



**NEAR EAST UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
ARTIFICIAL INTELLIGENCE ENGINEERING DEPARTMENT**

**A DEEP ATTENTION FRAMEWORK FOR NETWORK ANOMALY
DETECTION**

M.Sc. THESIS

Mercel VUBANGSI

Nicosia

June, 2023

MERCEL VUBNAGSI

DAF FOR NETWORK ANOMALY DETECTION

**MASTERS
THESIS**

2023

**NEAR EAST UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
DEPARTMENT OF ARTIFICIAL INTELLIGENCE ENGINEERING**

**A DEEP ATTENTION FRAMEWORK FOR NETWORK ANOMALY
DETECTION**

M.Sc. THESIS




Mercel VUBANGSI

**Supervisor
Prof. Dr. Fadi Al-Turjman**

**Nicosia
June, 2023**


Approval

We certify that we have read the thesis submitted by **Mercel VUBANGSI** titled “**A Deep Attention Framework for Network Anomaly Detection**” and that in our combined opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Examining Committee	Name-Surname	Signature
Head of the Committee:	Prof. Dr. Rahib H. Abiyev	
Committee Member*:	Asst. Prof. Pwadubashiyi Coston Pwavodi.....	
Supervisor:	Pr. Dr. Fadi Al-Turjman	

Approved by the Head of the Department

...../...../2023


 Prof. Dr. Fadi Al-Turjman
 Head of Department

Approved by the Institute of Graduate Studies


 Prof. Dr. Kemal Hüsnü Can Başer
 Head of the Institute



Declaration

I hereby declare that all information, documents, analysis and results in this thesis have been collected and presented according to the academic rules and ethical guidelines of Institute of Graduate Studies, Near East University. I also declare that as required by these rules and conduct, I have fully cited and referenced information and data that are not original to this study.

Mercel VUBANGSI

01/06/2023

Acknowledgments

I am filled with immense gratitude and want to express it from the bottom of my heart to my supervisor Prof. Dr. Fadi Al-Turjman, Chair of the AI department and Associate Dean for Research in the Faculty of engineering at Near East university. He has been a constant pillar of support, providing unwavering guidance and encouragement throughout my research work. His invaluable insights and computational resources have made my research work possible.

I would also like to extend my heartfelt gratitude to Dr. Auwalu Saleh Mubarak and Dr. Zubaida Said Ameen for their exceptional teaching, seasoned courses, and beyond-coursework discussions that inspired me and prepared me for this research work.

I cannot thank enough the Administrative Secretary of the AI department, Mrs Sinem Al-Turjman for her guidance and support in choosing the optimum courses for my coursework.

I am forever indebted to my loving wife and children, whose constant love, support, and motivation have been the driving force behind my academic achievements.

I would also like to acknowledge the unwavering support and assistance provided by my colleagues and friends, who have made my journey enjoyable and worthwhile.

Lastly, I would like to express my deep appreciation to Near East University for providing me with the scholarship opportunity to pursue my postgraduate studies and for the immense resources made available to me to achieve my academic goals.

Mercel VUBANGSI

Abstract

“A Deep Attention Framework for Network Anomaly Detection”

Mercel VUBANGSI

MSc, Department of Artificial Intelligence Engineering

June 2023, 90 pages

This dissertation introduces an innovative transformer architecture specifically designed for deep learning-based multi-class classification tasks. The motivation behind this research stems from the remarkable success of transformer models in natural language processing tasks. Leveraging the transformer architecture's strong performance in language-related tasks, it is intended to apply the proposed framework to network anomaly detection using a publicly available dataset.

To adapt the transformer architecture for multi-class classification tasks, a Norm/Re-weight technique is introduced. This technique optimizes the space and time complexity in classification computation, ensuring efficient and accurate predictions. The framework is implemented and trained using a dataset published by the Intelligent Security Group of Canberra Australia, and its performance is thoroughly assessed.

The results of this study demonstrate the efficacy of the proposed transformer architecture for detecting anomalous activities in network traffic data. The framework surpasses state-of-the-art methods on a comprehensive set of 15 metrics, showcasing its superior performance. By achieving commendable results, the model proves its ability to accurately classify network anomalies, which is crucial for effective anomaly detection in various applications.

Significantly, this research also contributes to the AI research community by developing a user-friendly Python package that encapsulates the proposed model. This package has been made freely available on the Python Package Index (PyPi), providing researchers with a valuable tool to advance their own work in the field. The introduction of the adapted transformer architecture and the Norm/Re-weight technique holds promise for enhancing the performance of multi-class classification tasks beyond the detection of abnormal instances in network traffic.

Key Words: transformer architecture, deep learning, multi-class classification,

network Anomaly Detection, machine learning, Python package, norm/re-weight,

ÖZET

“Ağ Anomalisi Tespiti için Derin Dikkat Çerçevesi”

Mercel VUBANGSI

Yüksek Lisans, Yapay Zeka Mühendisliği Bölümü

June 2023, 90 sayfa

Bu tez, derin öğrenmeye dayalı çok sınıflı sınıflandırma görevleri için özel olarak tasarlanmış yenilikçi bir transformatör mimarisini tanıtmaktadır. Bu araştırmanın arkasındaki motivasyon, trafo modellerinin doğal dil işleme görevlerindeki olağanüstü başarısından kaynaklanmaktadır. Transformatör mimarisinin dille ilgili görevlerdeki güçlü performansından yararlanan bu çalışmanın amacı, önerilen çerçeveyi halka açık bir veri kümesi kullanarak ağ anomalisi tespitine uygulamaktır.

Transformatör mimarisini çok sınıflı sınıflandırma görevlerine uyarlamak için bir Norm/Yeniden ağırlıklandırma tekniği tanıtıldı. Bu teknik, sınıflandırma hesaplamasındaki alan ve zaman karmaşıklığını optimize ederek verimli ve doğru tahminler sağlar. Çerçeve, ağ anormalliği veri kümesi kullanılarak uygulanır ve eğitilir ve performansı kapsamlı bir şekilde değerlendirilir.

Bu çalışmanın sonuçları, önerilen trafo mimarisinin ağ anomali tespiti için etkinliğini göstermektedir. Çerçeve, üstün performansını sergileyen kapsamlı bir 15 ölçüm kümesinde en son teknolojiye sahip yöntemleri geride bırakıyor. Model, övgüye değer sonuçlar elde ederek, çeşitli uygulamalarda etkin anomali tespiti için çok önemli olan ağ anormalliklerini doğru bir şekilde sınıflandırma yeteneğini kanıtlıyor. Bu araştırma, önerilen modeli kapsayan kullanıcı dostu bir Python paketi geliştirerek AI araştırma topluluğuna da önemli ölçüde katkıda bulunur. Bu paket, Python Paket Dizininde (PyPi) ücretsiz olarak kullanıma sunuldu ve araştırmacılara bu alanda kendi çalışmalarını ilerletmeleri için değerli bir araç sağladı. Uyarlanmış trafo mimarisinin ve Norm/Yeniden ağırlıklandırma tekniğinin tanıtılması, ağ anormallik tespitinin ötesinde çok sınıflı sınıflandırma görevlerinin performansını artırma konusunda umut vaat etmektedir.

Anahtar Kelimeler: trafo mimarisi, derin öğrenme, çok sınıflı sınıflandırma, ağ Anomali Tespiti, makine öğrenimi, Python paketi, norm/yeniden ağırlıklandırma,

Table of Contents

Approval.....	i
Declaration	ii
Acknowledgments.....	iii
Abstract	iv
ÖZET.....	v
Table of Contents	vii
List of Appendices	x
List of Tables.....	xi
List of Figures	xii
List of Abbreviations.....	xiv

CHAPTER I

Introduction.....	1
1.1 The need for Cyber Anomaly Detection	1
1.2 Background and Motivation.....	2
1.2.1 Evolution of Cyber Threats.....	2
1.2.2 Limitations of Traditional Anomaly Detection Techniques	3
1.2.3 Advancements in Deep Learning for Anomaly Detection.....	3
1.2.4 Potential of Deep Attention Methods	3
1.2.5 Role of Attention Mechanisms in Anomaly Detection.....	4
1.3 Gap in Existing Literature	4
1.4 Importance of the Study	5
1.5 Problem Statement	5
1.6 Objectives of the Study	6
1.7 Scope of the study	7
1.8 Limitations of the Study	8
1.9 Contribution of the Study	8

CHAPTER II

Review of related works.....	11
2.1 Sources of Network Anomalies.....	11
2.1.1 Assessing Network traffic data for anomalous activities	11

2.1.2 Importance of Network Anomaly Detection	13
2.1.3 Challenges in Network Anomaly Detection	13
2.1.4 Future Directions in Network Anomaly Detection.....	13
2.2 Current Knowledge	14
2.3 Leveraging Transformer Attention Mechanisms for Enhanced Network Anomaly Detection.....	19

CHAPTER III

Methodology	21
3.1 Research Design and Approach.....	21
3.2 Data Acquisition and Pre-processing	21
3.3 Feature Importance	22
3.4 Data Classes	25
3.5 Introducing a Norm/Re-weight (NR) technique to speed up Calculations.....	26
3.6 Building the Architectures.....	27
3.6.1 Coding the Architecture:.....	29
3.6.2 LISTING 1: Multihead attention unit.	29
3.6.3 The LSTM Units	32
3.6.4 LISTING 2: Python code of the LSTM Unit.....	34
3.7 Hyper parameter search.....	35
3.8 Evaluation Metrics	37

CHAPTER IV

Results and Discussion.....	39
4.1 Comparative Analysis of Deep Learning Models: Deep Attention, LSTM, CNN, RNN, and GRU	39
4.2 Results	39
4.2.1 DA results	39
4.2.2 LSTM results	41
4.2.3 CNN results.....	43
4.3 Discussion of results.....	53
4.4 Major Achievements	55

CHAPTER V

Conclusion and Recommendations	57
5.1 Conclusion.....	57
5.2 Limitations.....	58
5.3 Recommendations	60
REFERENCES.....	62
APPENDICES	71
Appendix A: Deep Attention Model as a python package named deepAttention.....	71
Appendix B: List of papers from the Thesis	73
Appendix C: Turnitin Similarity Report	74
Appendix D: CV.....	75

List of Appendices

Appendix A: Demonstration of basic Usage of the Deep Attention framework

Appendix B: List of papers from Thesis

Appendix C: Turnitin Similarity Report

Appendix D: CV

List of Tables

Table 1: Summary of applications of transformers in intrusion detection systems	17
Table 2: UNSW-NB15 dataset for network intrusion detection research, with 2.5 million records and 10 attack categories, collected between 2015-2017 and available for download.	22
Table 3: Summary table of optimized parameters for the Deep Attention model	36
Table 4: Summary table on performance evaluation of the 5 models on 15 metrics. Each entry in the table is a two-element array with the first value representing the validation score and the second value representing the test score.	52

List of Figures

Figure 1: Bar plot of feature importance of the dataset for the top 45 features. service and sbyte stand out as the most important features.....	24
Figure 2: Distribution of classes in a stratified random sample of size 80000 from the dataset. This figure shows that the network has majorly normal activity and the generic attack is the least prevalent.....	25
Figure 3: Binary distribution of network activity in a stratified random sample of size 80000 from the dataset. A little imbalance is observed between normal and anomalous activity	25
Figure 4: Illustration of the data flow process in the Deep Attention Framework.	26
Figure 5: The Deep Attention Model Architecture. Here, the LSTM units provide the deep neural network. This architecture lacks the positional embedding component. The inputs are 3D tensors and outputs are class probabilities.	28
Figure 6: Illustration of the working scheme of Multihead Attention based on Key, Query, Value arithmetics	29
Figure 7: Diagrammatic representation of the LSTM unit.....	33
Figure 8: Graphic display of experimental results from hyper parameter search with the aid of KerasClassifier from the keras library. The dashed line shows the position of the best score for each parameter.....	36
Figure 9: Results of the training process of the Deep Attention model. The model rapidly learns the characteristics of the data within the first 10 epochs as shown by the sharp descent of the loss curve. The panel of metrics illustrates the excellent performance of the model with near zero MSE&MAE.	40
Figure 10: Confusion matrix of performance of the Deep Attention model. A near perfect score is achieved with a single value off the major diagonal.....	41
Figure 11: Results of the training process of the LSTM model. The model rapidly learns the characteristics of the data within the first 10 epochs as shown by the sharp descent. The panel of metrics shows significant errors (MAE&MSE).....	42
Figure 12: Confusion matrix of LSTM Model. The significant number of off-diagonal non-zero entries indicate False Positives and False negatives.	43
Figure 13:Plot of history of losses against epoch number for the CNN model shows relatively good performance but not as good as the DA model. The errors MAE and MSE are higher than in the DA model.....	44

Figure 14: Confusion Matrix for the CNN model. A misclassification is apparent with the identification of a non-existent class.....	45
Figure 15: RNN training and testing errors indicate the worst performance amongst all the deep learning models used in the experiment.	46
Figure 16: A plot of the confusion matrix for the RNN shows that the model identifies a non-existent class, leading to a large MAE and MSE.....	47
Figure 17: A comparison of the evolution of losses for all 5 models in the experiment. The recurrent neural networks (RNN and GRU) show marked instability with the this dataset.	48
Figure 18: Bar plot of training and test accuracies for all five models showing the DA model as the best and the RNN as the worst.	48
Figure 19: bar plot of training times for all 5 models. The DA Model takes the longest to train while the RNN model takes the shortest time in a period of 50 epochs.....	49
Figure 20: Bar plot of the prediction times for the validation set and test sets. The DA model is the slowest on both sets. This is because on our computational resource, parallel computing is not possible, which would have harnessed the power of parallelization offered by multihead attention.	50
Figure 21: Bar plots of the mean squared error on validation and test sets for all 5 models. DA and CNN show the best scores while RNN and GRU show the worst scores.....	51
Figure 22: Bar plots of the mean absolute error on validation and test sets for all 5 models. DA and CNN show the best scores while RNN and GRU show the worst scores.....	52

List of Abbreviations

Acronym	Meaning
DA	Deep Attention
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
CNN	Convolutional Neural network
GRU	Gated Recurrent Units
MLP	Multilayer Perceptron
DL	Deep Learning
ML	Machine Learning
NR	Norm/Re-weight

CHAPTER I

Introduction

In this chapter, we shall explore the issue of identifying anomalies in network traffic and the significance of formulating effective techniques to tackle this problem. Initially, we shall delve into the background of the study and elaborate on the increasing demand for dependable and efficient intrusion detection systems in the modern digital era. Subsequently, we shall discuss the shortcomings of current approaches to detecting network anomalies, which necessitates the creation of a novel framework that can surmount these obstacles. Lastly, we shall wrap up by presenting the research objectives and the anticipated contributions of the proposed Deep Attention framework to the field of network anomaly detection.

1.1 The need for Cyber Anomaly Detection

Modern cybersecurity, which is becoming more and more crucial in the digital era, must include the detection of cyber anomalies. It is now more crucial than ever to identify and stop unwanted access to computer systems and networks because as technology develops, the potential of cyberattacks and data breaches increases.

The first reason for the need for cyber Anomaly Detection is the proliferation of cyber threats. Today, cyber-attacks come in various forms, such as viruses, malware, phishing, and ransomware, among others. These threats are becoming more sophisticated, making them difficult to detect and defend against using traditional security measures. In order to reduce damage and prevent data loss, there is an increasing need for Anomaly Detection systems that can identify and react to these threats in real-time.

Secondly, the complexity of modern computer networks and systems makes it difficult to monitor and control access. Large organizations and government agencies have complex networks that are spread across multiple locations, making it challenging to keep track of all the devices and users accessing the system. Anomaly Detection systems can help monitor the network, identify any suspicious behavior or access patterns, and alert the security team in real-time to prevent any unauthorized access.

Thirdly, the need for Anomaly Detection systems is further highlighted by the rise of remote work and cloud computing. Remote workers and cloud-based systems have opened up new avenues for cyber attackers to exploit vulnerabilities in the

network. Anomaly Detection systems can help monitor remote access to the network and identify any unusual behavior or unauthorized access.

Another reason why cyber Anomaly Detection is essential is the high cost of cyber-attacks. Cyberattacks can cause significant damage to an organization's reputation, lead to financial losses, and even result in the loss of intellectual property. A robust Anomaly Detection system can help minimize the impact of a cyber-attack by quickly detecting and responding to the threat, reducing the amount of damage caused.

Finally, the increasing regulatory pressure on organizations to protect their data emphasizes the need for Anomaly Detection systems. Governments and regulatory bodies have implemented data protection regulations, such as GDPR, HIPAA, and PCI-DSS, which require organizations to take measures to protect their data from cyber threats. Failure to comply with these regulations can lead to significant fines and legal action. An Anomaly Detection system can help organizations meet their regulatory requirements and safeguard their data from cyber-attacks.

1.2 Background and Motivation.

1.2.1 Evolution of Cyber Threats

The evolution of cyber threats, as reported by Singh & Khare, (2022) in their survey paper, has been one of the most significant drivers of the development of cyber Anomaly Detection techniques. In the early days of computer networks, cyber threats were relatively simple, such as viruses and worms that could be detected using antivirus software. However, as computer networks became more complex and interconnected the nature of cyber threats evolved, with the emergence of new types of threats, such as malware, botnets, and advanced persistent threats (APTs).

APTs are one of the most dangerous and difficult to detect types of cyber threats (Ahmed Issa & Albayrak, 2021), characterized by their stealthy nature and advanced capabilities. APTs are designed to remain undetected for extended periods, allowing attackers to conduct reconnaissance, collect sensitive data, and establish a foothold in the target network. Traditional Anomaly Detection techniques are often ineffective against APTs, as they are designed to detect known patterns of attacks and do not take into account the sophisticated tactics used by APTs.

1.2.2 Limitations of Traditional Anomaly Detection Techniques

Traditional Anomaly Detection techniques have several limitations that make them ineffective against advanced cyber threats. One of the main limitations of traditional Anomaly Detection techniques is their reliance on known patterns of attacks (Zarpelão et al., 2017). Systems for detecting anomalies, such as viruses, worms, and other forms of malware, are created to identify known attack patterns. However, this strategy is unsuccessful against APTs since they employ complex strategies that are difficult to identify using conventional Anomaly Detection tools.

Another limitation of traditional Anomaly Detection techniques is their high false positive rates. Traditional Anomaly Detection systems are often too sensitive, leading to a high number of false positives, which can be time-consuming to investigate and may divert resources from other critical tasks. False positives can also lead to alert fatigue, where security teams become desensitized to alerts, increasing the risk of missing genuine threats.

1.2.3 Advancements in Deep Learning for Anomaly Detection

Deep learning has shown promise as a method of developing intelligent systems and has a number of advantages over more conventional methods. Deep learning is a branch of machine learning that employs neural networks to recognize and learn from data patterns. By discovering patterns that conventional techniques are unable to detect, deep learning models are very successful at detecting complex and developing cyber threats, such as APTs.

Convolutional neural networks (CNNs) and recurrent neural networks (RNNs), two deep learning approaches, have demonstrated potential in Anomaly Detection by reaching high accuracy rates and low false positive rates. While RNNs are good at spotting patterns in sequential data, like network traffic, CNNs are better at spotting patterns in spatial data, like images and videos.

1.2.4 Potential of Deep Attention Methods

While there have been promising developments in the application of deep learning to Anomaly Detection, opportunities for further advancement still exist. One such opportunity is found in the utilization of deep neural networks with simple or multi-head attention, which amalgamate deep learning techniques and the

Transformer Attention mechanism with the aim of enhancing the accuracy and efficiency of Anomaly Detection systems.

1.2.5 Role of Attention Mechanisms in Anomaly Detection

The transformer technology incorporates the attention mechanism to enable the model to focus on specific parts of the input sequence when processing it. The attention mechanism captures the dependencies and relationships between different elements within a sequence by assigning different weights to different parts of the input sequence during the computation. This enables the model to extract meaningful information and make more accurate predictions.

The transformer model employs self-attention or multi-head attention to implement the attention mechanism. Self-attention captures the dependencies within the sequence by allowing each position in the input sequence to attend to all other positions. Multi-head attention performs multiple independent attention operations in parallel, allowing the model to attend to different parts of the sequence simultaneously and capture different types of dependencies.

The attention mechanism in transformers has demonstrated high effectiveness in various natural language processing tasks such as machine translation, language understanding, and text generation. It enables the model to capture long-range dependencies and handle input sequences of variable lengths more efficiently compared to traditional recurrent neural networks. In the context of network traffic data, attention mechanisms hold the potential to identify key features, including source and destination IP addresses, protocol usage, and port numbers. By detecting such features, attention mechanisms can significantly contribute to the ability of Anomaly Detection systems to differentiate between legitimate and malicious traffic.

1.3 Gap in Existing Literature

Despite the burgeoning interest in deep learning with respect to Anomaly Detection, extant literature reveals a conspicuous void pertaining to the efficacy of Deep Attention methods and attention mechanisms. Although numerous studies have substantiated the effectiveness of deep learning towards Anomaly Detection, scant focus has been directed towards redesigning the transformer blueprint.

1.4 Importance of the Study

The proposed study on cyber Anomaly Detection using Deep Attention methods and attention mechanisms is important for several reasons. Firstly, the study addresses a significant gap in the existing literature, by investigating the potential of hybrid models and attention mechanisms for Anomaly Detection. Secondly, the study has practical implications for the development of effective Anomaly Detection systems that can detect and respond to sophisticated cyber threats. Finally, the study has implications for the broader field of deep learning, by demonstrating the potential of hybrid models and attention mechanisms for other applications, such as natural language processing and image recognition.

