ATTACK DETECTION AND ANALYSIS WITH DEEP LEARNING IN CLOUD COMPUTING

2024
MASTER THESIS
COMPUTER ENGINEERING

Hayder Abdulameer Yousif AL-IESSA

Thesis Advisor
Assist. Prof. Dr. İsa AVCI

ATTACK DETECTION AND ANALYSIS WITH DEEP LEARNING IN CLOUD COMPUTING

Hayder Abdulameer Yousif AL-IESSA

Thesis Advisor
Assist. Prof. Dr. İsa AVCI

T.C.
Karabuk University
Institute of Graduate Programs
Department of Computer Engineering
Prepared as
Master Thesis

KARABUK
April 2024

I certify that in my opinion the thesis submitted by Hayder Abdulameer Yousif AL-IESSA titled “ATTACK DETECTION AND ANALYSIS WITH DEEP LEARNING IN CLOUD COMPUTING” is fully adequate in scope and quality as a thesis for the degree of Master of Science.

Assist. Prof. Dr. İsa AVCI ....

Thesis Advisor, Department of Computer Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science thesis. April 25, 2024

Examining Committee Members (Institutions)	Signature
Chairman : Assoc.Prof.Dr.Muhammed Ali AYDIN (ICU)	....
Member : Assist. Prof. Dr. Yasin ORTAKÇI (KBU)	....
Member : Assist. Prof. Dr. İsa AVCI (KBU)	....

The degree of Master of Science by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Assoc. Prof. Dr. Zeynep ÖZCAN ....

Director of the Institute of Graduate Programs

“I declare that all the information within this thesis has been gathered and presented in accordance with academic regulations and ethical principles and I have according to the requirements of these regulations and principles cited all those which do not originate in this work as well.”

Hayder Abdulameer Yousif AL-IESSA

ABSTRACT

M. Sc. Thesis

ATTACK DETECTION AND ANALYSIS WITH DEEP LEARNING IN CLOUD COMPUTING

Hayder Abdulameer Yousif AL-IESSA

Karabuk University
Institute of Graduate Programs
Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Dr. İsa AVCI

April 2024, 98 pages

In the current landscape, where digital networks are more intertwined than ever, ensuring the security of these networks against cyber threats has emerged as a paramount challenge. Traditional cybersecurity strategies, which once served as robust defenses, are now finding it increasingly difficult to match the pace and sophistication of contemporary cyber-attacks. This dynamic shift calls for a reevaluation of our approach to safeguarding digital infrastructures.

This study introduces an advanced solution to the cybersecurity conundrum by leveraging the potential of deep learning technologies for the purpose of automatic attack prediction. Our research specifically zeros in on the development of predictive models that are meticulously trained to identify and classify Denial of Service (DoS) attacks—a particularly prevalent and disruptive category of cyber-attacks. DoS attacks, characterized by their ability to overwhelm and incapacitate digital services, pose a significant threat to the integrity and availability of digital resources.

The cornerstone of our approach lies in the application of deep learning algorithms, renowned for their ability to dissect and learn from large datasets. These algorithms are employed to uncover subtle patterns and anomalies that are indicative of DoS attacks, thereby facilitating their early detection. The early identification of such attacks is crucial, enabling the implementation of proactive measures to mitigate their impact.

Our methodology involves rigorous experimentation and thorough evaluation of the developed models. Through this process, our study showcases encouraging outcomes, with the deep learning-based models attaining an accuracy rate of 82.76%. This achievement underscores the effectiveness and potential of deep learning techniques in enhancing the security of networked systems and combatting cyber threats in today's intricate digital sphere.

The findings of our research signify a significant leap forward in the domain of cybersecurity. By demonstrating the viability of using deep learning to predict and counteract cyber-attacks, our study makes a substantial contribution to the evolution of cybersecurity tactics. Organizations equipped with these advanced predictive tools are better positioned to navigate the increasingly complex and ever-changing threat landscape. Thus, our work not only sheds light on innovative cybersecurity solutions but also sets the stage for future research and development in the field, aiming at a safer and more secure digital environment.

Key Words : Network intrusion, Malicious activities, Data exchange, Cloud infrastructure, Deep learning technology, Denial of service (DoS) attacks, Attack prediction.

