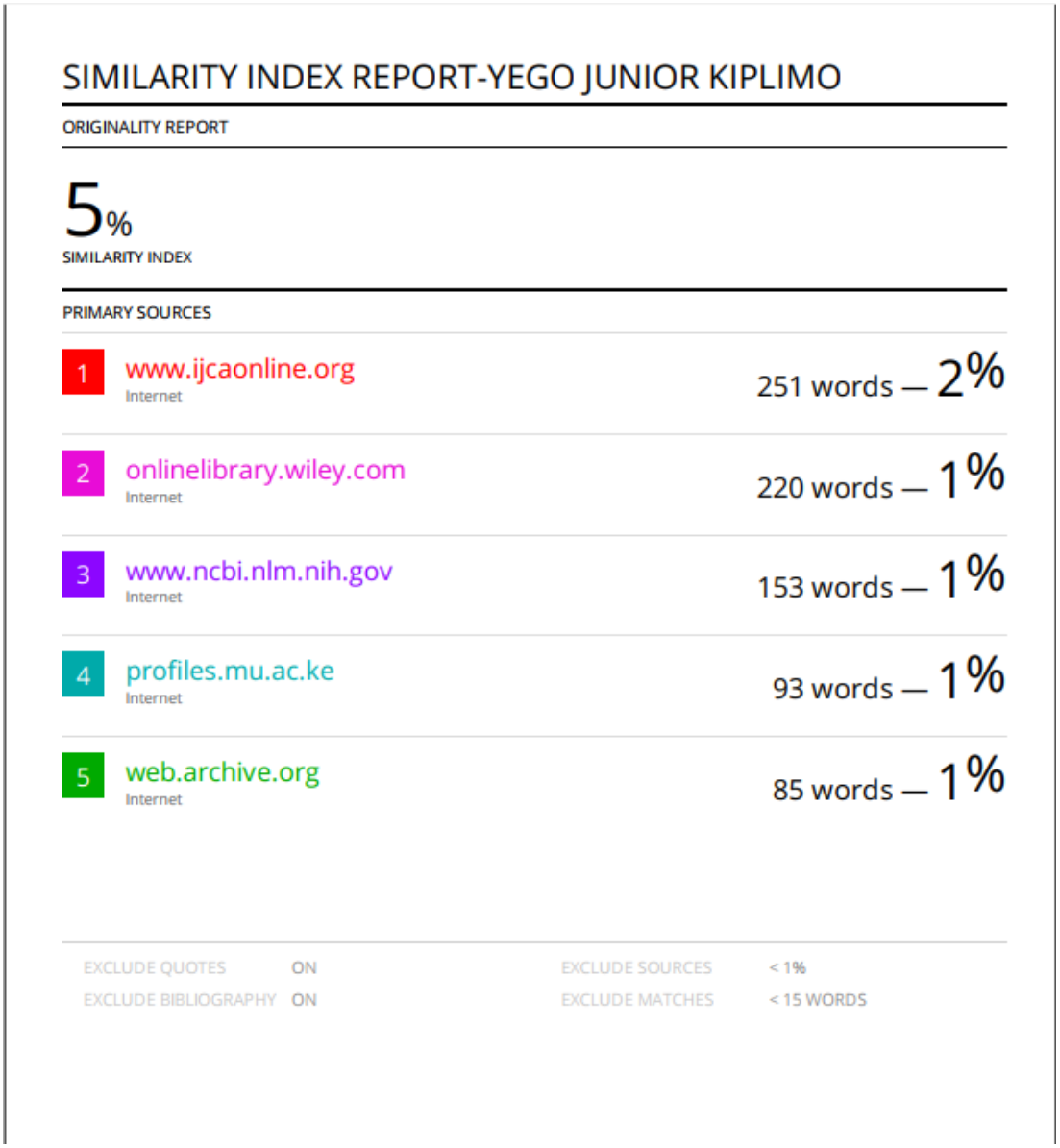
Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

Appendix II: Plagiarism Similarity Index Report



SIMILARITY INDEX REPORT-YEGO JUNIOR KIPLIMO

ORIGINALITY REPORT

5%
SIMILARITY INDEX

PRIMARY SOURCES

1	www.ijcaonline.org Internet	251 words — 2%
2	onlinelibrary.wiley.com Internet	220 words — 1%
3	www.ncbi.nlm.nih.gov Internet	153 words — 1%
4	profiles.mu.ac.ke Internet	93 words — 1%
5	web.archive.org Internet	85 words — 1%

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	< 1%
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	< 15 WORDS