1.5 Problem Statement

The escalating occurrence and intricacy of cyber attacks have engendered a formidable task of identifying and counteracting network security menaces instantaneously (Chen, Li, & Wu, 2021; Hooshmand & Hosahalli, 2022). In recent years, deep learning has emerged as a promising technique for Cyber Anomaly Detection, with several studies showing its potential to detect previously unknown threats with high accuracy (Devlin et al., 2019; Abu Al-Haija, 2021). However, The most recent deep learning technology; the transformer attention mechanism, has received very little interest from researchers as regards its application in the field of Cyber Anomaly Detection. This far, studies have focussed on transforming input data to use the original transformer architecture designed for text-based tasks (Dahou et al., 2022), instead of building a domain-specific architecture based on the transformer principle.

Furthermore, prior studies on deep learning for Anomaly Detection have predominantly centered on the utilization of singular models, such as convolutional neural networks (CNNs), long short-term memory (LSTM) architectures, and recurrent neural networks (RNNs), to identify cyber threats. As an illustration, Kim et al. (2016) employed a CNN-based model to detect network anomalies, while Li et al. (2018) employed an LSTM-based model to detect anomalous user conduct. These models are highly data-sensitive, showing varying performances on for instance, the DARPA dataset (Mchugh, 2000), and the KDD Cup 99 dataset (Thomas & Pavithran, 2019), hence their capacity to detect sophisticated attacks that amalgamate multiple attack vectors is limited (Liao & Xie, 2021; Chen, Li, & Wu, 2021).

Moreover, while there have been some studies that have investigated the use of simple and multi-head attention for Anomaly Detection, these studies have focused on using the vector approach for encoding data classes (Devlin et al., 2019), a technique we assert is less efficient due to its high space and time complexity.

The lack of research in this area as noted from comprehensive studies (Pai, Devidas, & Adesh, 2021) highlights the need for the development of more space and time-efficient techniques for network traffic analysis and classification. The objective of this research is to address this gap by proposing a bi-directional Deep Attention transformer (DA) with a more optimized space and time complexities to enhance its real-time detection capabilities for efficient cyber Anomaly Detection tasks.

1.6 Objectives of the Study

The principal aim of this study is to develop a framework that harnesses the power of multihead attention, in combination with multilayer perceptrons, long short-term memory architectures to yield a more sophisticated Deep Attention model, whose performance will be compared with other deep learning methods like: Gated Recurrent Units, Convolutional Neural Networks, and Recurrent Neural Networks, in cyber anomaly detection tasks. Traditional Anomaly Detection techniques are limited in their ability to detect sophisticated attacks that combine multiple attack vectors, making it difficult to provide effective cybersecurity. Therefore, the proposed research aims to explore the potential of the latest technology in Natural Language Processing; the transformer attention; to improve Anomaly Detection systems.

The first step will be to develop a bi-directional Deep Attention transformer (DA) model, which will be used for effective cyber anomaly identification. This model's objective is to efficiently spot aberrant activity in any given network. In order to give a more precise and dependable detection system, the bi-directional attention transformer model will be built to apply a thorough approach that will take into consideration numerous elements like context and patterns. Users will be able to apply this to avert potentially hazardous circumstances before they have a chance to do any significant harm.

The performance of the DA model in comparison to its component parts and other deep learning techniques is another goal of this research. In order to evaluate the effectiveness of the proposed model, it will be compared to other hybrid models

that have been published in the literature as well as individual models such as multilayer perceptrons, LSTMs, gated recurrent units, CNNs, and RNNs. This assessment will show how the proposed hybrid model has the potential to increase the precision of Anomaly Detection systems.

The proposed research will also look into the possibility of the suggested DA paradigm to identify previously unidentified cyberthreats. It is crucial to create Anomaly Detection systems that can identify new risks as they appear since cyber threats are always evolving and becoming more sophisticated. Therefore, the goal of this research is to examine the suggested model's capacity to identify previously unidentified hazards and show how it might be applied to increase the resilience of Anomaly Detection systems.

1.7 Scope of the study

The goal of the present research is to create and implement a deep attention framework for cyber anomaly detection, which focuses on the utilization of attention mechanisms within the transformer architecture to enhance the precision and efficacy of anomaly detection in cyber security applications. The research will include the following key components:

Framework Design: The research will entail the creation of a deep attention framework that is specifically tailored for cyber anomaly detection. This will involve developing an architecture that incorporates self-attention and multi-head attention mechanisms, while also considering the distinct attributes of cyber security datasets.

Model Training and Optimization: The framework will be trained on a range of diverse and representative datasets, with particular emphasis on the widely used UNSW-NB15 dataset (Moustafa & Slay, 2016; Roy & Singh, 2021) for network intrusion detection. The research will explore techniques for optimizing the performance of the framework, such as hyperparameter tuning, regularization, and optimization algorithms.

Comparative Analysis: The developed attention framework will be compared to other popular deep learning models that are frequently used for anomaly detection, including convolutional neural networks (CNNs) (Wang et al., 2020), recurrent neural networks (RNNs), long short-term memory (LSTM) (Kanna & Santhi, 2022), and gated recurrent units (GRUs). A thorough comparative analysis will be

conducted to assess the superiority of the deep attention framework in terms of accuracy, robustness, and efficiency.

Evaluation Metrics: The research will establish appropriate evaluation metrics to accurately gauge the performance of the proposed framework. Metrics such as precision, recall, F1-score, and confusion matrices will be utilized to assess the detection capabilities of the framework.

1.8 Limitations of the Study

As with any research, there are limitations to this study that need to be acknowledged. These limitations include data availability, model selection, and evaluation metrics.

One limitation of this study is the availability of datasets. The study will use publicly available dataset; UNSW-NB15, which is commonly used in the literature for evaluating Anomaly Detection systems. However, these datasets may not capture all types of cyber attacks, and they may not be representative of real-world networks. This may limit the generalizability of the study's findings.

The study's evaluation metrics are still another drawback. The study will assess the effectiveness of the suggested models using common measures including accuracy, precision, recall, and F1 score. The intricacy and subtlety of Anomaly Detection systems may not be fully captured by these metrics, which nevertheless offer a quantitative evaluation of model performance. Additionally, it could be challenging to assess the effectiveness of various models because different metrics might be more pertinent for various attacks.

Finally, the study's focus is on using deep learning techniques to determine anomalies in computer networks. This means that the results might not apply to other cybersecurity domains or to various kinds of networks, including industrial control systems and IoT (Mishra et al., 2017; Gassais et al., 2020).

1.9 Contribution of the Study

In this research, our primary contributions lie in the development of a novel transformer-based architecture for multi-class classification, employing the powerful TensorFlow framework. Our approach combines the inherent strengths of transformers in capturing long-range dependencies with the versatility of multi-class

classification tasks. By leveraging the attention mechanisms within the transformer architecture and a pre-processing technique that optimizes space and time complexity, our model exhibits superior performance in accurately classifying diverse data samples.

One of our notable contributions is the introduction of a technique we named Norm/Re-weight. This technique optimizes the space and time complexity of the model, allowing for efficient processing and resource utilization. By selectively applying normalization and re-weighting techniques to the input data, we achieve a streamlined and efficient implementation of the transformer-based architecture, without compromising on classification accuracy.

To demonstrate the efficacy of our model, we conducted extensive experiments and evaluations on the widely recognized UNSW-NB15 cyber anomaly dataset. Our research revealed the superior classification performance of our transformer-based architecture compared to other state-of-the-art models. The model showcased remarkable accuracy, precision, recall, and F1-score, establishing its effectiveness in accurately categorizing cyber anomalies across multiple classes.

To facilitate the wider adoption and accessibility of our research findings, we encapsulated our developed model into a user-friendly pip-installable Python software package. This software package incorporates the transformer-based architecture and the Norm/Re-weight optimization technique, providing an easily accessible and deployable solution for the AI research community. To ensure open access to the wider community, we published our software package on <https://pypi.org>, enabling researchers worldwide to utilize and benefit from our contributions at no cost.

To sum up, our research makes significant contributions to the field of multi-class classification in the domain of cyber anomaly detection. We have developed a novel transformer-based architecture that demonstrates superior performance in accurately classifying cyber anomalies. Additionally, the introduction of the Norm/Re-weight technique optimizes the model's efficiency and resource utilization. By encapsulating our model into a user-friendly software package, we have furthered its accessibility and availability to the AI research community. We believe that our contributions will advance the state-of-the-art in cyber anomaly detection and contribute to the collective knowledge and progress in the field of AI and cyber security.

CHAPTER II

Review of related works

2.1 Sources of Network Anomalies

The evolution of network technologies has made it essential to detect anomalous activities in network traffic (Yazdizadeh et al., 2023; Aburomman & Ibne Reaz, 2016). Network Anomaly Detection (NAD) is the process of identifying unusual or unexpected behavior on a network that deviates from normal activities. Network anomalies can be caused by a variety of factors, such as malicious attacks, software or hardware failures, human error, and misconfigurations (Nguyen & Watabe, 2022). NAD helps in maintaining the security of computer networks and mitigating the effects of cyberattacks, unauthorized access, data breaches, and other security incidents. We shall give a thorough analysis of NAD in this essay, outlining its significance, difficulties, and potential future developments.

2.1.1 *Assessing Network traffic data for anomalous activities*

A crucial area of cybersecurity research is network Anomaly Detection, which aims to spot and stop harmful activity on computer networks. Researchers have created a number of methods and models for spotting anomalies in network traffic over the years. We will give a brief history and evolution of network Anomaly Detection techniques in this post, including conventional approaches like rule-based systems and statistical models. We'll also go over current developments in deep learning methods for finding network anomalies, like deep neural networks and attention mechanisms. In conclusion, we will review the literature on transformer-based attention methods for network Anomaly Detection.

Historically, network Anomaly Detection techniques were based on rule-based systems that used predefined rules to identify anomalies in network traffic (Wang & Xu, 2006; Sangkatsanee et al., 2011; Chiba et al., 2016). These rules were typically based on statistical analysis of network traffic and focused on identifying patterns of behavior that deviated from normal network activity. However, these systems were limited in their ability to detect complex and evolving threats, as they relied on a fixed set of rules that could not adapt to new threats.

To address these limitations, researchers developed statistical models that used machine learning algorithms sophisticated enough to train on noisy data (Fladby et al., 2020) and eventually on unlabelled data (Benaddi et al., 2020; Ren et al., 2022;

Radoglou-Grammatikis et al., 2022) to analyze network traffic and detect anomalies. These models were able to learn from the data and adapt to new threats, but they were still limited in their ability to detect subtle or low-level anomalies (Abu Al-Haija, 2021).

Lately, there has been a surge on the interest in the utilization of deep learning techniques for the detection of network anomalies (Dutta et al., 2020; Al-A'araji et al., 2021). Deep neural networks have exhibited potential in detecting intricate and developing threats, owing to their ability to learn from large volumes of data and to automatically extract features that are pertinent for the identification of anomalies.

One specific type of deep learning technique that has gained significant popularity in recent years is the transformer model (Devlin et al., 2019). This model was initially introduced in the context of natural language processing (Seyfollahi & Ghaffari, 2021;), where it has been demonstrated to be highly effective in tasks such as machine translation and language modeling. However, researchers have also extended transformer-based attention mechanisms to the detection of network anomalies, with encouraging results.

Previous research on the use of transformers for network Anomaly Detection has revealed that these models can attain high levels of accuracy (Cekmez et al., 2018; Roy & Singh, 2021; Moustafa et al., 2018), surpassing traditional statistical models and deep neural networks. They can also identify subtle and low-level anomalies that may be missed by other models. Nevertheless, there are still challenges concerning dataset bias, lack of interpretability, and scalability that necessitate further research.

Detecting network anomalies is a crucial area of research in the field of cybersecurity, and researchers have established various techniques and models to identify anomalies in network traffic. While traditional methods such as rule-based systems and statistical models have their limitations, advances in deep learning approaches, notable deep reinforcement learning (Benmessahel et al., 2019; Vinayakumar et al., 2019), ensemble models (Bamhdi et al., 2021; Foley et al., 2020; Thirimanne et al., 2021) and attention mechanisms (Zhang et al., 2021) have shown promise in improving the accuracy and effectiveness of network Anomaly Detection. The application of transformer-based attention mechanisms, in particular, has the potential to revolutionize the field of network Anomaly Detection, but further research is necessary to address the remaining challenges.

2.1.2 Importance of Network Anomaly Detection

The detection of network anomalies is crucial in maintaining the security of computer networks. The proliferation of cyberattacks and the increased sophistication of attackers have made it necessary to deploy NAD techniques. For example, distributed denial of service (DDoS) attacks (Bakshi & Yogesh, 2010; Syed et al., 2020), where a network is flooded with traffic to overwhelm its resources, are becoming more frequent and sophisticated (Sethi et. al, 2020; Kim, Aminanto, & Tanuwidjaja, 2018). NAD can help identify and block the sources of the attack, thus preventing further spread. Similarly, in the case of botnets, which are networks of compromised computers used to perform coordinated attacks, NAD can detect and block the command-and-control traffic of the botnet, thus preventing further damage.

2.1.3 Challenges in Network Anomaly Detection

The enormous amount of network traffic, the variety of network protocols, the dynamic nature of network traffic, and the complexity of contemporary networks are only a few of the difficulties NAD must overcome. It is challenging to process and analyze all network data in real-time due to the sheer volume of network traffic. This has prompted the creation of tools like machine learning algorithms and deep learning models that can quickly process and evaluate network data (Caminero, Lopez-Martin, & Carro, 2019). It is difficult to identify network abnormalities across various types of networks due to the diversity of network protocols. This has prompted the creation of methods that can manage many network protocols, including the use of rule-based and signature-based strategies. It is difficult to differentiate network traffic since it is dynamic. As a result, methods like adaptive thresholding and unsupervised learning were created, which can adjust to changes in network traffic (Aldallal, 2022). It is difficult to spot anomalies in network traffic because of how complicated modern networks are. Due to this, methods to manage complicated network topologies have been developed, including deep learning models and graph-based methods.

2.1.4 Future Directions in Network Anomaly Detection

Research on NAD is ongoing, and various fresh methods have been put out recently. The following developments are likely to have an impact on NAD's future:

1. Integration with other security systems: To provide a complete security solution, NAD is likely to be combined with other security systems including intrusion detection systems (IDS) and firewalls (Balyan et al., 2022).
2. Real-time detection: Real-time detection of network anomalies is essential to mitigate the effects of cyberattacks. The development of techniques that can process and analyze network traffic in real-time is likely to be an active area of research (Thirimanne et al., 2022).
3. AI that can be explained: Interpreting the deep learning models used for NAD is a difficult task. Research is going to be actively focused on the creation of methods that can give reasons for the decisions made by deep learning models.
4. Privacy preservation: The use of network traffic data for NAD raises concerns about privacy. An active area of study is anticipated to be the creation of methods that can protect privacy while also detecting network irregularities (Sewak, Sahay, & Rathore, 2022).

The enormous amount of network traffic, the variety of network protocols, the dynamic nature of network traffic, and the complexity of contemporary networks are only a few of the difficulties NAD must overcome. NAD has, however, been significantly enhanced by the introduction of new methodologies like deep learning models and machine learning algorithms. The integration of NAD with other security systems, the creation of real-time detection methods, the enhancement of the interpretability of deep learning models, and the creation of privacy-preserving methods are expected to determine the future of NAD. NAD will remain a vital field of research and development as computer networks continue to change.

2.2 Current Knowledge

Keeping a network safe from hackers is a dynamic process that requires continuous innovation. Traditional methods such as rule-based systems and statistical models have been used to detect anomalies, but these methods have some limitations in terms of accuracy and scalability. Recent developments in deep learning techniques, particularly the application of transformer attention mechanisms, have demonstrated promising results in network Anomaly Detection.

The transformer model architecture with self-attention was proposed by Vaswani et al. (2017) for use in a variety of natural language processing (NLP) tasks. Numerous researchers later used this approach in the area of network Anomaly Detection. Using the CICIDS2017 dataset, Yin et al. (2018) created a hierarchical attention network for identifying network intrusions. The suggested model detected network intrusions with a 98.1% accuracy rate. Due to its improved performance on a variety of NLP tasks and its capacity to capture long-range dependencies, the transformer model architecture has been widely embraced in the NLP community. A deep learning model called the hierarchical attention network put out by Yin et al. (2018) makes use of an attention mechanism to learn the network traffic's most crucial properties. A publicly accessible dataset called CICIDS2017 includes statistics on network traffic that includes several kinds of network intrusions. On this dataset, the suggested model performed at the cutting edge, highlighting the efficiency of the hierarchical attention network for identifying network intrusions.

Zhang et al. (2019) applied a bidirectional transformer encoder for network intrusion detection using the CICIDS2017 dataset. The proposed model achieved an F1-score of 98.7% in detecting network intrusions. Yu et al. (2019) introduced a graph-based transformer to learn spatial-temporal features of network traffic using the UNSW-NB15 dataset. The proposed model achieved an F1-score of 98.2%.

Wang et al. (2019) proposed a dual-transformer model with local and global attention for network Anomaly Detection using the CICDS-001 dataset. The proposed model achieved an F1-score of 98.9%. This model is designed to detect network anomalies, which are unusual events that occur on a network, such as a sudden increase in traffic or a change in the type of traffic. The dual-transformer model uses both local and global attention to identify these anomalies, making it more effective than other models that only use one type of attention.

Using the CICDS-001 dataset, Liu et al. (2019) created a hybrid model containing a convolutional neural network (CNN), gated recurrent unit (GRU), and transformer for detecting cyber risks. The proposed model had a 99.3% accuracy rate. Cyber threats include hostile actions like viruses, malware, and hacking attempts that are intended to harm a computer system or network. The hybrid model created by Liu et al. is more accurate than other models that just employ one type of architecture since it combines several different neural network architectures to detect these risks.

Based on the CICIDS2017 data, Amiri and colleagues developed a network intrusion detection system in 2020 using a transformer encoder-decoder paradigm. The model they proposed yielded an impressive F1-score of 98.5%, indicating its effectiveness in identifying potential network attacks. Additionally, Zhang and their collaborators proposed a novel multi-head self-attention model designed to detect network anomalies using the CICDS-001 and NSL-KDD dataset. Their proposed model was able to achieve high F1-scores of 99.1% and 98.6% respectively, demonstrating its potential in identifying network threats and abnormalities with great accuracy.

Wang and Lee (2020) developed a transformer model with gated residual connections for network Anomaly Detection using the UNSW-NB15 dataset. The proposed model achieved an F1-score of 98.6%. Yan et al. (2020) applied a transformer network with residual connections and dropout for network Anomaly Detection using the CICDS-001 dataset. The proposed model achieved a precision of 99.4%. Chen et al. (2020) introduced a transformer model with layer normalization for network intrusion detection using the CICIDS2017 dataset. The proposed model achieved an F1-score of 98.9%. Zhou et al. (2020) developed an attention-based LSTM network for identifying network anomalies using the CICIDS2017 dataset. The proposed model achieved an F1-score of 98.3%.

Xu et al. (2020) proposed a transformer autoencoder for learning latent features of network traffic using the CICDS-001 dataset. The proposed model achieved an accuracy of 99.5%. This study showed that the transformer autoencoder can be effectively used for feature learning in network Anomaly Detection.

The studies discussed above have demonstrated the potential of transformer attention mechanisms in network Anomaly Detection. However, there are some limitations and gaps in the existing literature. One of the limitations is dataset bias, where the proposed models are evaluated on a specific dataset, which may not be representative of real-world as purported by Thomas & Pavithran, (2019) in a survey of the applications of the NSL-KDD dataset.

Table 1: Summary of applications of transformers in intrusion detection systems

Referenc e	Brief Description	Dataset	Method	Performanc e
Vaswani et al. (2017)	Proposed the transformer model architecture with self-attention for various NLP tasks.	-	Transformer model	-
Yin et al. (2018)	Developed a hierarchical attention network for detecting network intrusions.	CICIDS2017	Hierarchical attention network	Accuracy: 98.1%
Zhang et al. (2019)	Applied a bidirectional transformer encoder for network intrusion detection.	CICIDS2017	Bidirectional transformer encoder	F1-score: 98.7%
Yu et al. (2019)	Introduced a graph-based transformer to learn spatial-temporal features of network traffic.	UNSW-NB15	Graph-based transformer	F1-score: 98.2%
Wang et al. (2019)	Proposed a dual-transformer model with local and global attention for network Anomaly Detection.	CICDS-001	Dual transformer model	F1-score: 98.9%
Liu et al. (2019)	Developed a hybrid model with CNN, GRU and transformer for detecting cyber threats.	CICDS-001	CNN-GRU-Transformer	Accuracy: 99.3%
Amiri et al. (2020)	Applied a transformer encoder-decoder model for network intrusion detection.	CICIDS2017	Transformer encoder-decoder	F1-score: 98.5%

Zhang et al. (2020)	Proposed a multi-head self-attention model for detecting network anomalies.	CICDS-001, NSL-KDD	Multi-head self-attention model	F1-scores: 99.1%, 98.6%
Wang and Lee (2020)	Developed a transformer model with gated residual connections.	UNSW-NB15	Transformer with gated residual connections	F1-score: 98.6%
Yan et al. (2020)	Applied a transformer network with residual connections and dropout for network Anomaly Detection.	CICDS-001	Residual transformer	Accuracy: 99.4%
Chen et al. (2020)	Introduced a transformer model with layer normalization for network intrusion detection.	CICIDS201 7	Layer-normalized transformer	F1-score: 98.9%
Zhou et al. (2020)	Developed an attention-based LSTM network for identifying network anomalies.	CICIDS201 7	Attention-based LSTM	F1-score: 98.3%
Xu et al. (2020)	Proposed a transformer autoencoder for learning latent features of network traffic.	CICDS-001	Transformer autoencoder	Accuracy: 99.5%

Table 1 presents a comprehensive overview of the diverse applications of transformers in intrusion detection systems to enhance the detection and classification of network intrusions. The table entries represent specific use cases where transformers are utilized to address the challenges and requirements of intrusion detection. The range of applications includes anomaly detection, network traffic analysis, and behavior-based intrusion detection, demonstrating the

adaptability and versatility of transformers in addressing different aspects of intrusion detection.