Science Code : 92432

ÖZET

Yüksek Lisans Tezi

BULUT BİLGİSAYARINDA DERİN ÖĞRENME İLE SALDIRI TESPİT VE ANALİZİ

Hayder Abdulameer Yousif AL-IESSA

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi İsa AVCI

Nisan 2024, 98 sayfa

Bilgisayar ve internet ağlarının genişlemesi, ağ ihlali olasılığının artmasına neden olmuş, böylece bilgisayar korsanlığı ve diğer kötü amaçlı faaliyetler potansiyelini artırmıştır. İnternet, yaygın olarak erişilebilir bir kamu ağı olarak, genellikle sunucular veya bulut platformları aracılığıyla kolaylaştırılan çeşitli varlıklar arasında veri alışverişini gerektirir. Ancak, bu verilerin güvenliği, olası saldırıları önlemek için dağıtılan bulut altyapısının, sunucuların ve ilişkili güvenlik duvarlarının sağlamlığına bağlıdır.

Ne yazık ki, yazılım teknolojisindeki sürekli gelişmeler, mevcut güvenlik duvarlarını ezici saldırı etkinliklerine karşı giderek daha savunmasız hale getirmiştir. Bu teknolojik gelişmeler, geleneksel güvenlik duvarı çözümlerinin etkinliğini geride bırakarak, modern siber tehditlerin sofistike ve kalıcı doğasını ele almada yetersiz kalmıştır. Bu acil endişeye yanıt olarak, çalışmamız derin öğrenme teknolojisinin gücünden yararlanan otomatik bir saldırı tahmin yaklaşımı önermektedir.

Araştırmamızın temel amacı, beş farklı hizmet reddi (DoS) saldırısı türünü tespit etmek ve sınıflandırmak için eğitilmiş tahmine dayalı modeller geliştirmektir. Derin öğrenme algoritmalarının yeteneklerinden yararlanarak, önerilen modellerimiz DoS saldırılarıyla ilişkili kalıpları ve anormallikleri etkili bir şekilde tanımlama potansiyelini sergileyerek erken tespit ve proaktif karşı önlemlere olanak tanır.

Titiz deneyler ve değerlendirmeler sonucunda çalışmamız, etkileyici bir yüzde 82,7586207'ye karşılık gelen saldırı tahmininin en iyi doğruluğuyla dikkat çekici sonuçlar verdi. Bu önemli doğruluk seviyesi, DoS saldırılarını doğru bir şekilde tahmin etme ve önleyici olarak azaltmada önerdiğimiz yaklaşımın etkinliğini ve uygulanabilirliğini gösterir. Sonuçta, araştırmamız, dijital alanda gelişen tehdit ortamına karşı koruma sağlayan pratik ve son teknoloji bir çözüm sunarak ağ güvenliğinin ilerlemesine katkıda bulunuyor.

Anahtar Kelimeler : Ağa izinsiz giriş, Kötü amaçlı faaliyetler, Veri alışverişi, Bulut altyapısı, Derin öğrenme teknolojisi, Hizmet reddi (DoS) saldırıları, Saldırı tahmini.

Bilim Kodu **:** 92432

ACKNOWLEDGMENT

I owe to my dear parents; I first and foremost give thanks to Allah the Almighty for this success and facilitation. My wonderful father gave me the most precious and priceless things to make me a man of honor. My lovely mother, who is skilled at engineering my heart with her prayers. To my family, in which I grew up, and its extension, which brings me pride and dignity. I owe a special appreciation to my thesis supervisor, Assist. Prof. Dr. İsa AVCI, who spared no effort in providing infinite assistance and direction till this thesis was completed to the best of my ability.

I also extend my thanks and gratitude to the University of Karabuk, including the wonderful professors, doctors, and colleagues who accompanied us throughout our academic journey.

I dedicate my thesis with gratitude to our motherland Iraq and the wonderful Turkey, which welcomed our scientific experiment and assisted in providing every possibility to graduate in this magnificent manner.