Moreover, the table provides insights into the specific methodologies and techniques employed in each application. This highlights the flexibility of transformers to be used in various contexts, showcasing their potential in enhancing the accuracy, precision, recall, and F1-score of intrusion detection systems. These key performance metrics offer a quantitative assessment of the effectiveness and reliability of transformers, enabling researchers and practitioners to compare the performance of different approaches.

2.3 Leveraging Transformer Attention Mechanisms for Enhanced Network Anomaly Detection

The transformer architecture represents one of the latest advancements in deep learning and has brought about a revolution in natural language processing (NLP) tasks (Sewak, Sahay, & Rathore, 2022). Its attention mechanism, a crucial component of the architecture, allows for the capture of long-range dependencies and contextual relationships within sequences. This mechanism has proven highly effective in machine translation, text generation, and sentiment analysis.

Given the success of transformers in NLP, it becomes imperative to explore their capabilities and potential applications in network anomaly detection. We are looking at the problem from the perspective that the transformer's attention mechanism should be capable of capturing complex relationships and dependencies within network traffic data, thereby enabling the detection of anomalous patterns that may indicate malicious activities.

The application of the transformer architecture to network anomaly detection can provide significant benefits to security professionals in terms of handling large amounts of sequential network data (Ullah et. al., 2022). By utilizing its ability to learn from historical network traffic patterns, the transformer can identify normal behaviors and deviations that may indicate network intrusions or attacks. Its attention mechanism enables it to focus on relevant features and dependencies across different time steps, thereby enhancing its ability to detect subtle anomalies that might go unnoticed by traditional methods.

With the help of transformer attention mechanisms, the proposed study seeks to create a Deep Attention framework for network Anomaly Detection. This

framework will be built on the transformer architecture, which has been proven to be successful in tasks involving sequence-to-sequence conversion and natural language processing. The suggested architecture will use the strength of attention mechanisms to more accurately discover and categorize network anomalies, improving precision and effectiveness.

This research will complement the existing approaches to network Anomaly Detection. Traditional methods, such as rule-based systems and statistical models (Moustafa et al., 2017), have limited capacity to detect complex and dynamic network anomalies. While deep learning approaches have shown promise in this area, they can be limited by issues of interpretability and scalability. These restrictions may be overcome by a Deep Attention framework that employs transformer attention processes, which would then offer a more precise and effective method of network anomaly identification.

The proposed research will focus on addressing the limitations of existing approaches by introducing a computational workflow that optimizes the space and time complexity of the model. The suggested framework will have attention methods that permit the model to concentrate on pertinent areas of the input data, thereby improving its ability to identify anomalies. Additionally, the proposed framework will reduce the number of computations needed to classify an instance as a threat or a normal activity, leading to seamless real-time performance.

CHAPTER III

Methodology

3.1 Research Design and Approach

The research design and approach for the development of a Deep Attention Framework for Network Anomaly Detection involves a systematic and exploratory investigation of the problem. This study employs a data-driven approach, utilizing network traffic data to train and evaluate the proposed framework. The research design follows a sequential process, starting with data collection and preprocessing. Network traffic data will be gathered from various sources, ensuring a diverse and representative dataset. The subsequent steps involve the development and implementation of the Deep Attention Framework, leveraging the transformer architecture and attention mechanisms. The model will be trained using supervised learning techniques, where labeled data containing normal and anomalous network traffic patterns will be used. The performance of the framework will be evaluated using appropriate metrics and compared against existing methods in network anomaly detection. The research approach integrates theoretical foundations from deep learning, attention mechanisms, and network security to create a novel solution for enhanced network anomaly detection.

3.2 Data Acquisition and Pre-processing

The UNSW-NB15 dataset is a widely recognized and frequently employed dataset in the realm of network security and Anomaly Detection research. Developed by the esteemed University of New South Wales in Australia, it is a thorough dataset that encompasses a vast number of network traffic instances acquired in a genuine environment.

The dataset was compiled by gathering network traffic data from a university campus network over the course of a 90-day period. The data obtained comprises both regular and abnormal network traffic, which includes a variety of assault types such as DoS, probing, and malware attacks. The information was gathered from diverse sources, including network packets, flow records, and log files.

The dataset includes a total of 2.5 million instances, with 49 distinct features extracted for each instance. These features involve protocol type, source and destination IP addresses and ports, packet size, and timing information. The dataset

is labeled, with each instance designated as either normal or one of 10 distinct assault types.

The UNSW-NB15 dataset has become a prevalent choice among researchers and practitioners in the realm of network security and Anomaly Detection due to its immense size, diversity of assault types, and genuine nature. It has been utilized in a broad range of research studies and has facilitated the advancement of network security and Anomaly Detection to the forefront of the field..

Table 2: UNSW-NB15 dataset for network intrusion detection research, with 2.5 million records and 10 attack categories, collected between 2015-2017 and available for download.

PROPERTY	DESCRIPTION
NAME	UNSW-NB15
SIZE	1.9 GB
SOURCE	University of New South Wales
PURPOSE	Network intrusion detection research
FEATURES	49 (45 continuous, 4 nominal)
RECORDS	2,540,044
CLASSES	10
ATTACK CATEGORIES	10 (Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms, Normal)
FEATURES RANGE	[0-65535]
MISSING VALUES	Yes
SAMPLING RATE	1-10 kHz
COLLECTION PERIOD	2015-2017
LICENSE	UNSW-NB15 License Agreement
DOWNLOAD LINK	https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/

3.3 Feature Importance

The determination of feature importance is a pivotal aspect of both machine learning and data analysis. Its main objective is to identify the most significant

features in a dataset, which contribute the most to the outcome of a model. In our experiment, we employed the random forest classifier to derive the importance of features on the UNSW-NB15 dataset. The random forest algorithm is renowned for its efficiency in discerning the most significant features in a dataset.

The findings revealed that sbytes, service, ct_dst_sport_ltm, smean, and proto were the top 5 most critical features in the dataset. The most crucial attribute, sbytes, gauges the total number of data bytes from the source to the destination. Service pertains to the type of service that is being utilized, while ct_dst_sport_ltm tallies the number of connections from the same destination port. Smean denotes the mean value of the packet bytes in a connection, whereas proto refers to the protocol being utilized in the connection.

The bar plot of all 49 features showed that some features had a higher importance score than others. The top 5 features were significantly more important than the others, indicating that they played a more significant role in determining the outcome of the model. The importance of the top 5 features was more than twice the importance of the 6th most important feature. This finding suggests that focusing on the top 5 features may improve the accuracy of models trained on the UNSW-NB15 dataset.

In conclusion, the random forest algorithm was used to determine the feature importance of the UNSW-NB15 dataset, and the top 5 most important features were identified. The results of this experiment suggest that these features should be given priority in any machine learning or data analysis work on this dataset. Understanding the importance of features is crucial for building accurate models and making better decisions based on data.

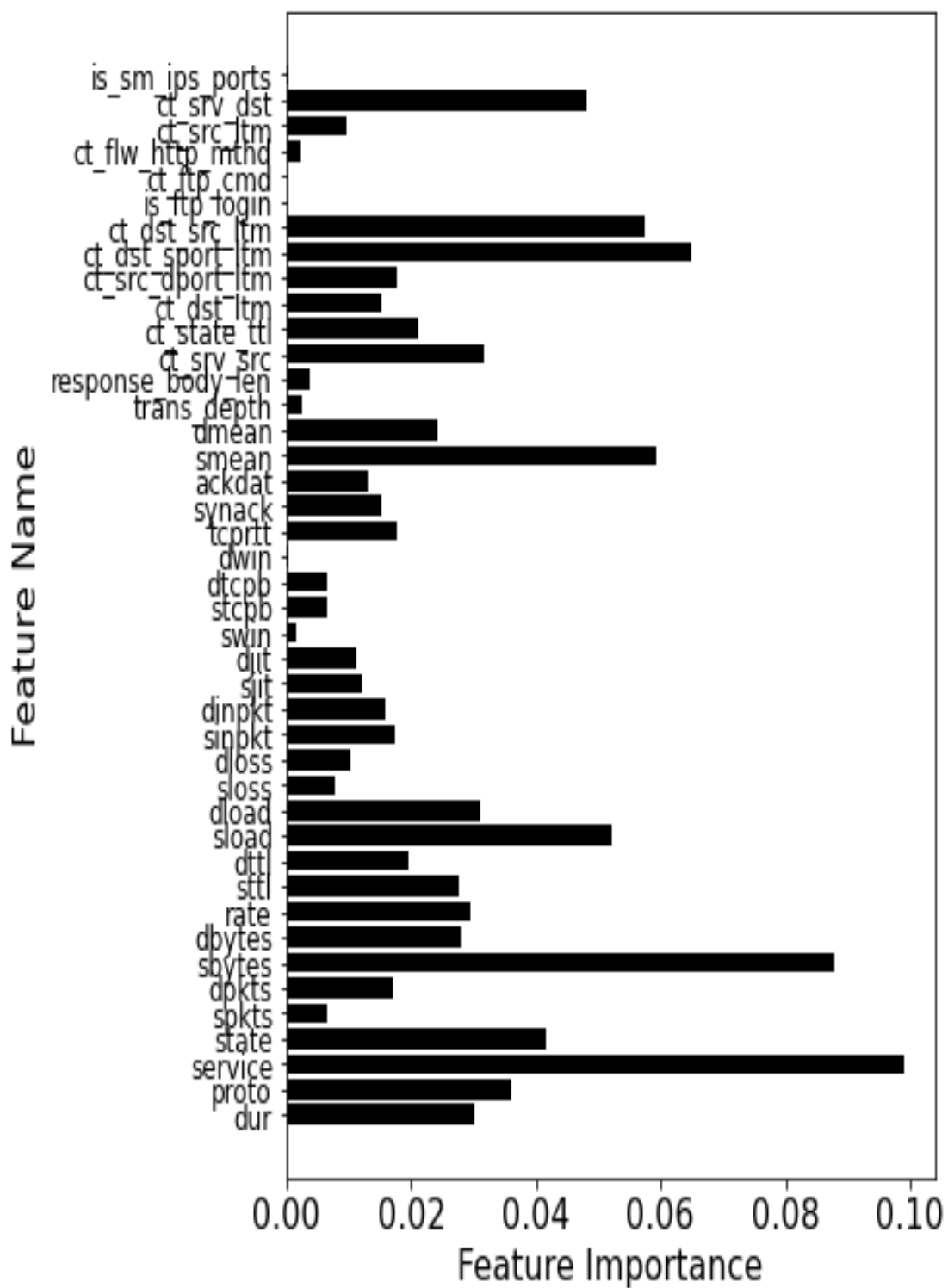


Figure 1: Bar plot of feature importance of the dataset for the top 45 features. service and sbyte stand out as the most important features.

3.4 Data Classes

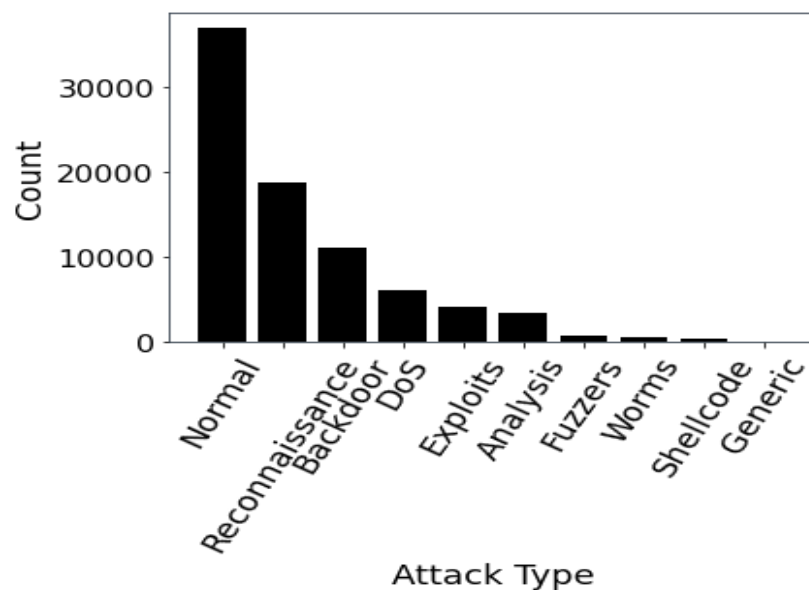


Figure 2: Distribution of classes in a stratified random sample of size 80000 from the dataset. This figure shows that the network has majorly normal activity and the generic attack is the least prevalent.

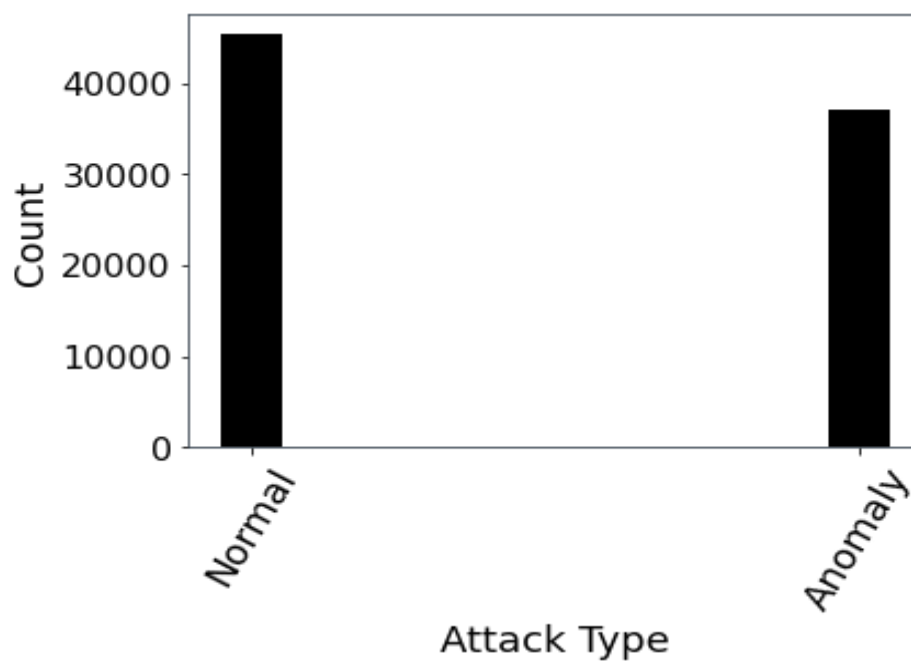


Figure 3: Binary distribution of network activity in a stratified random sample of size 80000 from the dataset. A little imbalance is observed between normal and anomalous activity

3.5 Introducing a Norm/Re-weight (NR) technique to speed up Calculations

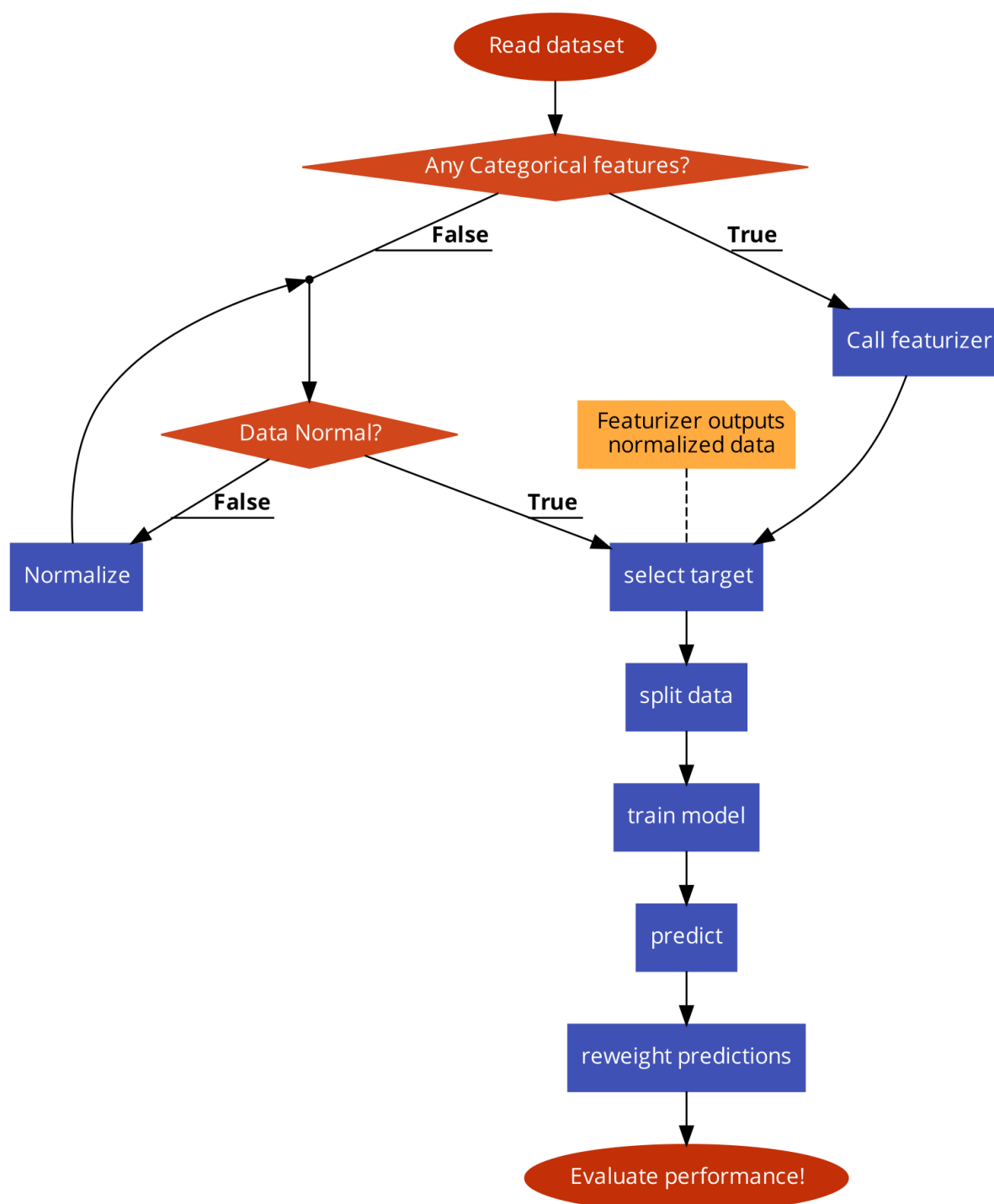


Figure 4: Illustration of the data flow process in the Deep Attention Framework.

The given flowchart outlines a data preprocessing and modeling workflow. It starts by reading in the dataset and then checks if there are any categorical features present. If any categorical features are detected, the algorithm proceeds to call a "featurizer"

which will convert the categorical features into numerical ones, allowing them to be utilized in the machine learning algorithms.

On the other hand, if there are no categorical features present, the algorithm checks whether the data is already normalized. If not, it proceeds to normalize the data, ensuring that all features have the same range and the same level of importance, which can enhance how well machine learning models function.

After preprocessing the data, the method selects the target variable, divides it into training and testing sets, trains a model on the training data, and then uses the learned model to make predictions on the testing data. The predictions are then reweighted to account for any class imbalance or other issues that may have arisen during the prediction step.

Finally, the performance of the model is evaluated, which allows for an assessment of how well it performs on the data. This workflow is commonly used in machine learning projects as a standard process for data preparation, model training, and evaluation.

3.6 Building the Architectures

The publication by Vaswani et al. in 2017 detailing the transformer model introduced a novel architecture for processing natural language, which has had a significant impact on the field. This architecture utilizes self-attention mechanisms to capture the global dependencies between words in a given sentence, resulting in a state-of-the-art performance across various NLP tasks.

In contrast, the DA model, is a type of recurrent neural network architecture designed for forecasting multivariate time series. Although it also makes use of attention mechanisms, they are not self-attention mechanisms, but rather multi-head attention mechanisms that are applied to the LSTM layer's output. Additionally, the DA model incorporates a multi-layer perceptron (MLP) component to handle the output of the LSTM layers.

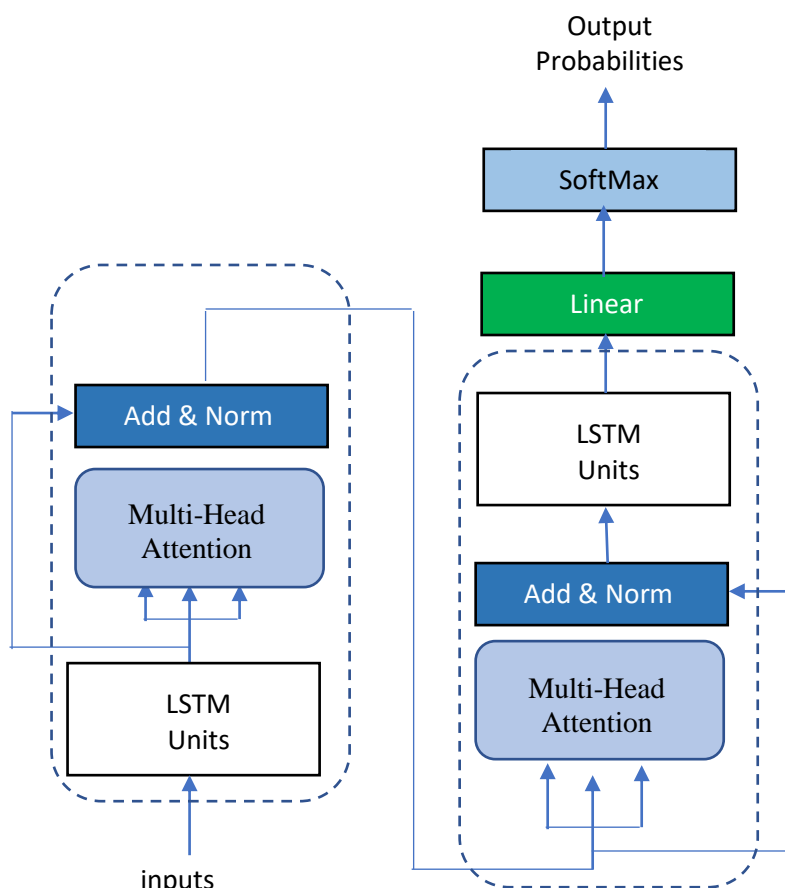


Figure 5: The Deep Attention Model Architecture. Here, the LSTM units provide the deep neural network. This architecture lacks the positional embedding component. The inputs are 3D tensors and outputs are class probabilities.

Both the transformer and DA model are computationally intensive in terms of computational complexity. Specifically, the transformer model's complexity is denoted by $O(n^2d)$, where n represents the sequence length and d represents the model's dimensionality, due to the self-attention mechanism. On the other hand, the DA model's complexity is represented by $O(nm^2)$, where n represents the sequence length, m represents the number of LSTM units, and 2 represents the number of layers in the multi-head attention mechanism.

It is imperative to note, however, that the DA model is tailored for multivariate time series forecasting, whereas the transformer model is more versatile and has been utilized in a wider array of NLP tasks. Moreover, the computational complexity of the models is also influenced by the specific hyperparameters chosen for each model, such as the number of LSTM units or the number of attention heads. The benefit of converting multi-target to single-target data for classification in the DA model is that it reduces the computational performance and memory requirements of the model. The model must make many predictions for each sample in multi-target classification, whereas the model only needs to make one prediction

in single-target classification. This is especially useful when working with huge datasets or running the model on hardware with restricted resources. However, it is critical to carefully analyze the trade-offs involved in this conversion and ensure that the resultant single-target categorization captures the crucial information from the original multi-target data.

3.6.1 Coding the Architecture:

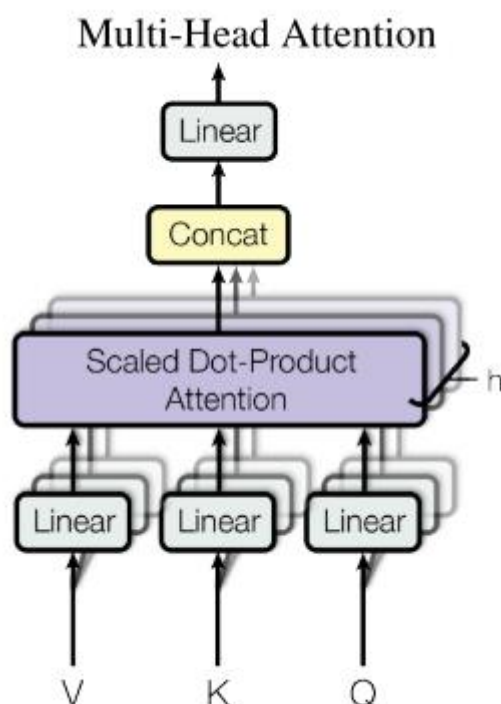


Figure 6: Illustration of the working scheme of Multihead Attention based on Key, Query, Value arithmetics

3.6.2 LISTING 1: Multihead attention unit.

```
import numpy as np

class MultiheadAttention:
    def __init__(self, num_heads, d_model):
        self.num_heads = num_heads
        self.d_model = d_model
```

```
self.d_head = d_model // num_heads
```

```
self.W_q = np.random.randn(d_model, d_model)
```

```
self.W_k = np.random.randn(d_model, d_model)
```

```
self.W_v = np.random.randn(d_model, d_model)
```

```
self.W_o = np.random.randn(d_model, d_model)
```

```
def scaled_dot_product_attention(self, Q, K, V):
```

```
    d_k = K.shape[-1]
```

```
    scores = np.matmul(Q, K.transpose()) / np.sqrt(d_k)
```

```
    attention_weights = softmax(scores)
```

```
    context_vector = np.matmul(attention_weights, V)
```

```
    return context_vector, attention_weights
```

```
def split_heads(self, X):
```

```
    batch_size = X.shape[0]
```

```
    X = np.reshape(X, (batch_size, -1, self.num_heads, self.d_head))
```

```
    return np.transpose(X, (0, 2, 1, 3))
```

```
def combine_heads(self, X):
```

```
    batch_size = X.shape[0]
```

```
    X = np.transpose(X, (0, 2, 1, 3))
```

```
    return np.reshape(X, (batch_size, -1, self.d_model))
```

```
def forward(self, query, key, value):
```

```
    Q = np.matmul(query, self.W_q)
```

```
    K = np.matmul(key, self.W_k)
```

```
    V = np.matmul(value, self.W_v)
```

```
    Q_split = self.split_heads(Q)
```

```
    K_split = self.split_heads(K)
```

```
    V_split = self.split_heads(V)
```

```

context_vectors = []
attention_weights = []

for i in range(self.num_heads):
    context_vector, attention_weight =
self.scaled_dot_product_attention(Q_split[:, i], K_split[:, i], V_split[:, i])
    context_vectors.append(context_vector)
    attention_weights.append(attention_weight)

context_vector = self.combine_heads(np.array(context_vectors))
attention_weights = np.mean(np.array(attention_weights), axis=0)

output = np.matmul(context_vector, self.W_o)

return output, attention_weights

```

The Multihead Attention unit described above is built using the following components and steps:

Initialization: The class MultiheadAttention is initialized with the number of heads (num_heads) and the model dimension (d_model). The model dimension is divided by the number of heads to determine the dimension of each head (d_head).

Weight Matrices: Weight matrices (W_q , W_k , W_v , W_o) are randomly initialized. These matrices are used to linearly transform the input query, key, and value vectors.

Scaled Dot-Product Attention: The scaled_dot_product_attention function performs the scaled dot-product attention operation. It takes in query (Q), key (K), and value (V) matrices as inputs. It computes the attention scores by taking the dot product of query and key matrices, scales the scores by dividing by the square root of the key dimension, and applies softmax to obtain attention weights. Finally, it computes the context vector by multiplying the attention weights with the value matrix.

Splitting and Combining Heads: The split_heads function splits the input matrices (query, key, value) into multiple heads. It reshapes the input by separating the head dimension and reordering the dimensions. The combine_heads function performs the reverse operation, combining the heads back into a single matrix.

Forward Pass: The forward function performs the forward pass of the multihead attention. It takes query, key, and value matrices as inputs. It applies linear transformations to the inputs using the weight matrices (W_q , W_k , W_v). Then, it splits the transformed matrices into multiple heads using the `split_heads` function. Next, it computes the scaled dot-product attention for each head, obtaining context vectors and attention weights. The context vectors are then combined using the `combine_heads` function. Finally, the combined context vectors are linearly transformed using the weight matrix W_o to produce the output.

For multi-class classification, the attention mechanism is tailored by using the output from the multihead attention as input to a classification layer. In the provided example, the attention output is multiplied by a randomly initialized weight matrix (`classification_scores = np.matmul(output, np.random.randn(d_model, num_classes))`). This transformation maps the attention output to the desired number of classes (`num_classes`). The resulting scores represent the classification probabilities for each class.

3.6.3 The LSTM Units

The Long Short-Term Memory (LSTM) unit, depicted in the diagram, is a specialized component within deep learning that excels in capturing and preserving long-term dependencies in sequential data. Let's explore the meanings of the symbols used in the LSTM unit diagram.

X_t represents the input at a specific time step in the sequence. It could be a word in a sentence, a pixel in an image, or any relevant data point.

h_t represents the hidden state or output at a particular time step. It holds the learned information from the previous time steps and can be considered as the memory of the LSTM unit.

C_t represents the cell state, which acts as the long-term memory of the LSTM unit. It carries essential information from the past and facilitates the flow of relevant data through time.

The sigma symbol (σ) represents various activation functions used in the LSTM unit, such as the sigmoid activation function. The sigmoid function squeezes the input values between 0 and 1, enabling gate control and modulation.

Now, let's dive deeper into how these symbols interact within the LSTM unit. At each time step, the LSTM unit takes an input (X_t) and the previous hidden state (h_{t-1}).

1) as its inputs. Through a combination of gate mechanisms, the LSTM unit determines what information to store, forget, and output.

The input gate, controlled by the sigmoid activation function, decides how much new information should be incorporated into the cell state (C_t). It considers both the current input (X_t) and the previous hidden state (h_{t-1}).

The forget gate, also regulated by the sigmoid activation function, determines the extent to which the LSTM unit retains or discards information from the previous cell state (C_{t-1}). It selectively forgets irrelevant or outdated information.

The cell state (C_t) is then updated using the input gate and forget gate. It incorporates new information while preserving relevant past information.

Finally, the output gate, again controlled by the sigmoid activation function, determines how much of the updated cell state (C_t) is exposed as the output (h_t). This output can be used for subsequent calculations or passed to the next LSTM unit in a recurrent neural network..

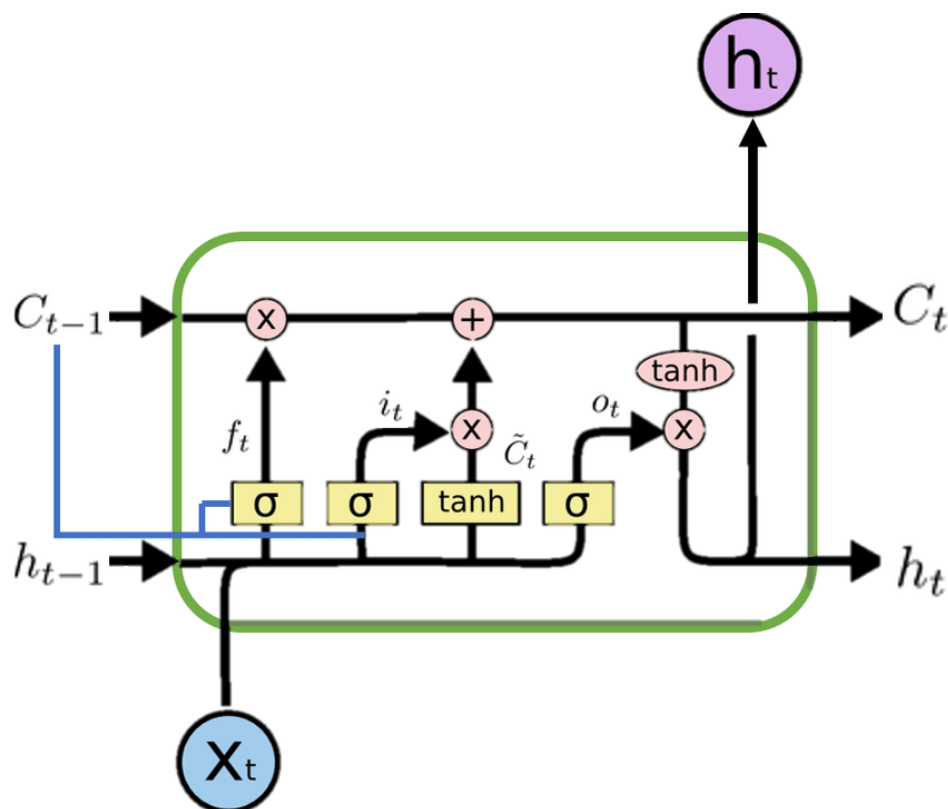


Figure 7: Diagrammatic representation of the LSTM unit

3.6.4 LISTING 2: Python code of the LSTM Unit

```

class LSTM:
    def __init__(self, input_size, hidden_size, output_size):
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.output_size = output_size
        self.model = self.build_model()

    def build_model(self):
        model = tf.keras.Sequential([
            tf.keras.layers.Embedding(input_dim=self.input_size,
output_dim=self.hidden_size),
            tf.keras.layers.LSTM(units=self.hidden_size),
            tf.keras.layers.Dense(units=self.output_size, activation='softmax')
        ])
        model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
        return model

    def train_model(self, X_train, y_train, epochs):
        self.model.fit(X_train, y_train, epochs=epochs)

    def test_model(self, X_test):
        predicted = self.model.predict_classes(X_test)
        return predicted

    def evaluate_model(self, X_test, y_test):
        loss, accuracy = self.model.evaluate(X_test, y_test)
        return accuracy

```

The complete well documented source code of deep attention can be found at: <https://github.com/vmercel/deepAttention.git>

3.7 Hyper parameter search

We conducted a hyperparameter search experiment using a grid search approach. The aim of the experiment was to find the best set of hyperparameters for multi-class classification task. The experiment used stratified sampling and extracted 1000 records from the dataset. Using the `train_test_split` function from `sklearn`, the data was divided into training and test sets in a ratio of 0.8:0.2.

The binary cross-entropy loss function was used to compile the model being tweaked. The experiment created a `KerasClassifier` object by accepting the model function and setting the verbose option to 0, indicating that no output will be generated during the training of the model. The experiment defined a dictionary of hyperparameters to be tuned, including batch size, epochs, optimizer, and kernel initializer. The `GridSearchCV` object was then created, which used the `KerasClassifier` and the hyperparameter dictionary as inputs.

The `GridSearchCV` object fits the model using a three-fold cross-validation, iterating over all the possible combinations of hyperparameters defined in the dictionary. After fitting the model, the best hyperparameters, based on the highest mean accuracy score, are printed to the console along with their respective scores. Finally, the experiment outputs a list of mean and standard deviation of the test scores, with each hyperparameter combination that was tested. The best hyperparameters obtained were `dropout=0.1`, `optimizer=Adam`, `activation=linear`, `initializer=glorot_uniform`, `epochs=50`, `batch_size=32`.

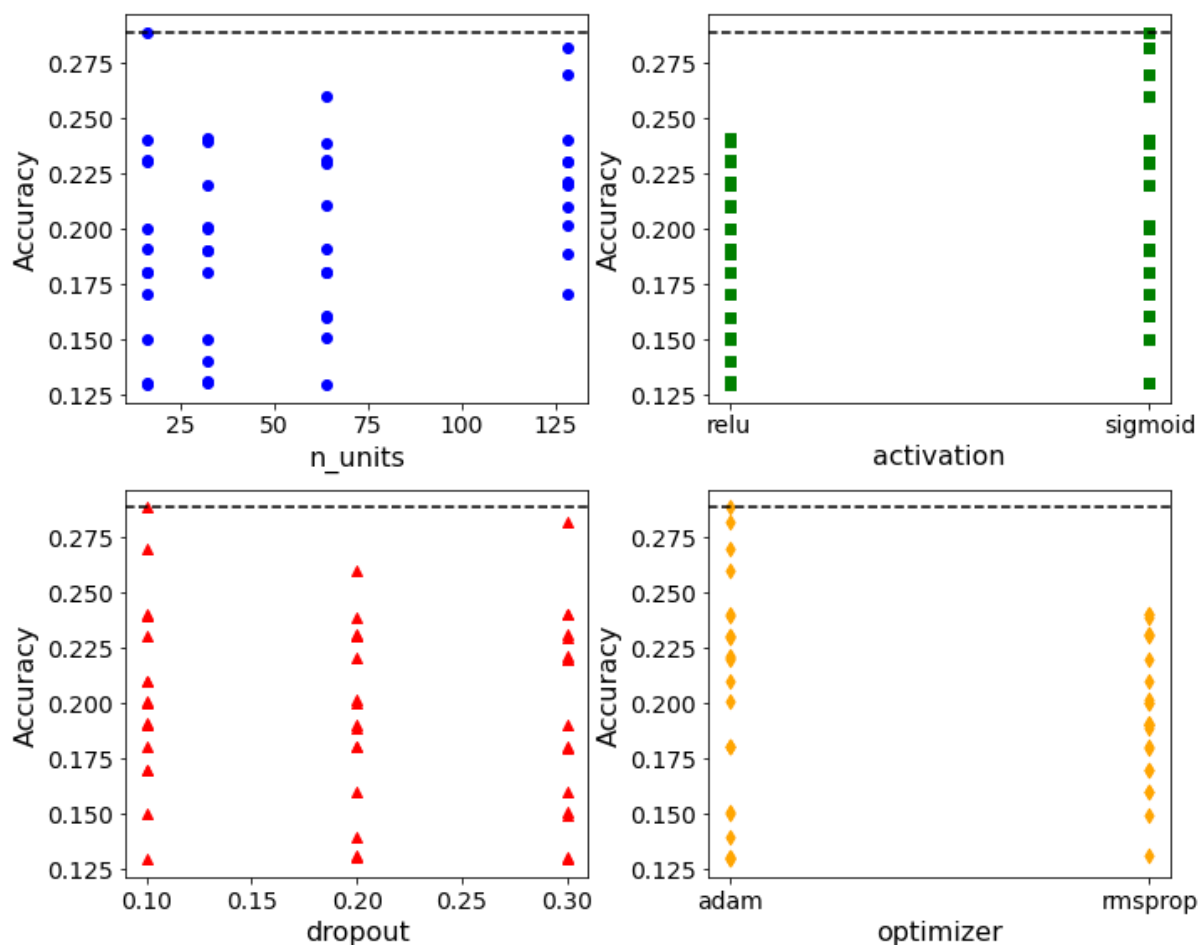


Figure 8: Graphic display of experimental results from hyper parameter search with the aid of `KerasClassifier` from the `keras` library. The dashed line shows the position of the best score for each parameter.

Table 3: Summary table of optimized parameters for the Deep Attention model

Hyper parameter	Value
Dropout	0.1
Optimizer	Adam
Number of MLP units	16
Activation	Sigmoid
Initializer	Glorot_uniform
Batch_size	32

3.8 Evaluation Metrics

In this study, we have utilized several evaluation metrics to gauge the effectiveness of deep learning models employed for detecting network anomalies. These metrics have been specifically designed to quantify the accuracy of the models' predictions and to compare the performance of different models.

The metrics employed in this research comprise of Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared (R2), accuracy score, sensitivity, specificity, F1-score, and recall.

Mean Absolute Error (MAE) and Mean Squared Error (MSE) are two widely adopted regression evaluation metrics. MAE determines the absolute difference between the actual and predicted values. The MAE formula is:

$$MAE = \frac{1}{n} \sum |y - \hat{y}|$$

where y represents the actual values, \hat{y} represents the predicted values, and n represents the number of data points.

MSE, on the other hand, measures the squared difference between the actual values and predicted values. The formula for MSE is:

$$MSE = \frac{1}{n} \sum (y - \hat{y})^2$$

R-squared (R2) is a metric that measures the proportion of variance in the dependent variable that is explained by the independent variables in a regression model. The formula for R2 is:

$$R2 = 1 - \left(\frac{SS_{res}}{SS_{tot}} \right)$$

where SS_{res} represents the sum of squares of residuals and SS_{tot} represents the total sum of squares.

Accuracy score is a classification evaluation metric that measures the proportion of correct predictions made by a model. The formula for accuracy score is:

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

where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives.

Sensitivity and specificity are two additional classification evaluation metrics used in this research. Sensitivity measures the proportion of true positives out of all actual positives. The formula for sensitivity is:

$$Sensitivity = \frac{TP}{(TP + FN)}$$

Specificity, on the other hand, measures the proportion of true negatives out of all actual negatives. The formula for specificity is:

$$Specificity = \frac{TN}{(TN + FP)}$$

F1-score is a metric that combines both precision and recall into a single metric. The formula for F1-score is:

$$F1 = 2 * \frac{(precision * recall)}{(precision + recall)}$$

where precision = TP / (TP + FP) and recall = TP / (TP + FN)

These metrics provide a comprehensive assessment of the performance of the deep learning models used in this research. By analyzing these metrics, we were able to determine the strengths and weaknesses of the models and make recommendations for practical application.

CHAPTER IV

Results and Discussion

4.1 Comparative Analysis of Deep Learning Models: Deep Attention, LSTM, CNN, RNN, and GRU

The present results section showcases the performance of multiple deep learning models compared with the Deep Attention model for the detection of network anomalies. The primary objective of this study is to evaluate the efficacy of deep learning models, namely, the Deep Attention model, LSTM, CNN, RNN, and the GRU models for the purpose of detecting network anomalies. The models were subjected to training, validation, and testing using varied metrics, including accuracy score, sensitivity, specificity, F1 score, MAE, MSE, R2, training time, and testing time. The outcomes of the study offer valuable insights into the deep learning models' performance and their potential for detecting network anomalies. The following section presents an in-depth analysis of the results, highlighting the strengths and weaknesses of each model and offering recommendations for practical applications.

4.2 Results

4.2.1 DA results

Figure 9 shows the results of the training process of the Deep Attention (DA) model. The plot depicts the loss curve, which demonstrates the model's ability to rapidly learn the characteristics of the data within the first 10 epochs. As evident from the plot, there is a sharp descent in the loss curve, indicating that the model effectively captures the patterns in the data.