CONTENTS

Page

APPROVAL ii

ABSTRACT iv

ÖZET vi

ACKNOWLEDGMENT viii

CONTENTS ix

LIST OF FIGURES xii

LIST OF ABBREVIATIONS xiv

PART 1 1

INTRODUCTION 1

1.1. PROBLEM STATEMENT 2

1.2. RESEARCH QUESTIONS 3

1.3. RESEARCH OBJECTIVES 4

1.4. SIGNIFICANCE OF THE STUDY 5

PART 2 6

BACKGROUND AND LITERATURE REVIEW 6

2.1. OVERVIEW 6

2.2. INTRODUCTION TO CYBERSECURITY AND ATTACK DETECTION. 6

2.3. THEORETICAL FRAMEWORKS IN CYBERSECURITY 7

2.4. WIRELESS SENSOR NETWORKS (WSNs) 9

2.5. INTRUSION DETECTION SYSTEMS (IDS) 10

2.6. ATTACK VECTORS AND TYPES 12

2.7. MACHINE LEARNING 13

2.8. DEEP LEARNING 14

2.8.1. Fundamental Concepts of Deep Learning 15

2.8.2. Types of Deep Learning Models 18

2.8.2.1. Convolutional Neural Networks (CNNs) 18

2.8.2.2. Long Short-Term Memory (LSTM) 20

2.8.2.3. BidirectionalLong Short-Term Memory (BiLSTM) 21

Page

2.9. LITERATURE REVIEW 23

2.10. SUMMARY 27

PART 3 28

METHODOLOGY 28

3.1. INTRODUCTION 28

3.2. PROPOSED APPROACH 28

3.2.1. Datasets 29

3.2.2. Data Preprocessing 32

3.2.3. Modeling 34

3.2.3.1. Artificial Neural Network (ANN) 35

3.2.3.2. Long Short-Term Memory (LSTM) 37

3.2.3.3. Bidirectional Long Short-Term Memory (BiLSTM) 39

3.2.4. Evaluation Measures 45

3.3. SUMMARY 46

PART 4 47

EXPERIMENTS AND RESULTS 47

4.1. INTRODUCTION 47

4.2. EXPERIMENTAL SETTING 47

4.3. RESULTS ON WSNBFSF DATASET 48

4.3.1. Results of ANN Model 48

4.3.2. Results of LSTM Model 51

4.3.3. Results of BiLSTM Model 54

4.3.4. Results of CNN Model 57

4.3.5. Comparison 60

4.4. RESULTS OF KDD DATASET 67

4.4.1. Results of ANN Model 67

4.4.2. Results of LSTM Model 70

4.4.3. Results of BiLSTM Model 73

4.4.4. Results of CNN Model 75

4.4.5. Comparison 78

4.5. COMPARISON BETWEEN WSNBFSF AND KDD DATASET 83

Page

PART 5 89

SUMMARY 89

5.1. CONCLUSION AND FUTURE WORKS 89

5.1.1. Conclusion 89

5.2. FUTURE WORKS 90

REFERANCE 92

RESUME 98

LIST OF FIGURES

Page

Figure 2.1. ML types. 14
Figure 2.2. Artificial and biological neuron analogy 16
Figure 2.3. ANN structure 17
Figure 2.4. CNN architecture 19
Figure 2.5. LSTM architecture. 21
Figure 2.6. BiLSTMarchitecture 23
Figure 3.1. Proposed Approach. 29
Figure 3.2. Distribution of Attack Types of WSNBFSF dataset. 30
Figure 3.3. Distribution of Attack Types of KDD Dataset. 32
Figure 3.4. Data Balancing for WSNBFSF. 33
Figure 3.5. ANN architecture for WSNBFSF dataset. 36
Figure 3.6. ANN architecture for KDDdataset. 37
Figure 3.7. LSTM architecture for WSNBFSF dataset. 38
Figure 3.8. LSTM architecture for KDDdataset. 39
Figure 3.9. BiLSTMarchitecture for WSNBFSF dataset. 40
Figure 3.10. BiLSTMarchitectureforKDDdataset. 41
Figure 3.11. CNN architecture for WSNBFSF dataset. 43
Figure 3.12. CNNarchitecture forKDDdataset. 44
Figure 4.1. Training curves of ANN. 49
Figure 4.2. CM of ANN. 50
Figure 4.3. Classification report of ANN. 51
Figure 4.4. Training curves of LSTM. 52
Figure 4.5. CM of LSTM. 53
Figure 4.6. Classification report of LSTM. 54
Figure 4.7. Training curves of BiLSTM. 55
Figure 4.8. CM of BiLSTM. 56
Figure 4.9. Classification report of BiLSTM. 57
Figure 4.10. Training curves of CNN. 58
Figure 4.11. CM of CNN. 59
Figure 4.12. Classification report of CNN. 60