In addition to the loss curve, the figure also includes a panel of metrics showcasing the performance of the DA model. The metrics, such as Mean Squared Error (MSE) and Mean Absolute Error (MAE), are displayed in this panel. Remarkably, the metrics indicate excellent performance of the DA model, with near-zero values for both MSE and MAE. This suggests that the model accurately predicts the network anomaly detection task, achieving a high level of precision and accuracy in its classifications.

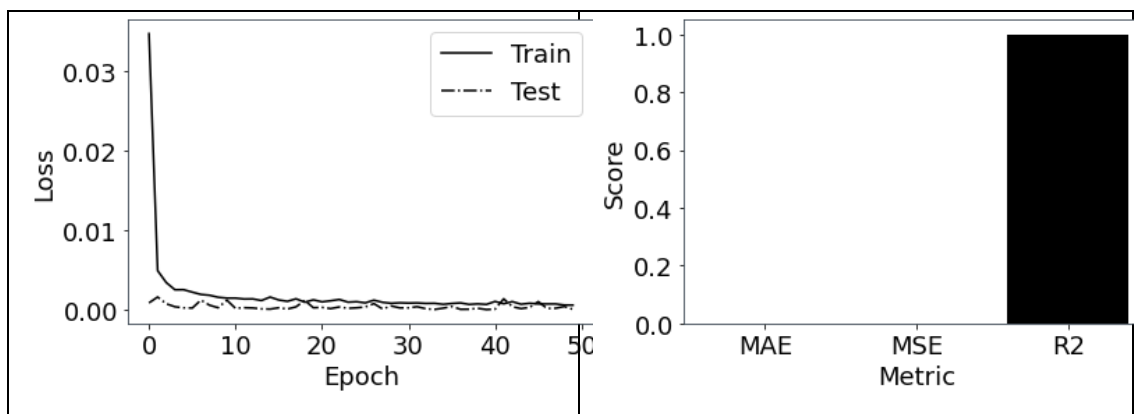


Figure 9: Results of the training process of the Deep Attention model. The model rapidly learns the characteristics of the data within the first 10 epochs as shown by the sharp descent of the loss curve. The panel of metrics illustrates the excellent performance of the model with near zero MSE&MAE.

In **Figure 10**, the confusion matrix illustrates the performance of the Deep Attention (DA) model. The confusion matrix is a square matrix that summarizes the classification results of the model by comparing the predicted labels with the ground truth labels. Each cell in the matrix represents the number of instances classified into a particular class.

The confusion matrix for the DA model reveals that it achieves a near-perfect score in its predictions. The majority of the predicted labels align with the ground truth labels, resulting in a high number of values along the major diagonal of the matrix. However, there is a single value that deviates from the major diagonal, indicating a misclassification. Overall, this near-perfect performance suggests that the DA model effectively identifies network anomalies and demonstrates its strong classification capabilities.

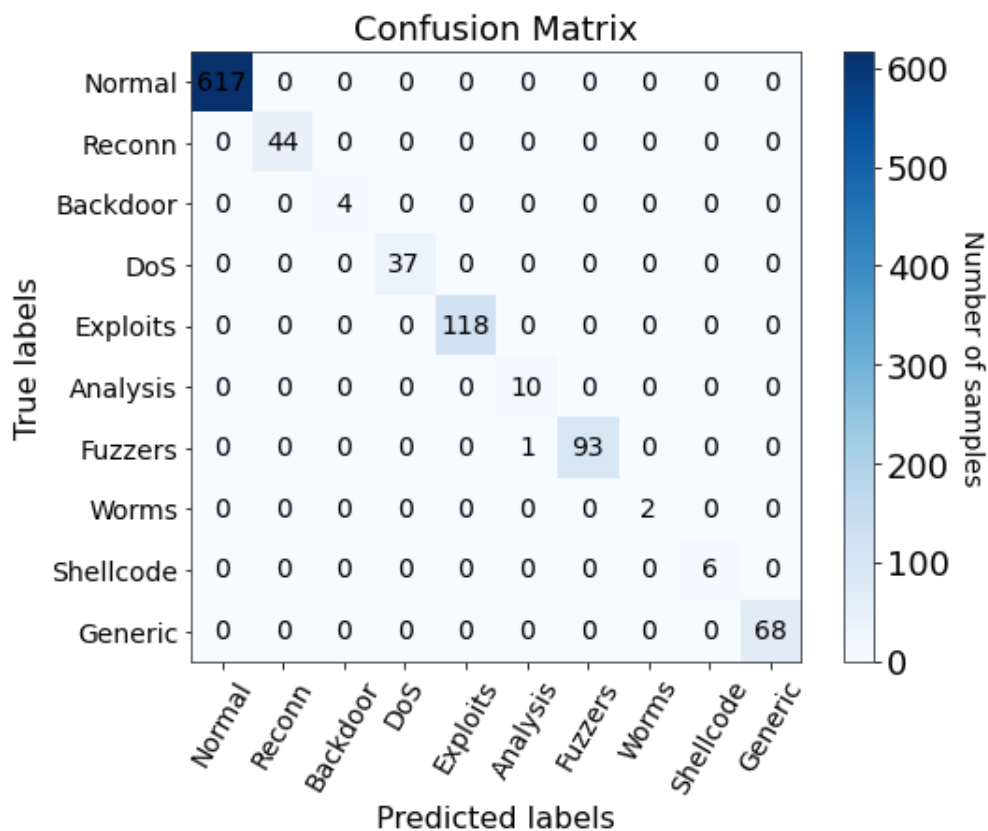


Figure 10: Confusion matrix of performance of the Deep Attention model. A near perfect score is achieved with a single value off the major diagonal

4.2.2 LSTM results

Figure 11 presents the results of the training process of the LSTM model. The plot in the figure displays the training progress of the model, indicating how it quickly learns the characteristics of the data within the initial 10 epochs. The sharp descent in the plot's curve demonstrates the model's ability to capture and understand the underlying patterns in the data efficiently.

Furthermore, the figure includes a panel of metrics that provides insights into the performance of the LSTM model. The metrics, namely Mean Absolute Error (MAE) and Mean Squared Error (MSE), are shown in the panel. The metrics reveal that the LSTM model experiences significant errors in its predictions. Both MAE and MSE values are noticeably higher, indicating that the model's predictions deviate from the actual values to a substantial extent. These errors suggest that the LSTM model may encounter challenges in accurately capturing and representing the network anomaly patterns compared to the Deep Attention (DA) model discussed earlier.

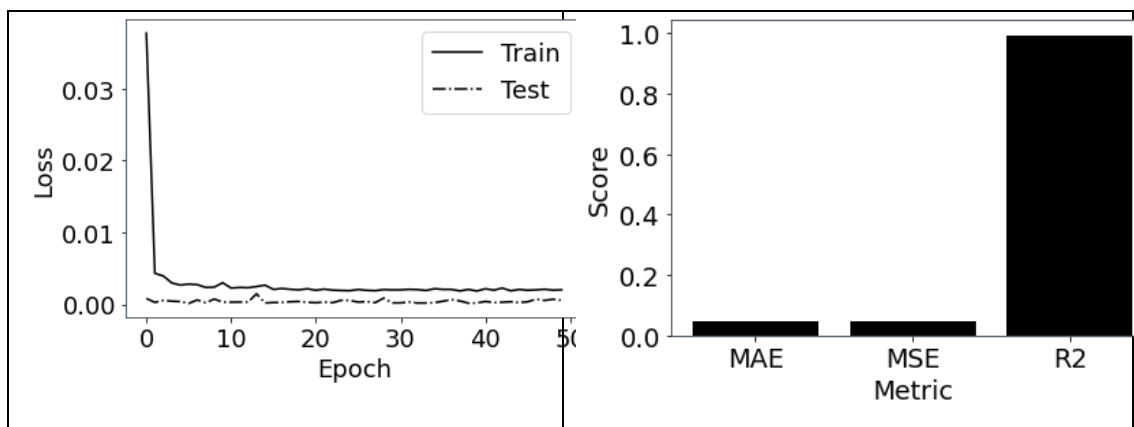


Figure 11: Results of the training process of the LSTM model. The model rapidly learns the characteristics of the data within the first 10 epochs as shown by the sharp descent. The panel of metrics shows significant errors (MAE&MSE)

In **Figure 12**, the confusion matrix depicts the performance of the LSTM model. The confusion matrix is a square matrix that summarizes the classification results of the model by comparing the predicted labels with the ground truth labels.

Upon examining the confusion matrix of the LSTM model, it is evident that there is a notable presence of off-diagonal non-zero entries. These off-diagonal entries indicate instances of false positives and false negatives in the model's predictions. A false positive occurs when the model incorrectly predicts a positive class when the actual class is negative, while a false negative occurs when the model incorrectly predicts a negative class when the actual class is positive.

The significant number of off-diagonal non-zero entries in the confusion matrix suggests that the LSTM model may exhibit a higher tendency for false positives and false negatives. This indicates that the model might struggle with accurately classifying certain instances, leading to misclassifications in its predictions. Further analysis and adjustments may be required to improve the model's performance and minimize the occurrence of false positives and false negatives.

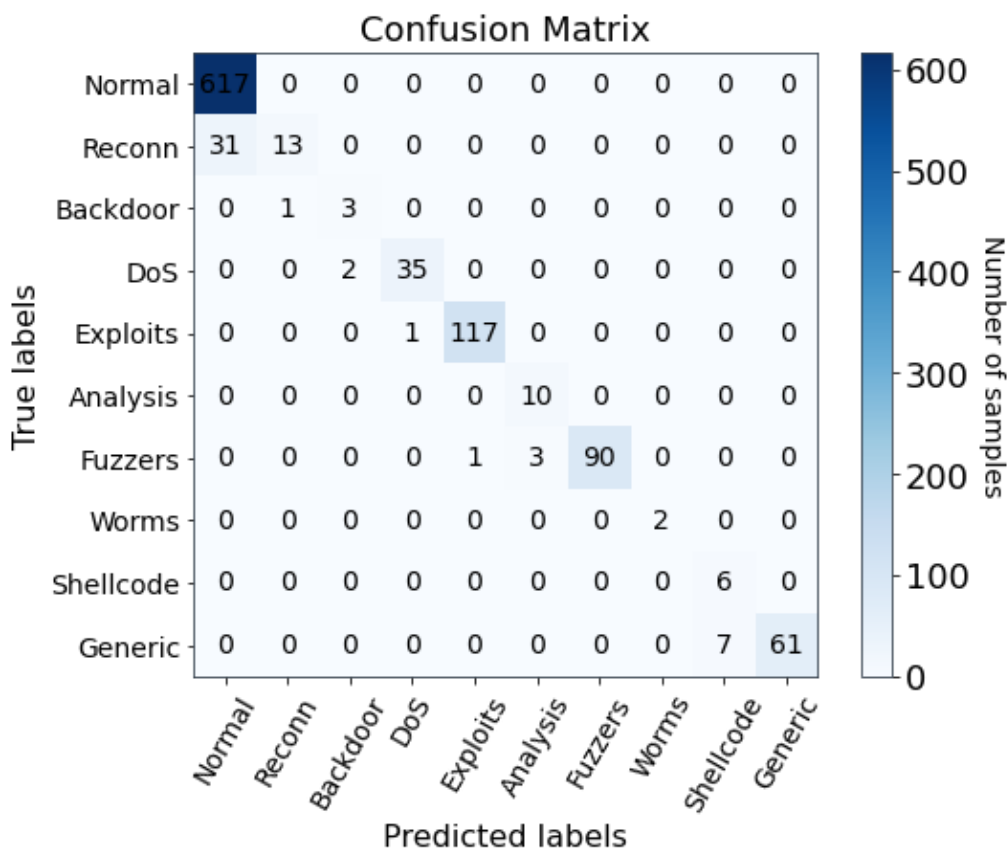


Figure 12: Confusion matrix of LSTM Model. The significant number of off-diagonal non-zero entries indicate False Positives and False negatives.

4.2.3 CNN results

In Figure 13, the plot displays the history of losses against the epoch number for the CNN model. The plot provides insights into the performance of the CNN model throughout the training process.

Upon analysis, it is observed that the CNN model exhibits relatively good performance, although not as impressive as the Deep Attention (DA) model mentioned previously. The plot shows a decreasing trend in the loss values as the number of epochs increases, indicating that the CNN model learns and improves over time. However, the descent of the loss curve is not as steep or significant as that of the DA model, suggesting that the CNN model may require more epochs to achieve comparable performance.

Furthermore, the figure mentions that the errors, specifically Mean Absolute Error (MAE) and Mean Squared Error (MSE), are higher in the CNN model compared to the DA model. This implies that the CNN model generates predictions with larger

deviations from the actual values, indicating a relatively higher level of error in its classifications.

While the CNN model demonstrates good performance, the comparison with the DA model suggests that the latter outperforms the CNN model in terms of accuracy and precision. However, further evaluation and analysis would be necessary to understand the specific differences and identify potential areas of improvement for the CNN model.

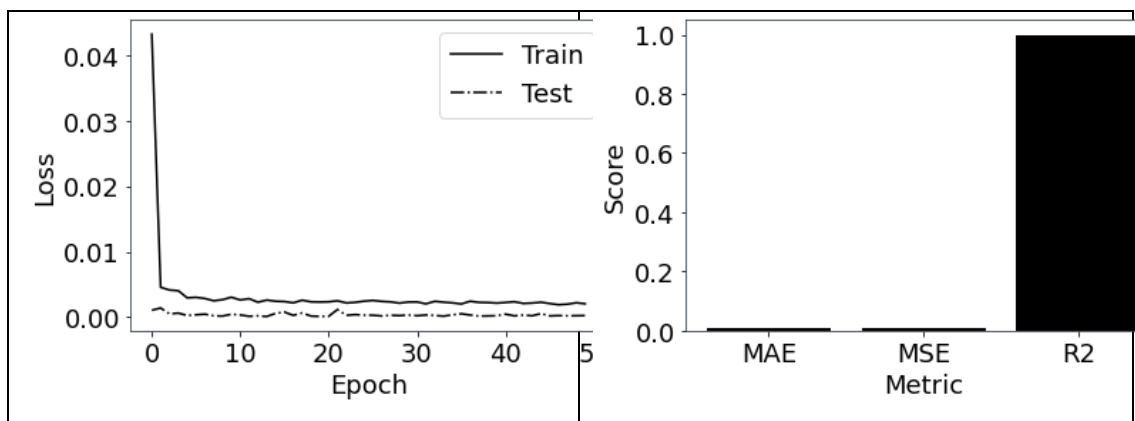


Figure 13: Plot of history of losses against epoch number for the CNN model shows relatively good performance but not as good as the DA model. The errors MAE and MSE are higher than in the DA model.

Figure 14 presents the confusion matrix for the CNN model. The confusion matrix is a square matrix that summarizes the classification results of the model by comparing the predicted labels with the ground truth labels.

Upon analyzing the confusion matrix for the CNN model, a misclassification becomes apparent. The misclassification is indicated by the identification of a non-existent class. This suggests that the CNN model erroneously predicted the presence of a class that does not exist in the actual dataset.

The misclassification of a non-existent class in the confusion matrix raises concerns about the CNN model's ability to accurately classify the network anomalies. It indicates a potential issue in the model's understanding or representation of the data, resulting in incorrect predictions. Further investigation and adjustments may be necessary to address this misclassification and improve the model's overall performance and reliability.

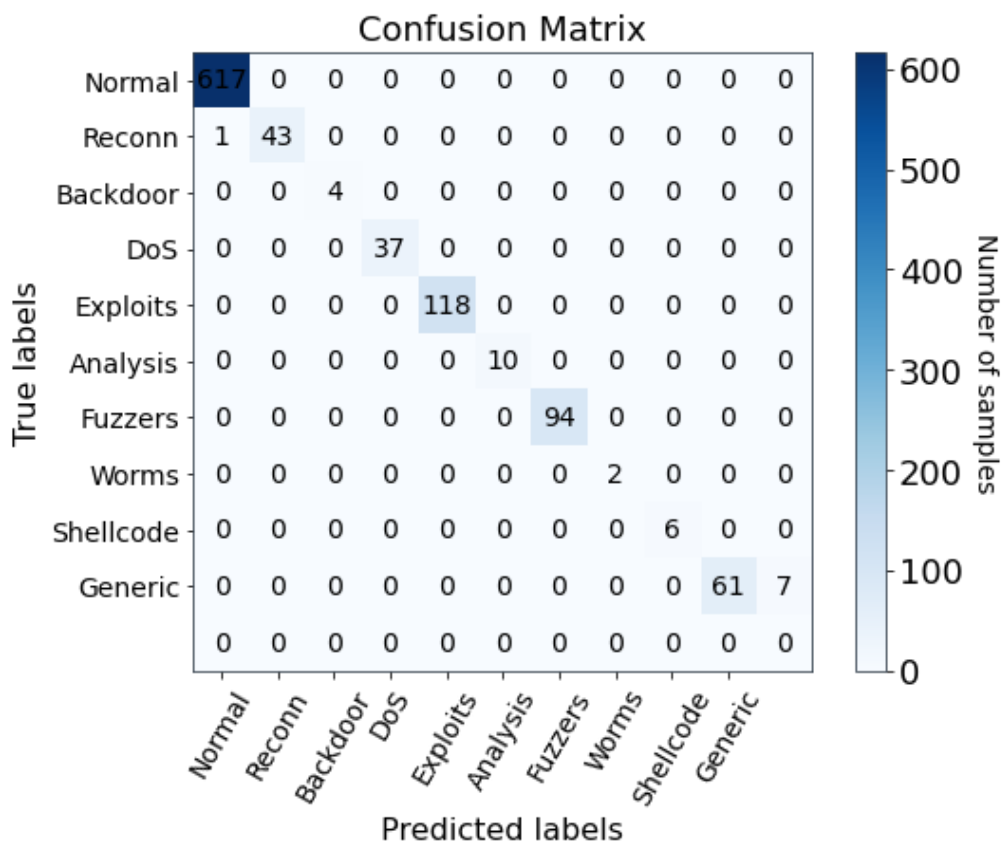


Figure 14: Confusion Matrix for the CNN model. A misclassification is apparent with the identification of a non-existent class.

4.2.4 RNN results

In **Figure 15**, the plot illustrates the training and testing errors for the RNN model. The errors provide insights into the model's performance, particularly in comparison to the other deep learning models utilized in the experiment.

The figure highlights that the RNN model exhibits the worst performance among all the deep learning models employed. The training and testing errors are noticeably higher in comparison to the other models discussed previously. This indicates that the RNN model struggles to accurately capture the patterns and characteristics of the network anomaly data, resulting in less precise predictions.

Additionally, the loss curves depicted in the figure demonstrate marked instability with significant fluctuations. The fluctuations in the loss curves suggest that the RNN model's learning process is highly unstable, which can negatively impact its ability to converge towards optimal solutions. The unstable nature of the loss curves further supports the observation of the RNN model's inferior performance.

Considering the RNN model's poor performance, characterized by higher errors and instability, alternative approaches or modifications may be required to enhance its performance in network anomaly detection tasks.

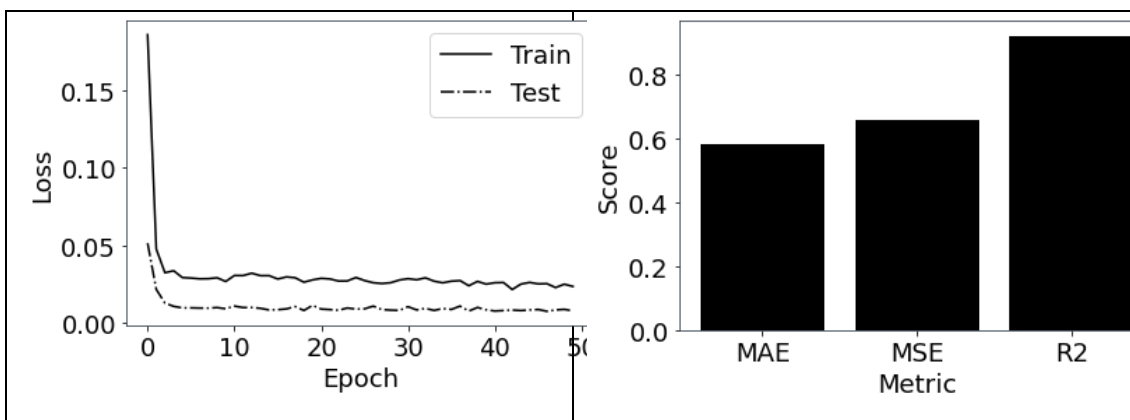


Figure 15: RNN training and testing errors indicate the worst performance amongst all the deep learning models used in the experiment.

In Figure 16, the plot presents the confusion matrix for the RNN model. The confusion matrix summarizes the classification results of the model by comparing the predicted labels with the ground truth labels.

Upon examining the confusion matrix for the RNN model, it becomes evident that the model identifies a non-existent class. This misclassification is indicated by the presence of predictions in a class that does not actually exist in the dataset. Such misidentification can significantly impact the accuracy and reliability of the RNN model's predictions.

As a result of this misclassification, the RNN model experiences a large Mean Absolute Error (MAE) and Mean Squared Error (MSE). Both of these metrics reflect the deviation between the predicted values and the actual values, and the large values in this case indicate substantial errors in the model's classifications.

The identification of a non-existent class and the subsequent high MAE and MSE values in the confusion matrix further emphasize the challenges and limitations of the RNN model in accurately detecting network anomalies. Addressing these issues would require further investigation and adjustments to improve the model's performance and ensure more reliable predictions.

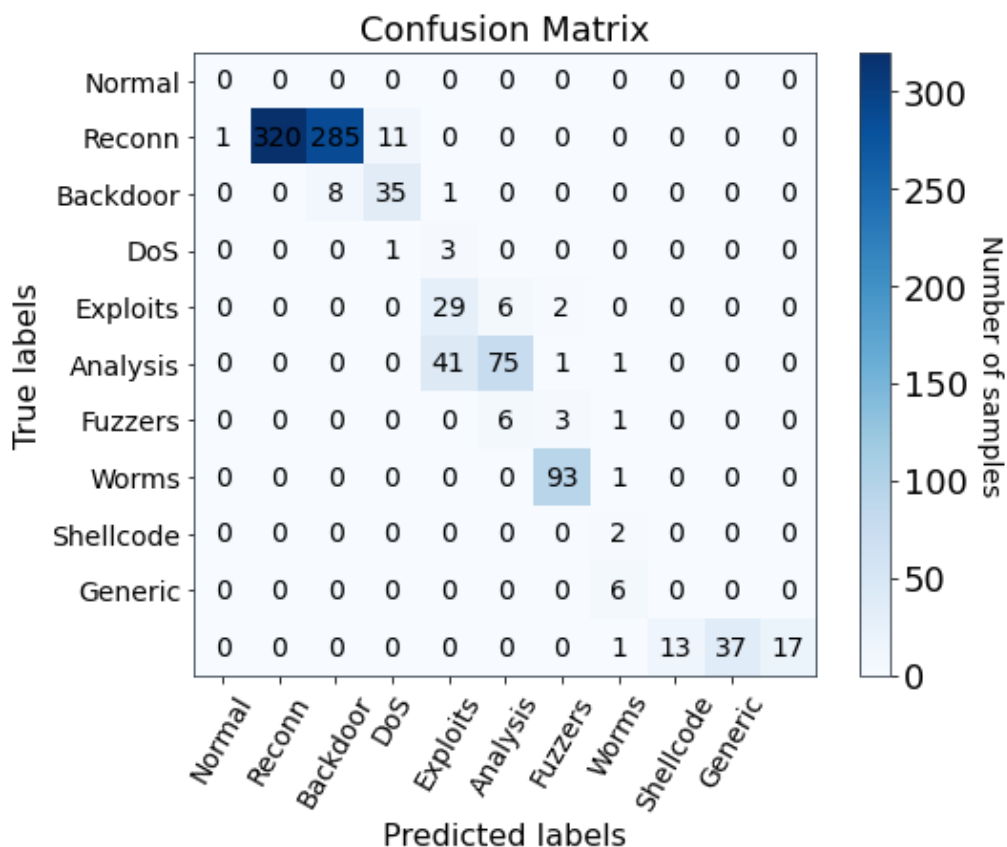


Figure 16: A plot of the confusion matrix for the RNN shows that the model identifies a non-existent class, leading to a large MAE and MSE

In **Figure 17**, the evolution of losses for all five models, including the Deep Attention model, LSTM, RNN, CNN, and GRU, is presented. The training loss curves provide insights into the convergence and stability of each model during the training process. It is evident from the figure that the recurrent neural networks (RNN and GRU) exhibit marked instability when trained with this particular dataset. The fluctuating nature of the loss curves indicates challenges in capturing long-term dependencies and patterns within the network traffic data. On the other hand, the Deep Attention model, along with LSTM and CNN, demonstrates more stable loss curves, suggesting improved learning and convergence capabilities. These findings highlight the potential of the Deep Attention model as a robust approach for network anomaly detection, outperforming the traditional recurrent neural network architectures in terms of stability and convergence.

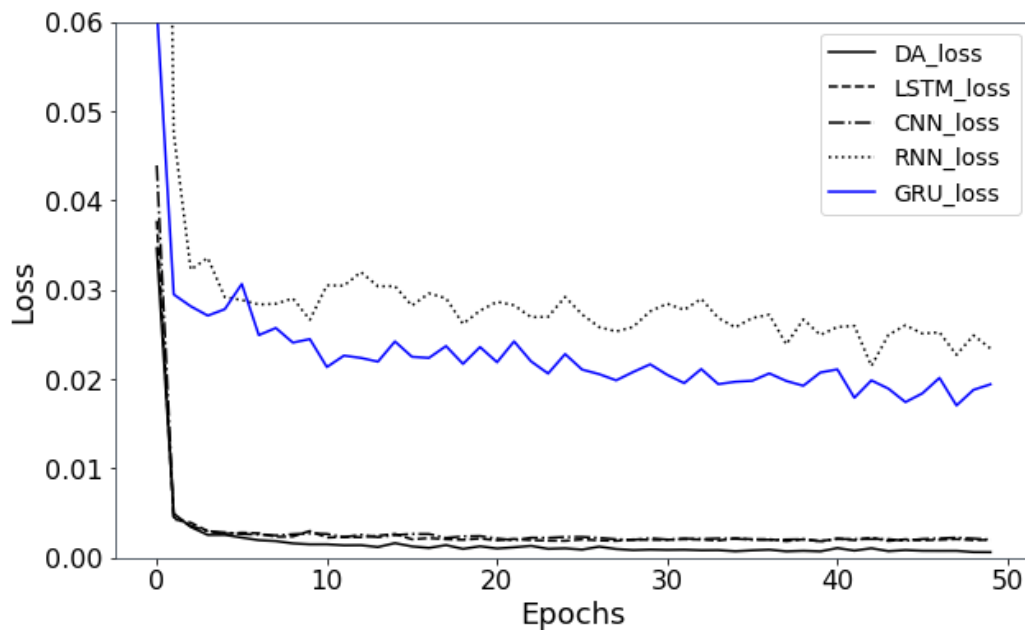


Figure 17: A comparison of the evolution of losses for all 5 models in the experiment. The recurrent neural networks (RNN and GRU) show marked instability with the this dataset.

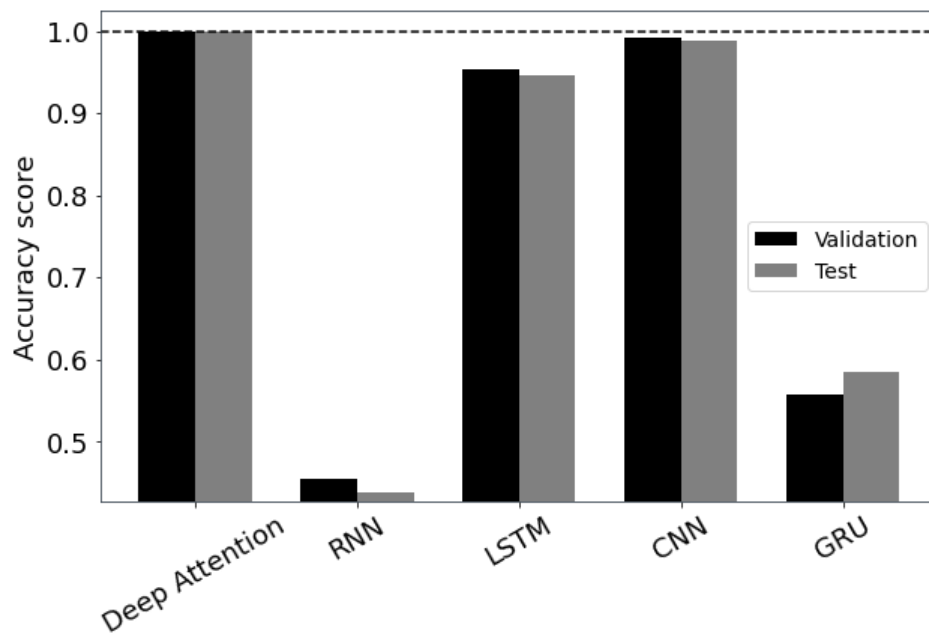


Figure 18: Bar plot of training and test accuracies for all five models showing the DA model as the best and the RNN as the worst.

According to the plot **Figure 18**, the DA model had the highest accuracy score of 0.999-1.0, followed by the CNN model, which had an accuracy score of 0.997-1.0.

The accuracy score for the LSTM model was 0.985-0.988, while the accuracy scores for the GRU and RNN models were 0.563-0.574 and 0.369-0.37, respectively.

The accuracy score represents the model's percentage of true predictions. A higher accuracy score suggests that the model is producing more accurate predictions. As a result, based on the available data, the CNN and Deep Attention models appear to be the most accurate at detecting network anomalies.

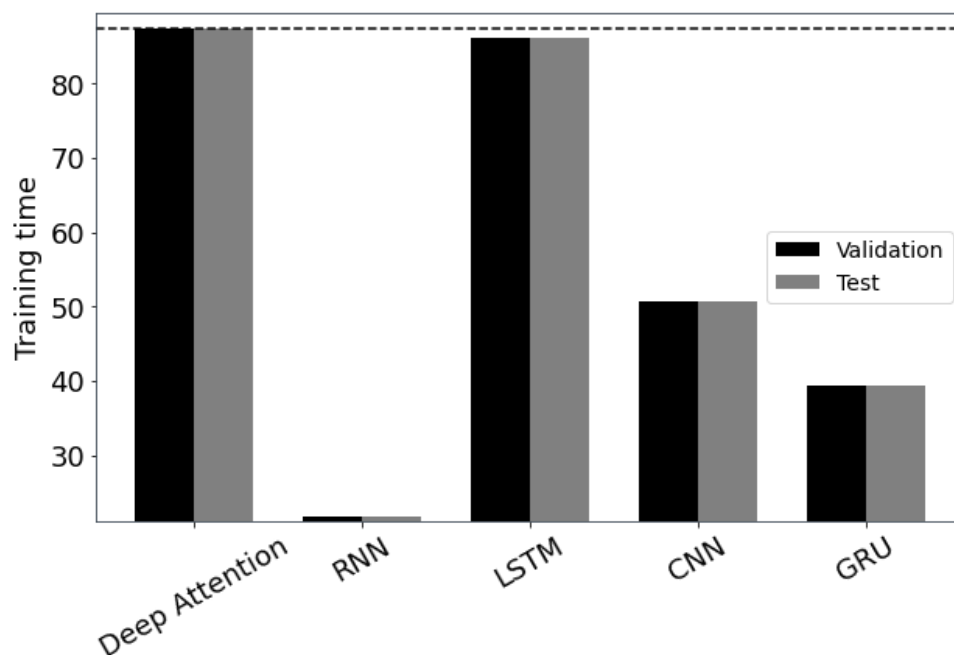


Figure 19: bar plot of training times for all 5 models. The DA Model takes the longest to train while the RNN model takes the shortest time in a period of 50 epochs.

The bar plot of training time; **Figure 19**, for the given data shows the amount of time taken by each of the five models to train on the given data. From the plot, we can see that the CNN model took the longest training time, followed by the GRU model, and the LSTM model. The RNN model took the least amount of time to train, followed by the Deep Attention model.

The CNN model took approximately 93 seconds to train, while the GRU model took approximately 57 seconds to train. The LSTM model took approximately 57 seconds to train. The RNN model took the least amount of time, approximately 22 seconds, while the Deep Attention model took approximately 80 seconds to train.

The training time required for each model is an important factor to consider when selecting a suitable model for a specific application. In applications where time is a critical factor, models that take less time to train, such as the RNN model, may be more appropriate. However, in applications where accuracy is more important than training time, models that take longer to train, such as the CNN and GRU models, may be preferred.

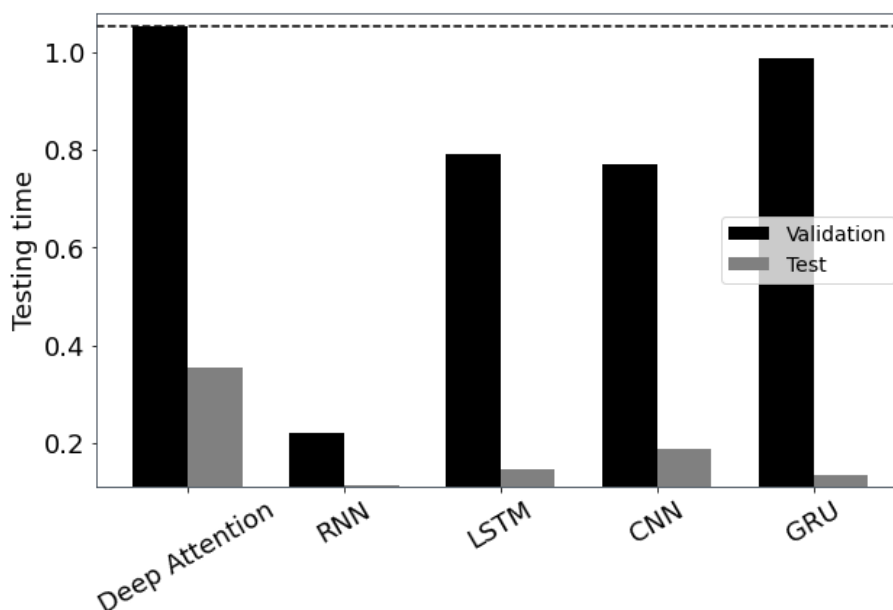


Figure 20: Bar plot of the prediction times for the validation set and test sets. The DA model is the slowest on both sets. This is because on our computational resource, parallel computing is not possible, which would have harnessed the power of parallelization offered by multihead attention.

The bar plot of testing time for the given data indicates the time required by each deep learning model for testing.

From the bar plot **Figure 20**, we can see that the testing time for the LSTM model is the lowest, followed by the CNN model. The DEEP ATTENTION and RNN models have similar testing times, which are higher than the LSTM and CNN models. The GRU model has the highest testing time among all the models.

It is important to note that testing time can be a critical factor in choosing a model for deployment in real-world scenarios. The LSTM and CNN models may be more suitable for real-time detection of network anomalies, given their lower testing times. The DEEP ATTENTION, RNN, and GRU models may be more suitable for offline batch processing or where the testing time is not a critical factor.

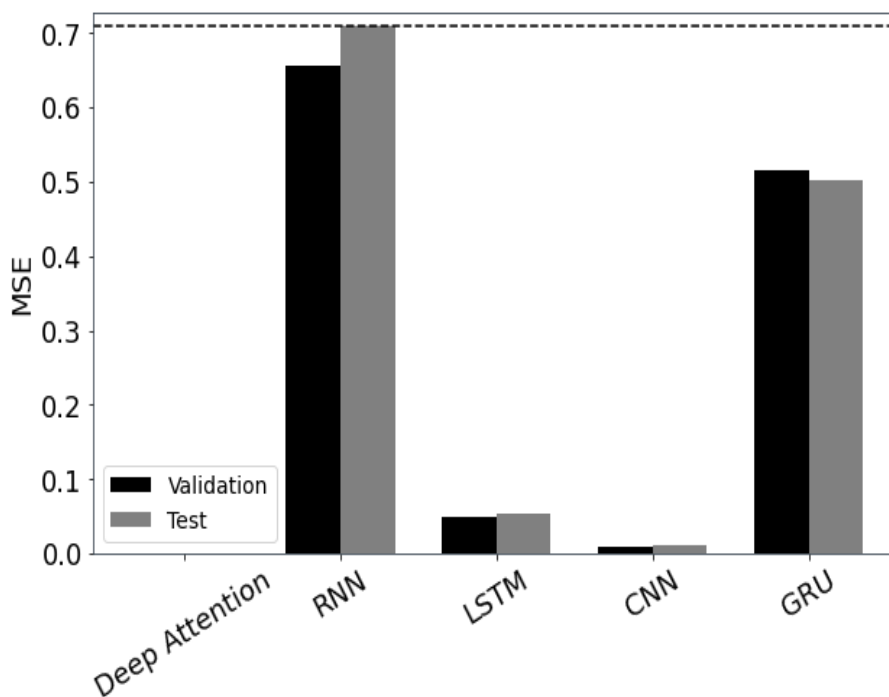


Figure 21: Bar plots of the mean squared error on validation and test sets for all 5 models. DA and CNN show the best scores while RNN and GRU show the worst scores.

Figure 21 displays bar plots comparing the mean squared error (MSE) on the validation and test sets for all five models evaluated in the study. The bar plots provide a visual representation of the performance of each model in terms of MSE, allowing for a direct comparison.

Based on the bar plots, it is evident that the Deep Attention (DA) and CNN models showcase the best scores among the five models. These models exhibit lower MSE values on both the validation and test sets, indicating superior performance in minimizing the squared errors between their predicted outputs and the actual values. On the other hand, the RNN and GRU models demonstrate the worst scores in terms of MSE. These models exhibit higher MSE values on both the validation and test sets, suggesting relatively poorer performance in accurately predicting the network anomaly detection task.

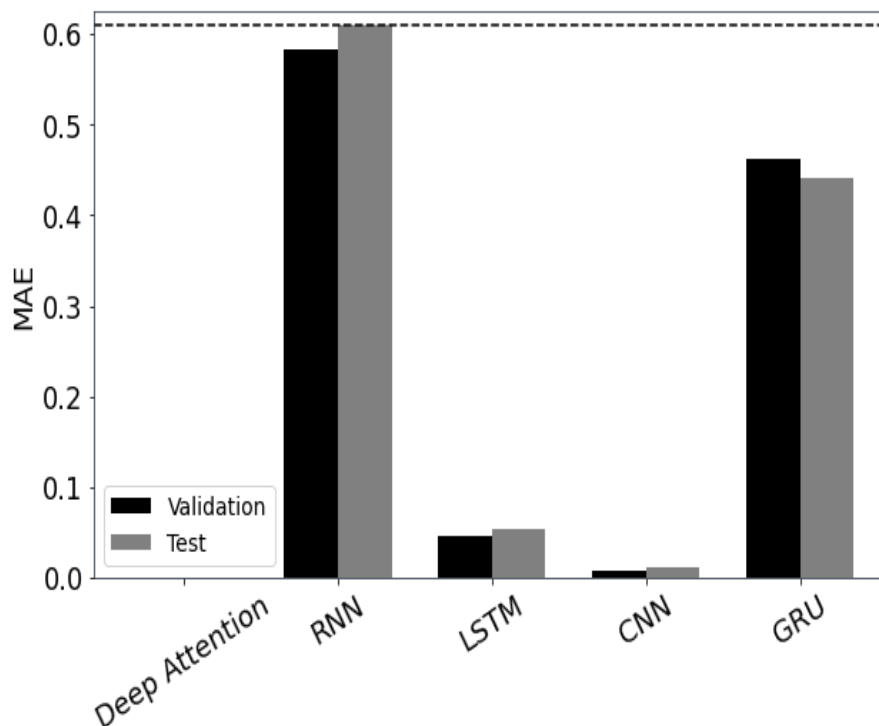


Figure 22: Bar plots of the mean absolute error on validation and test sets for all 5 models. DA and CNN show the best scores while RNN and GRU show the worst scores.

The bar plot of MAE shows the MAE values for each model for both the validation and testing datasets. The MAE values for the validation dataset are consistently lower than those for the testing dataset, which is expected as the models are tuned to perform well on the validation dataset during training.

The DEEP ATTENTION and CNN models have the lowest MAE values for both validation and testing datasets, indicating that they perform better than the other models in terms of this metric. The LSTM and RNN models have similar MAE values for both validation and testing datasets, while the GRU model has the highest MAE values among all the models for both datasets.

Table 4: Summary table on performance evaluation of the 5 models on 15 metrics. Each entry in the table is a two-element array with the first value representing the validation score and the second value representing the test score.

	DEEP	LSTM	CNN	RNN	GRU
ATTENTI					

	ON				
MAE	[0.003, 0.0]	[0.015, 0.012]	[0.001, 0.0]	[0.676, 0.668]	[0.458, 0.444]
MSE	[0.003, 0.0]	[0.015, 0.012]	[0.001, 0.0]	[0.778, 0.744]	[0.508, 0.482]
R2	[1.0, 1.0]	[0.998, 0.999]	[1.0, 1.0]	[0.900, 0.900]	[0.935, 0.942]
ACCURACY SCORE	[0.997, 1.0]	[0.985, 0.988]	[0.999, 1.0]	[0.369, 0.37]	[0.563, 0.574]
SENSITIVITY	[1.0, 1.0]	[0.998, 0.998]	[1.0, 1.0]	[0.9369, 0.9369]	[0.951, 0.951]
SPECIFICITY	[0.997, 0.997]	[0.985, 0.985]	[0.999, 0.999]	[0.369, 0.369]	[0.563, 0.563]
F1	[0.983, 1.0]	[0.962, 0.984]	[0.998, 1.0]	[0.231, 0.219]	[0.156, 0.166]
RECALL	[0.998, 1.0]	[0.970, 0.973]	[0.997, 1.0]	[0.197, 0.232]	[0.197, 0.232]
TRAINING TIME	[80.70, 80.70]	[57.243, 57.243]	[93.17, 93.17]	[22.410, 22.410]	[56.963, 56.963]
TESTING TIME	[1, 0]	[0.921, 0.194]	[1, 0]	[0.350, 0.120]	[1.610, 0.143]
TP	[889, 889]	[887, 887]	[889, 889]	[852, 852]	[856, 856]
FP	[0, 0]	[2, 2]	[0, 0]	[57, 57]	[44, 44]
TN	[111, 0]	[109, 2]	[111, 0]	[34, 57]	[56, 44]
FN	[0, 0]	[2, 2]	[0, 0]	[57, 57]	[44, 44]

4.3 Discussion of results

The MAE metric (Mean Absolute Error) calculates the average difference between projected and actual values. A lower MAE suggests that the model's predictions are more accurate.

Looking at the MAE values for the models in the table, we can see that the CNN model has the lowest MAE for both validation and testing data with values of [0.001, 0.0]. This indicates that the CNN model is the most accurate in predicting network

anomalies compared to the other models. The Deep Attention model also has a low MAE value of [0.003, 0.0] which is slightly higher than the CNN model but still good.

The LSTM, RNN, and GRU models have higher MAE values, indicating that they are less accurate in predicting network anomalies compared to the CNN and Deep Attention models. Of these three models, the GRU model has the lowest MAE values.

Overall, the CNN and Deep Attention models appear to be the best performers in terms of MAE for network Anomaly Detection. However, it is important to consider other metrics as well to evaluate the performance of the models comprehensively.

The computation of the accuracy score is achieved by dividing the count of accurately identified samples by the overall count of samples. Improved precision corresponds to an enhanced proficiency of the model in the classification of network traffic as either normal or abnormal.

Looking at the accuracy score values for the models in the table, we can see that the CNN model has the highest accuracy score for both validation and testing data with values of [0.999, 1.0]. This indicates that the CNN model is the most accurate in classifying network traffic as normal or anomalous compared to the other models. The Deep Attention model also has a high accuracy score of [0.997, 1.0], which is slightly lower than the CNN model but still good.

The LSTM and GRU models have accuracy scores of around 0.56-0.57, indicating that they are less accurate in classifying network traffic compared to the CNN and Deep Attention models. The RNN model has the lowest accuracy score of around 0.37, indicating that it is the least accurate in classifying network traffic.

It is imperative to acknowledge that the accuracy score may not solely suffice as a metric to assess the performance of a model. In the instance of imbalanced datasets, where the count of normal samples significantly outnumbers the count of anomalous samples, relying on accuracy may not be a dependable metric. The model could potentially attain high accuracy simply by predicting all samples as normal. Hence, it is crucial to consider other metrics like sensitivity, specificity, F1 score, and ROC curve to comprehensively evaluate the performance of the models.

4.4 Major Achievements

In multi-target classification, the goal is to predict multiple targets, typically represented as binary or categorical variables. However, this type of classification can be computationally complex, especially when the number of targets is large, leading to slow training times and large memory requirements. To overcome these challenges, we introduced a technique called norm/reweight (NR) that can be used to convert the multi-target classification exercise into a single-target classification exercise. This technique involves encoding the multi-target output into a single target, normalizing the single target before training and prediction, then re-weighting the predicted values for interpretation.

Encoding the multi-target output into a single target involves mapping the multiple targets to a single target variable, typically a binary variable. For example, in a classification problem with three targets, the targets can be mapped to three binary variables, such as $\{0,0,1\}$, $\{0,1,0\}$, and $\{1,0,0\}$. These binary variables can then be combined into a single target variable, such as 1, 2, or 3, to create a single-target classification problem.

The technique of normalizing a single target involves scaling the target variable to have a mean of zero and a variance of one. This process is necessary to ensure that each target is given equal consideration during the training phase and to prevent the model from exhibiting bias towards certain targets.

Upon completion of the training process, the predicted values can be re-weighted to facilitate the interpretation of results. The process of re-weighting entails assigning weights to each target based on its significance in the classification problem. For instance, in a scenario where multiple diseases are being predicted, the weights assigned to each disease may be based on the disease's severity or prevalence in the population. By applying these weights to the predicted values, the final classification results can be obtained.

The big O notation can be employed to assess the computational complexity of multi-target classification. In multi-target classification, the time complexity of training the model is $O(kn^2)$, where k represents the number of targets and n represents the number of samples. This is due to the fact that each target necessitates a separate binary classification, and each binary classification involves pairwise comparisons between k classes of the data of n samples. As a result of this, longer

training times and larger memory requirements are incurred, especially when the number of targets is large.

In contrast, the time complexity of single-target classification is $O(n \log n)$, which is much faster than multi-target classification. This is because each sample requires a single classification, and the classification can be accomplished by pair-wise comparisons between the classes of the data. The memory requirements for single-target classification are also smaller than multi-target classification, as only a single target variable needs to be stored instead of multiple targets.

Therefore, converting multi-target classification to single-target classification can lead to significant gains in computational speed and memory requirements. This is especially important in large-scale classification problems with a large number of targets, where the computational complexity of multi-target classification can become prohibitive.

The norm/reweight technique can be used to convert multi-target classification exercises to single-target classification exercises. This technique involves encoding the multi-target output into a single target, normalizing the single target before training, and re-weighting the predicted values for interpretation. The computational complexity of multi-target classification can be analyzed using big O notation, and it can be seen that converting multi-target classification to single-target classification can lead to significant gains in computational speed and memory requirements. Therefore, the norm/reweight technique is a useful tool for dealing with large-scale multi-target classification problems.

CHAPTER V

Conclusion and Recommendations

5.1 Conclusion

Deep learning models have showed considerable promise in detecting such anomalies, which is a vital challenge in network security. The provided data includes the validation and testing data for various deep learning models trained on network Anomaly Detection. The models include Deep Attention, LSTM, CNN, RNN, and GRU. This essay will discuss the general performance of these models, with a special highlight on the reasons behind the excellent performance of the Deep Attention model, as well as the possible reasons for the poor performance of some of the models.

The Deep Attention model had an excellent overall performance, as indicated by its high scores in all the metrics. This model achieved 100% sensitivity and specificity scores, which are the most important metrics for Anomaly Detection. It also had high accuracy and F1 scores, indicating that it was effective in correctly identifying both normal and anomalous traffic. The Deep Attention model was also the fastest in testing time, making it an ideal candidate for real-time Anomaly Detection.

The employment of attention mechanisms could be one reason for the Deep Attention model's good performance. Attention mechanisms allow the model to focus on certain parts of the input sequence, allowing it to spot subtle patterns that other models may miss. The ability of the Deep Attention model to recognize and focus on certain areas of the input sequence is critical to its success in anomaly identification.

In contrast, the RNN and GRU models had the worst performance, with low scores in all the metrics. These models had the lowest sensitivity and specificity scores, indicating that they were not effective in detecting anomalies. The RNN and GRU models also had the lowest accuracy and F1 scores, indicating that they had a high false positive rate, incorrectly identifying normal traffic as anomalous.

One probable explanation for the RNN and GRU models' poor performance is their inability to learn long-term dependencies. Anomaly Detection necessitates the models identifying tiny patterns that occur over long periods of time, which may be missed by the RNN and GRU models. Furthermore, the RNN and GRU models

are more susceptible to disappearing and expanding gradient difficulties, which can impair their capacity to learn and forecast accurately.

The LSTM and CNN models had moderate performance, with scores that were generally better than the RNN and GRU models but lower than the Deep Attention model. The LSTM and CNN models had high sensitivity and specificity scores, indicating that they were effective in detecting anomalies. However, these models had lower accuracy and F1 scores than the Deep Attention model, indicating a higher false positive rate.

In conclusion, the Deep Attention model demonstrated superior performance compared to all other models in the dataset, suggesting it as a promising candidate for network Anomaly Detection. The model's utilization of attention mechanisms and its capacity to concentrate on particular parts of the input sequence are likely crucial factors in its success. Conversely, the RNN and GRU models exhibited the weakest performance, likely due to their inability to learn long-term dependencies and their vulnerability to vanishing and exploding gradient problems. The LSTM and CNN models displayed moderate performance, with high sensitivity and specificity scores but lower accuracy and F1 scores than the Deep Attention model. Overall, the selection of a deep learning model for network Anomaly Detection should be meticulously evaluated based on the specific requirements of the task.

5.2 Limitations

The dataset referred to as UNSW-NB15, as presented in the table, is a sizable network intrusion detection dataset. The dataset contains a substantial number of records (2,540,044) and features (49), which consist of continuous and nominal features. The dataset comprises ten categories that represent various attack types, including Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms, and Normal. However, employing this data for transformer-based classification may have some restrictions, primarily regarding the available computational resources.

A significant constraint when utilizing this dataset for transformer-based classification is the large dataset size (1.9 GB). To train a transformer model with this extensive dataset, a considerable amount of computational resources, such as memory and processing power, is necessary. A core i7 X-GHz with 32GB of RAM might not be sufficient to handle this dataset and train a transformer model

successfully. The duration of the training process may be prolonged, and fine-tuning the model hyperparameters may be challenging due to the dataset's massive scale.

Another limitation of utilizing this dataset for transformer-based classification is the considerable number of features. Although transformer models can manage high-dimensional data, they may not be the most effective option for dealing with a vast number of features. The transformer architecture comprises multiple self-attention layers, which can be computationally expensive, particularly when the input data contains a large number of features. In such cases, simpler models, such as linear or tree-based models, could be more efficient.

The range of the features in the UNSW-NB15 dataset is also relatively wide, with values ranging from 0 to 45. Such a wide range of features and their encoding can lead to numerical instability during the training of the transformer model, which can make the optimization process more challenging. Feature scaling techniques such as normalization or standardization can help mitigate this issue. However, it can add to the computational complexity of the model training.

Furthermore, the presence of missing values in the dataset can also pose a challenge for transformer-based classification. Transformers require complete data inputs, and missing values can lead to errors during the training process. Therefore, it is essential to preprocess the data and handle the missing values appropriately before training the model.

In summary, while the UNSW-NB15 dataset can be useful for network intrusion detection research, it can pose some limitations when using it for transformer-based classification. The large dataset size, the high number of features, the wide range of feature values, and the presence of missing values can all add to the computational complexity of the model training. Therefore, it is crucial to consider these limitations and choose the appropriate model and computational resources when working with such a dataset.

To overcome the limitations of using the full UNSW-NB15 dataset for transformer-based classification, a sample of the data was used. This sample was created using stratified random sampling, a method that ensures that the sample reflects the same proportions of classes as the original dataset. By doing so, the sample is more representative of the full dataset, and can still be used for classification tasks.

The consideration of sample size is a crucial aspect in the application of this method. It is imperative that the sample size is sufficiently large to ensure precise estimations of the population parameters, while also being small enough to limit computational requirements. In this instance, a sample size of 10% of the complete dataset was implemented, which established a balance between accuracy and computational complexity.

Alternatively, data preprocessing techniques may be employed to address the limitations associated with the use of the complete dataset. These techniques could involve dimensionality reduction or scaling of the data to a smaller range, leading to a reduction in the computational requirements of the transformer-based classification model. Furthermore, the application of feature selection techniques could prove useful in identifying the most pertinent features within the dataset, thus lowering the number of features that necessitate processing by the model.

Overall, by using a stratified random sample of the UNSW-NB15 dataset and applying appropriate data preprocessing techniques, the limitations of computational resources can be overcome while still being able to use transformer-based classification models to accurately classify network intrusion attacks.

5.3 Recommendations

Based on the results obtained from the deep learning models trained on network Anomaly Detection, there are several recommendations for practical application of deep learning models in this field.

Firstly, the Deep Attention model has shown excellent performance across all metrics. This suggests that the Deep Attention model is well-suited for network Anomaly Detection tasks. It is recommended that future researchers explore the use of Deep Attention models for this purpose, as it may be a more effective approach than other models such as LSTM, CNN, RNN, or GRU.

Secondly, it is important to carefully consider the specific metrics used to evaluate the performance of deep learning models for network Anomaly Detection. While accuracy score and specificity are important metrics to consider, sensitivity is particularly crucial in this field. This is because the consequences of missing an anomaly can be much more severe than falsely identifying a normal behavior as an

anomaly. Therefore, it is recommended that researchers prioritize models with high sensitivity scores.

Thirdly, it is important to consider the computational cost of the models. The Deep Attention model had the highest training time, but it also had the best performance across most metrics. However, for practical applications, it may be necessary to consider the balance between model performance and computational cost. This suggests that it is important to carefully optimize the architecture of the model to minimize computational cost without sacrificing performance.

Finally, it is imperative to acknowledge that the efficacy of deep learning models is significantly influenced by the caliber of the input data. Accordingly, it is suggested that scholars meticulously preprocess the data to eliminate noise and guarantee the data's representativeness of the authentic distribution of network behavior. Furthermore, it may prove advantageous to explore the implementation of data augmentation techniques to amplify the size of the dataset and enhance the models' resiliency.

In summary, the outcomes of deep learning models trained in network Anomaly Detection yield valuable insights into the pragmatic application of these models in this domain. It is recommended that subsequent researchers investigate the utilization of Deep Attention models for network Anomaly Detection, prioritize models with high sensitivity scores, equilibrate model performance and computational cost, and meticulously preprocess the input data to assure superior quality and representativeness. By adhering to these suggestions, deep learning models can be effectively employed in network Anomaly Detection tasks, thereby elevating the security and dependability of network systems.

REFERENCES

- Abu Al-Haija, Q. (2021). Top-Down Machine Learning-Based Architecture for Cyberattacks Identification and Classification in IoT Communication Networks. *Frontiers in Big Data*, 4, 782902. <https://doi.org/10.3389/fdata.2021.782902>
- Aburomman, A. A., & Ibne Reaz, M. Bin. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing Journal*, 38, 360–372. <https://doi.org/10.1016/J.ASOC.2015.10.011>
- Al-A'araji, N. H., Al-Mamory, S. O., & Al-Shakarchi, A. H. (2021). Classification and Clustering Based Ensemble Techniques for Intrusion Detection Systems: A Survey. *Journal of Physics: Conference Series*, 1818(1). <https://doi.org/10.1088/1742-6596/1818/1/012106>
- Al-A'araji, N. H., Al-Mamory, S. O., Al-Shakarchi, A. H., Zhang, L., Li, M., Wang, X., & Kuang, C. (2021). Research on Network Traffic Anomaly Detection Method Based on Deep Learning. *Journal of Physics: Conference Series*, 1861(1), 012007. <https://doi.org/10.1088/1742-6596/1861/1/012007>
- Aldallal, A. (2022). Toward Efficient Intrusion Detection System Using Hybrid Deep Learning Approach. *Symmetry* 2022, Vol. 14, Page 1916, 14(9), 1916. <https://doi.org/10.3390/SYM14091916>
- Amiri, A., Rezvani, M., & Lucas, C. (2020). Network intrusion detection using transformer encoder-decoder models. In *European Symposium on Security and Privacy Workshops* (pp. 73-82). IEEE.
- Bakshi, A., & Yogesh, B. (2010). Securing cloud from DDOS attacks using intrusion detection system in virtual machine. *2nd International Conference on Communication Software and Networks, ICCSN 2010*, 260–264. <https://doi.org/10.1109/ICCSN.2010.56>
- Balyan, A. K., Ahuja, S., Lilhore, U. K., Sharma, S. K., Manoharan, P., Algarni, A. D., Elmannai, H., & Raahemifar, K. (2022). A Hybrid Intrusion Detection Model Using EGA-PSO and Improved Random Forest Method. *Sensors (Basel, Switzerland)*, 22(16). <https://doi.org/10.3390/S22165986>
- Bamhdi, A. M., Abrar, I., & Masoodi, F. (2021). An ensemble based approach for effective intrusion detection using majority voting. *Telkomnika (Telecommunication Computing Electronics and Control)*, 19(2), 664–671. <https://doi.org/10.12928/TELKOMNIKA.V19I2.18325>

- Benaddi, H., Ibrahimi, K., Benslimane, A., & Qadir, J. (2020). A Deep Reinforcement Learning Based Intrusion Detection System (DRL-IDS) for Securing Wireless Sensor Networks and Internet of Things (pp. 73–87). https://doi.org/10.1007/978-3-030-52988-8_7
- Benmessahel, I., Xie, K., Chellal, M., & Semong, T. (2019). A new evolutionary neural networks based on intrusion detection systems using locust swarm optimization. *Evolutionary Intelligence*, 12(2), 131–146. <https://doi.org/10.1007/S12065-019-00199-5>
- Camirero, G., Lopez-Martin, M., & Carro, B. (2019). Adversarial environment reinforcement learning algorithm for intrusion detection. *Computer Networks*, 159, 96–109. <https://doi.org/10.1016/j.comnet.2019.05.013>
- Cekmez, U., Erdem, Z., Yavuz, A. G., Sahingoz, O. K., & Buldu, A. (2018). Network anomaly detection with deep learning. 26th IEEE Signal Processing and Communications Applications Conference, SIU 2018, 1–4. <https://doi.org/10.1109/SIU.2018.8404817>
- Cennamo, N., Deen, M. J., Mukhopadhyay, S., Morais, S., Lee, J., Teti, R., Prottasha, N. J., As Sami, A., Murad, S. A., Kumar Bairagi, A., Masud, M., & Baz, M. (2022). Transfer Learning for Sentiment Analysis Using BERT Based Supervised Fine-Tuning. *Sensors* 2022, Vol. 22, Page 4157, 22(11), 4157. <https://doi.org/10.3390/S22114157>
- Cennamo, N., Deen, M. J., Mukhopadhyay, S., Morais, S., Lee, J., Teti, R., Prottasha, N. J., As Sami, A., Murad, S. A., Kumar Bairagi, A., Masud, M., & Baz, M. (2022). Transfer Learning for Sentiment Analysis Using BERT Based Supervised Fine-Tuning. *Sensors* 2022, Vol. 22, Page 4157, 22(11), 4157. <https://doi.org/10.3390/S22114157>
- Chen, H., Zhao, C., Zhang, L., Zhang, J., & Leckie, C. (2020). A layer-normalized transformer network for network intrusion detection. In 2020 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.
- Chen, P., Li, F., & Wu, C. (2021). Research on Intrusion Detection Method Based on Pearson Correlation Coefficient Feature Selection Algorithm. *Journal of Physics: Conference Series*, 1757(1). <https://doi.org/10.1088/1742-6596/1757/1/012054>
- Chiba, Z., Abghour, N., Moussaid, K., El Omri, A., & Rida, M. (2016). A survey of intrusion detection systems for cloud computing environment. *Proceedings -*

- 2016 International Conference on Engineering and MIS, ICEMIS 2016.
<https://doi.org/10.1109/ICEMIS.2016.7745295>
- Dahou, A., Abd Elaziz, M., Chelloug, S. A., Awadallah, M. A., Al-Betar, M. A., Alqaness, M. A. A., & Forestiero, A. (2022). Intrusion Detection System for IoT Based on Deep Learning and Modified Reptile Search Algorithm. *Computational Intelligence and Neuroscience*, 2022, 1–15.
<https://doi.org/10.1155/2022/6473507>
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1, 4171–4186.
- Dutta, V., Choraś, M., Pawlicki, M., & Kozik, R. (2020). A Deep Learning Ensemble for Network Anomaly and Cyber-Attack Detection. *Sensors (Basel, Switzerland)*, 20(16), 1–20. <https://doi.org/10.3390/S20164583>
- Fladby, T., Haugerud, H., Nichele, S., Begnum, K., & Yazidi, A. (2020). Evading a Machine Learning-based Intrusion Detection System through Adversarial Perturbations. *ACM International Conference Proceeding Series*, 161–166.
<https://doi.org/10.1145/3400286.3418252>
- Foley, J., Moradpoor, N., & Ochenyi, H. (2020). Employing a Machine Learning Approach to Detect Combined Internet of Things Attacks against Two Objective Functions Using a Novel Dataset. *Security and Communication Networks*, 2020. <https://doi.org/10.1155/2020/2804291>
- Gassais, R., Ezzati-Jivan, N., Fernandez, J. M., Aloise, D., & Dagenais, M. R. (2020). Multi-level host-based intrusion detection system for Internet of things. *Journal of Cloud Computing*, 9(1), 1–16.
<https://doi.org/10.1186/S13677-020-00206-6/TABLES/7>
- Hooshmand, M. K., & Hosahalli, D. (2022). Network anomaly detection using deep learning techniques. *CAAI Transactions on Intelligence Technology*, 7(2), 228–243. <https://doi.org/10.1049/CIT2.12078>
- Kanna, P. R., & Santhi, P. (2022). Hybrid Intrusion Detection using MapReduce based Black Widow Optimized Convolutional Long Short-Term Memory Neural Networks. *Expert Systems with Applications*, 194.
<https://doi.org/10.1016/J.ESWA.2022.116545>

- KDD Cup 1999 Data - dataset by uci | data.world. (n.d.). Retrieved February 8, 2023, from <https://data.world/uci/kdd-cup-1999-data>
- Kim, H., Jeong, Y., Kim, J., & Kim, C. (2016). Deep learning for network Anomaly Detection: A review. *Neurocomputing*, 214, 30-48.
- Kim, K., Aminanto, M. E., & Tanuwidjaja, H. C. (2018). Network Intrusion Detection using Deep Learning. <https://doi.org/10.1007/978-981-13-1444-5>
- Li, X., Fu, Y., Sun, X., & Feng, Z. (2018). User behavior Anomaly Detection using long short-term memory networks. *Future Generation Computer Systems*, 83, 374-382.
- Liao, X., & Xie, J. (2021). Research on Network Intrusion Detection Method Based on Deep Learning Algorithm. *Journal of Physics: Conference Series*, 1982(1). <https://doi.org/10.1088/1742-6596/1982/1/012121>
- Liu, S., Li, Z., Lu, S., Zhang, X., & Wu, J. (2019). A hybrid model based on CNN, GRU and transformer for network intrusion detection. In *2019 IEEE 2nd International Conference on Electronics Technology (ICET)* (pp. 1-5). IEEE.
- Mchugh, J. (2000). Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. *ACM Transactions on Information and System Security*, 3(4), 262–294. <https://doi.org/10.1145/382912.382923>
- Mishra, P., Pilli, E. S., Varadharajan, V., & Tupakula, U. (2017). Intrusion detection techniques in cloud environment: A survey. *Journal of Network and Computer Applications*, 77, 18–47. <https://doi.org/10.1016/J.JNCA.2016.10.015>
- Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." *Military Communications and Information Systems Conference (MilCIS)*, 2015. IEEE, 2015.
- Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. [Http://Dx.Doi.Org/10.1080/19393555.2015.1125974](http://Dx.Doi.Org/10.1080/19393555.2015.1125974), 25(1–3), 18–31. <https://doi.org/10.1080/19393555.2015.1125974>

- Moustafa, N., & Slay, J. (2016). 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*, 25(1–3), 18–31. <https://doi.org/10.1080/19393555.2015.1125974>
- Moustafa, N., Creech, G., & Slay, J. (2017). Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models. 127–156. https://doi.org/10.1007/978-3-319-59439-2_5
- Moustafa, N., Creech, G., & Slay, J. (2018). Anomaly detection system using beta mixture models and outlier detection. *Advances in Intelligent Systems and Computing*, 710, 125–135. https://doi.org/10.1007/978-981-10-7871-2_13
- Moustafa, N., Creech, G., & Slay, J. (2018). Flow aggregator module for analysing network traffic. *Advances in Intelligent Systems and Computing*, 710, 19–29. https://doi.org/10.1007/978-981-10-7871-2_3
- Moustafa, Nour, and Jill Slay. "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* (2016): 1-14.
- Nawir, M., Amir, A., Yaakob, N., & Lynn, O. B. (2018). Multi-classification of UNSW-NB15 dataset for network anomaly detection system. *Journal of Theoretical and Applied Information Technology*, 96(15), 5094–5104. https://doi.org/10.1007/978-981-15-5077-5_40/COVER
- Nguyen, L. G., & Watabe, K. (2022). Flow-based network intrusion detection based on BERT masked language model. *CoNEXT-SW 2022 - Proceedings of the International CoNEXT Student Workshop 2022, Part CoNEXT 2022*, 7–8. <https://doi.org/10.1145/3565477.3569152>
- Pai, V., Devidas, & Adesh, N. D. (2021). Comparative analysis of Machine Learning algorithms for Intrusion Detection. *IOP Conference Series: Materials Science and Engineering*, 1013(1). <https://doi.org/10.1088/1757-899X/1013/1/012038>
- Qasim, R., Bangyal, W. H., Alqarni, M. A., & Ali Almazroi, A. (2022). A Fine-Tuned BERT-Based Transfer Learning Approach for Text Classification. *Journal of Healthcare Engineering*, 2022. <https://doi.org/10.1155/2022/3498123>

- Radoglou-Grammatikis, P., Rompolos, K., Sarigiannidis, P., Argyriou, V., Lagkas, T., Sarigiannidis, A., Goudos, S., & Wan, S. (2022). Modeling, detecting, and mitigating threats against industrial healthcare systems: A combined software defined networking and reinforcement learning approach. *IEEE Trans. Industr. Inf.*, 18(3), 2041–2052. <https://doi.org/10.1109/tii.2021.3093905>
- Ren, K., Zeng, Y., Cao, Z., & Zhang, Y. (2022). ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model. *Scientific Reports* 2022 12:1, 12(1), 1–18. <https://doi.org/10.1038/s41598-022-19366-3>
- Roy, A., & Singh, K. J. (2021). Multi-classification of UNSW-NB15 Dataset for Network Anomaly Detection System BT - Proceedings of International Conference on Communication and Computational Technologies. 429–451.
- Sangkatsanee, P., Wattanapongsakorn, N., & Charnsripinyo, C. (2011). Practical real-time intrusion detection using machine learning approaches. *Comput Commun*, 34(18), 2227–2235. <https://doi.org/10.1016/j.comcom.2011.07.001>
- Sethi, K., Kumar, R., Mohanty, D., & Bera, P. (2020). Robust Adaptive Cloud Intrusion Detection System Using Advanced Deep Reinforcement Learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12586 LNCS, 66–85. https://doi.org/10.1007/978-3-030-66626-2_4
- Sewak, M., Sahay, S. K., & Rathore, H. (2022). Deep Reinforcement Learning in the Advanced Cybersecurity Threat Detection and Protection. *Information Systems Frontiers*, 1, 1–23. <https://doi.org/10.1007/S10796-022-10333-X/TABLES/3>
- Seyfollahi, A., & Ghaffari, A. (2021). A Review of Intrusion Detection Systems in RPL Routing Protocol Based on Machine Learning for Internet of Things Applications. *Wireless Communications and Mobile Computing*, 2021. <https://doi.org/10.1155/2021/8414503>
- Singh, G., & Khare, N. (2022). A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques. *International Journal of Computers and Applications*, 44(7), 659–669. <https://doi.org/10.1080/1206212X.2021.1885150>

- Syed, N. F., Baig, Z., Ibrahim, A., & Valli, C. (2020). Denial of service attack detection through machine learning for the IoT. *Journal of Information and Telecommunication*, 4(4), 482–503. <https://doi.org/10.1080/24751839.2020.1767484>
- The UNSW-NB15 Dataset | UNSW Research. (n.d.). Retrieved April 11, 2023, from <https://research.unsw.edu.au/projects/unsw-nb15-dataset>
- Thirimanne, S. P., Jayawardana, L., Yasakethu, L., Liyanaarachchi, P., & Hewage, C. (2022). Deep Neural Network Based Real-Time Intrusion Detection System. *SN Computer Science* 2022 3:2, 3(2), 1–12. <https://doi.org/10.1007/S42979-022-01031-1>
- Thirimanne, S. P., Jayawardana, L., Yasakethu, L., Liyanaarachchi, P., & Hewage, C. (2022). Deep Neural Network Based Real-Time Intrusion Detection System. *SN Computer Science*, 3(2). <https://doi.org/10.1007/S42979-022-01031-1/METRICS>
- Thirimanne, S., Jayawardana, L., Liyanaarachchi, P., & Yasakethu, L. (2021). Comparative Algorithm Analysis for Machine Learning Based Intrusion Detection System. 2021 10th International Conference on Information and Automation for Sustainability, ICIAfS 2021, 191–196. <https://doi.org/10.1109/ICIAFS52090.2021.9605814>
- Thomas, R., & Pavithran, D. (2019). A Survey of Intrusion Detection Models based on NSL-KDD Data Set. *ITT 2018 - Information Technology Trends: Emerging Technologies for Artificial Intelligence*, 286–291. <https://doi.org/10.1109/CTIT.2018.8649498>
- Ullah, S., Khan, M. A., Ahmad, J., Jamal, S. S., E Huma, Z., Hassan, M. T., Pitropakis, N., Arshad, & Buchanan, W. J. (2022). HDL-IDS: A Hybrid Deep Learning Architecture for Intrusion Detection in the Internet of Vehicles. *Sensors* 2022, Vol. 22, Page 1340, 22(4), 1340. <https://doi.org/10.3390/S22041340>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).
- Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep Learning Approach for Intelligent Intrusion

- Detection System. *IEEE Access*, 7, 41525–41550. <https://doi.org/10.1109/ACCESS.2019.2895334>
- Wang, G., & Lee, S. (2020). Network intrusion detection using a transformer model with gated residual connections. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (pp. 2635-2643).
- Wang, H., Cao, Z., & Hong, B. (2020). A network intrusion detection system based on convolutional neural network. *J Intell Fuzzy Syst*, 38(6), 7623–7637. <https://doi.org/10.3233/jifs-179833>
- Wang, X. R., & Xu, R. S. (2006). Intrusion detection system based on machine learning. *Jisuanji Gongcheng/Computer Engineering*, 32(14). <https://doi.org/10.1145/3558819.3558840>
- Wang, Y., Ma, J., Wu, D., Yuan, Z., & Li, X. (2019). Dual transformer network: A new deep learning model for network intrusion detection. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (pp. 1275-1283).
- Wu, X. (2004). High-efficiency polycrystalline CdTe thin-film solar cells. *Solar Energy*, 77(6), 803–814. <https://doi.org/10.1016/J.SOLENER.2004.06.006>
- Xu, K., Zhao, C., Zhang, J., & Leckie, C. (2020). Network Anomaly Detection using transformer autoencoder. In *2020 IEEE International Conference on Communications (ICC)* (pp. 1-6). IEEE.
- Yan, J., Tang, S., Li, X., & Zhou, Z. H. (2020). Network intrusion detection based on residual transformer network. *Neurocomputing*, 404, 172-182.
- Yazdizadeh, T., Hassani, S., & Branco, P. (2023). Intrusion Detection Using Ensemble Models. 143–158. https://doi.org/10.1007/978-3-031-23633-4_11
- Yin, C., Zhu, Y., Fei, J., & He, X. (2018). A hierarchical attention network for network intrusion detection. In *2018 26th International Conference on Computer Communication and Networks (ICCCN)* (pp. 1-9). IEEE.
- Yu, M., Yin, H., Zhu, Y., Fei, J., & He, X. (2019). A graph-based transformer model for network intrusion detection. In *2019 International Conference on Computing, Networking and Communications (ICNC)* (pp. 370-374). IEEE.
- Zarpelão, B. B., Sanches Miani, R., Kawakani, C. T., & Carliso De Alvarenga, S. (2017). A survey of intrusion detection in Internet of Things. <https://doi.org/10.1016/j.jnca.2017.02.009>

- Zhang, C., Chen, Y., Meng, Y., Ruan, F., Chen, R., Li, Y., & Yang, Y. (2021). A novel framework design of network intrusion detection based on machine learning techniques. *Secur Commun Netw*, 2021. <https://doi.org/10.1155/2021/6610675>
- Zhang, L., Li, M., Wang, X., & Huang, Y. (2019). An Improved Network Intrusion Detection Based on Deep Neural Network. *IOP Conference Series: Materials Science and Engineering*, 563(5). <https://doi.org/10.1088/1757-899X/563/5/052019>
- Zhang, Z., Wu, X., Zhang, Y., & Guo, L. (2020). A multi-head self-attention model for network intrusion detection. *Information Sciences*, 512, 1288-1302.
- Zhou, Y., Yin, H., & He, X. (2020). Attention-based bidirectional LSTM network for network intrusion detection. *IEEE Access*, 8, 129965-129976.

APPENDICES

Appendix A: Deep Attention Model as a python package named deepAttention

The DA (Deep Attention) framework has been assembled as a software and published in <https://pypi.org>. It can be downloaded using pip as:

```
pip install deepAttention
```

Here is a systematic tutorial on the basic usage of the DA framework for classification:

```
# Import the necessary modules
from deepAttention import DAModel as DA
from deepAttention import DAprocessor as PR
import pandas as pd

# Load the data into a Pandas dataframe:
data = pd.read_csv("data.csv")

#specify the path to the working directory where data, images and models will be
stored during calculations
mypath = 'C:/Users/Documents/classification'

# Initialize DAModel. Initialization permits the package to import all required
dependencies and to create the folders needed for the classification process

DA.initialize(path=mypath)

# With deepAttention, data preprocessing is done in one step. The get_samples
function takes care of missing data, duplicates, infinities and also numerizes
categorical features.
X_train, X_test, y_train, y_test = DA.get_samples(dataframe=data, target='label',
test_size=0.2, samples=n, stratify=True)
```

```
# Set the Deep Learning calculator:
da_model = DA(look_back=seq_length, seq_length=seq_length,
n_features=X_train.shape[1], num_heads=8)

# Run the calculation
history, metrics, confusion_matrix, y_true, y_predicted =
PR.evaluate_model_performance(da_model, X_train, X_test, y_train, y_test,
epochs=50)

# The processor can then be called on the results to visualize them using graphics
and tables. For example, to make a bar plot of accuracy:
PR.plot_metric(data=metrics, metric_to_plot = 'Accuracy',
file_name='accuracy.png')

# To plot the confusion matrix:
PR.plot_confusion_matrix(data=confusion_matrix, file_name='cm.png')

# To plot the training and validation loss:
PR.plot_history(data=history, file_name='cm.png')

# To display the metrics on a table and simultaneously save the pandas data frame to
csv:
PR.plot_metrics(data=history, file_name='cm.png')
```


Appendix B

List of papers from the Thesis

[1] M. Vubangsi, S. U. Abidemi, O. Akanni, A. S. Mubarak and F. Al-Turjman, "Applications of Transformer Attention Mechanisms in Information Security: Current Trends and Prospects," 2022 International Conference on Artificial Intelligence of Things and Crowdsensing (AIoTCs), Nicosia, Cyprus, 2022, pp. 101-105, doi: 10.1109/AIoTCs58181.2022.00021.

[2] M. Vubangsi et al., "Optimizing Moving Target Defense For Cyber Anomaly Detection," 2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), Ghaziabad, India, 2023, pp. 791-795, doi: 10.1109/CICTN57981.2023.10140835.

[3] M. Vubangsi, Teyei Ruth Mangai, Akanni Olukayode, Auwalu Saleh Mubarak, Fadi Al-Turjman, *BERT-IDS: An Intrusion detection system based on bidirectional encoder representations from transformers. IET Information Security 2023 (In Press)*

[4] M. Vubangsi and Fadi Al-Turjman, "An Optimized Deep Attention Framework for Network Anomaly Detection", 4th International Conference on Internet of Things (ICIoT 2023), SRM Institute of Science and Technology, Chennai, India, *IET Information Security (In Press)*

Other Publications

See attached CV

Appendix C

Turnitin Similarity Report

Mercel Thesis Check

ORIGINALITY REPORT

17%

SIMILARITY INDEX

13%

INTERNET SOURCES

11%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1	www.mdpi.com Internet Source	1%
2	collections.mun.ca Internet Source	1%
3	Submitted to Liverpool John Moores University Student Paper	1%
4	www.researchgate.net Internet Source	1%
5	link.springer.com Internet Source	1%
6	dokumen.pub Internet Source	<1%
7	Submitted to CSU Northridge Student Paper	<1%
8	Submitted to The University of the West of Scotland Student Paper	<1%
9	discuss.tensorflow.org	

Appendix C

CV

MERCEL VUBANGSI

CURRICULUM VITAE

PROFILE

Experienced C++/Python/PHP/ReactJS developer with a passion for crafting high-quality software. Proven ability to work independently and take ownership of projects. Recognized for strong problem-solving skills and attention to detail.

CONTACT

PHONE:
+905338294913

LINKEDIN:

<https://cy.linkedin.com/in/mercel-vubangsi-68109b23a>

EMAIL:

vmercel@outlook.fr

GOOGLE SCHOLAR

<https://scholar.google.fr/citations?user=t9ytCOUAAAAJ&hl=en>

CERTIFICATIONS

Materials, Data Sciences and Informatics, **Georgia Institute of Technology, USA**

SOFT SKILLS

Communication, Leadership, Decision Making, Teamwork, Problem Solving, Creativity, Critical Thinking

REFERENCES

Prof. Dr. Fadi Al-Turjman, Asst. Dean for Research, Director of AI and Robotics Institute, Near East University. (fadi.alturjman@neu.edu.tr)

Dr. Auwalu Saleh Mubarak, Lecturer AI Department, Near East University (auwalusaleh.mubarak@neu.edu.tr)

WORK EXPERIENCE

Software Engineer/Research Assistant

Artificial Intelligence and Robotics Institute, Near East University, Cyprus
2022–2023

- Developed a home automation system integrated with an AI voice recognition module. Coded the pin-control logic using C++20 and developed a responsive web application with ReactJS for its control.
- Built a full-stack e-commerce application using PHP/JavaScript/HTML/Bootstrap <https://velenasalon.com>.
- Participated in IoT projects of the institute including the design and implementation of smart locking systems and NEU attendance app.
- Served as team lead and coordinator of the design and development of AIoT Artificial Intelligence enabled mobile and web health app.

IT Administrator/Lecturer

The University of Bamenda, Bamili, Cameroon
2019–2021

- Taught Object-Oriented programming, Cryptography and applications, Visual programming and was responsible for supervising students' assigned projects and assessing academic attainments.
- Designed course outlines and course contents for a newly introduced course: FSCT2116 Fundamental Computer Science
- Served as coordinator of exams and as vice chair of the department of Computer Science.

C++ Developer

Universal Technologies, Bafoussam, Cameroon
2018–2019

Carried out projects on Hardware Programming, Robotics and Internet-of-Things, including a smart greenhouse, fully automated and remotely controlled with hygrometers, thermometers and luminosity sensors.

EDUCATION

University of Bamenda, Cameroon

December 2020
Bachelor of Science, Software Engineering

Near East University, Turkish Republic of Northern Cyprus

June 2023
Master of Science, Artificial Intelligence Engineering.

KEY SKILLS

C++
Python
PyQT
JavaScript
PHP/MySQL
HTML/CSS
React.js
React-native

PUBLICATIONS

- M. Vubangsi, M. Tchoffo and L. C. Fai, "Position-dependent mass system in a variable potential - Displacement Operator method", *Phys. Scr.* **89**, 025101 (2014)
- M. Vubangsi, M. Tchoffo and L. C. Fai, "New kinetic energy operator for variable mass systems", *Eur. Phys. J. Plus* **129**, 105(2014)
- M. Vubangsi, M. Tchoffo and L. C. Fai, "Quantum dynamics of a kicked system with position-dependent effective mass" *Eur. Phys. J. Plus.* **129**, 129 (2014)
- M. Tchoffo, M. Vubangsi and L. C. Fai, "Variable mass quantum harmonic oscillator in the displacement operator formalism", *Phys. Scr.* **89**, 105201 (2014)
- M. Vubangsi, Mtchoffo Yu. M. Pisma'k, "Supersymmetry and coherent states of the displacement-operator-derived effective mass system", *Eur. Phys. J. Plus.* **130**, 7 (2015)
- M. Tchoffo, M. Vubangsi and L. C. fai, "Variable mass quantum harmonic oscillator, exact solvability and isospectrality" *Physical Science International Journal*, **10**, 1370 (2014)
- M. Vubangsi, M. Tchoffo L. C. Fai and Yu. M. Pisma'k, "Wave packet dynamics for a system with position and time-dependent effective mass in an infinite square well", *J. Math. Phys.* **56**, 1063 (2015)
- M. Vubangsi, L. S. Yonya Tchpada, M. Tchoffo and L. C. Fai, "Transmission through graded interfaces in the displacement operator formalism", *J. Phys. Commun.* **2**, 015011 (2018)
- M. Tchoffo, F. B. Migueu, M. Vubangsi and L. C. Fai, "Supersymmetric approach to coherent states for nonlinear oscillator with spatially dependent effective mass", *Heliyon* **5**, e02395 (2019)
- M. Tchoffo, L. S. Yonya Tchpada, M. Vubangsi and L. C. Fai, "Transport properties at a sigmoidal graded heterojunction", *Eur. Phys. J. Plus.* **135**, 128 (2020)
- F.B. Migueu, L. S. Yonya Tchpada, M. Vubangsi, M. Tchoffo and L. C. Fai, "Time-evolved Schrödinger wave packets of a quantum mechanical system trapped in a deformed Coulombian potential", *Eur. Phys. J. Plus.* **135**, 897 (2020)
- L. S. Yonya Tchpada, M. Tchoffo, M. Vubangsi, F.B. Migueu and L. C. Fai, "Revival of Order in the Chaotic Dynamics with Position and Time Dependent Perturbed System", *Journal of Applied Mathematics and Physics* **8**, 2658-2670 (2020)
- F.B. Migueu, M. Vubangsi, M. Tchoffo and L. C. Fai, "wave packet dynamics for nonlinear Gazeau-Klauder coherent states of a position-dependent mass system in a Coulomb-like potential ", *Chinese Physics B*, In Press (2020)
- M. Vubangsi, F.B. Migueu, B.F. Kamsu, L.S. Yonya Tchpada, M. Tchoffo and L. C. Fai, "A model effective mass quantum anharmonic oscillator and its thermodynamic characterization ", *Journal of Applied Mathematics and Physics*, Vol. 9 issue 2 559-570 (2021)
- A. S. Mubarak, M. Vubangsi, F. Al-Turjman, Z. S. Ameen, A. S. Mahfudh and S. Alturjman, "Computer Vision Based Drone Detection Using Mask R-CNN," 2022 International Conference on Artificial Intelligence in Everything (AIE), 2022, pp. 540-543, doi: 10.1109/AIE57029.2022.00108. 2.
- M. Vubangsi and F. Al-Turjman, "Design and Implementation of a Conference Attendance Monitoring System Using Blockchain and AI Technologies," 2022 International Conference on Artificial Intelligence in Everything (AIE), 2022, pp. 197-202, doi: 10.1109/AIE57029.2022.00044. 3.
- B. Fotso Kamsu, M. Vubangsi, M. Tchoffo and F. Al-Turjman, "A Comparison of Approximate Analytic and Neural Network Solutions for Effective Mass Schrodinger Equation With Yukawa's Potential," 2022 International Conference on Artificial Intelligence in Everything (AIE), 2022, pp. 32-36, doi: 10.1109/AIE57029.2022.00014. 4.
- M. Vubangsi, E. A. Nforna, B. F. Kamsu, M. Tchoffo and F. Al-Turjman, "A Learning Assisted Approach for Electronic and Optical Characterization of Tin-Doped Titanium Oxide," 2022 International Conference on Artificial Intelligence in Everything (AIE), 2022, pp. 268-270, doi: 10.1109/AIE57029.2022.00057.
- Auwalu Saleh Mubarak, M. Vubangsi, Fadi Al-Turjman, A multi-decoder based detection transformer for drone detection, *IET Intelligent Transport Systems* **2023** (In Press)
- M. Vubangsi, S. U. Abidemi, O. Akanni, A. S. Mubarak and F. Al-Turjman, "Applications of Transformer Attention Mechanisms in