Page

Figure 4.13. Model accuracy and loss comparison. 62

Figure 4.14. Performance of various deep learning models on the WSNBFSF Dataset through Precision, Recall, and F1-Score metrics. 63

Figure 4.15. Heatmap of Model Performance. 64

Figure 4.16. Radar chart about the performance of different deep learning. 66

Figure 4.17. Scatter plot depicts the relationship between loss and accuracy. 67

Figure 4.18. Training curves of ANN. 68

Figure 4.19. Confusion matrix of ANN. 69

Figure 4.20. Classification report of ANN. 70

Figure 4.21. Training curves of LSTM. 71

Figure 4.22. CM of LSTM. 72

Figure 4.23. Classification report of LSTM. 72

Figure 4.24. Training curves of BiLSTM. 74

Figure 4.25. CM of BiLSTM. 74

Figure 4.26. Classification report of BiLSTM 75

Figure 4.27. Training curves of CNN. 76

Figure 4.28. CM of CNN. 77

Figure 4.29. Classification report ofCNN. 78

Figure 4.30. Model accuracy and loss comparison on the KDD dataset. 79

Figure 4.31. Performance of various deep learning models on the KDD Dataset through Precision, Recall, and F1-Score metrics. 80

Figure 4.32. Heatmap of Model Performanceon KDD Dataset. 81

Figure 4.33. Scatter plot depicts the relationship between loss and accuracy on KDD dataset. 82

Figure 4.34. Radar chart about the performance of different deep learning on KDD dataset. 83

Figure 4.35. Accuracy comparison between WSNBFSF and KDD Dataset. 84

Figure 4.36. Precision comparison between WSNBFSF and KDD Dataset. 85

Figure 4.37. Comparative analysis of the performance of different DL models on the WSN and KDD datasets. 87

Figure 4.38. Best Model Accuracy Comparison. 88

LIST OF ABBREVIATIONS

DoS	: Denial of service
ANN	: Artificial Neural Networks
LSTM	: Long Short-Term Memory
BiLSTM	: Bidirectional Long Short-Term Memory
CNN	: Convolutional Neural Networks
WSNs	: Wireless Sensor Networks
DL	: Deep learning
IDS	: Intrusion Detection Systems
OSI	: Open Systems Interconnection
NIST	: National Institute of Standards and Technology
GDPR	: General Data Protection Regulation
AI	: Artificial intelligence
ML	: Machine Learning
GPU	: Graphics Processing Unit
GANs	: Generative Adversarial Networks
ReLU	: Rectified Linear Unit
Tanh	: Hyperbolic Tangent
CV	: Computer Vision
NLP	: Natural Language Processing
CN	: Convolutional Layer
PL	: Pooling Layer
FC	: Fully Connected Layer
GR	: Grain Ratio
CA	: Correlation Attribute
SDAE	: Symmetric Deep Autoencoder
IOT	: Internet of Things
DNN	: Deep Neural Networks
SI-SLnO	: Self-Improved Sea Lion Optimization
TDMA	: Time Division Multiple Access

TPR	: True Positive Rate
TNR	: True Negative Rate
FPR	: False Positive Rate
FNR	: False Negative Rate
TP	: True positive
TN	: True Negatives
FP	: False Positives
FN	: False Negatives
VAE	: Variational Autoencoder

PART 1

INTRODUCTION

In the 21 st-century digital expanse, cybersecurity has emerged as a fundamental shield against a multitude of cyber threats to our interconnected society's fabric. This domain encompasses practices and technologies intended to secure networks, devices, programs, and data from attacks, damage, or unauthorized access. Developments in digital infrastructure have made cybersecurity a feature of national security and personal privacy. Economic, social, and political operations have become increasingly reliant on digital infrastructure, making it a target for unprecedentedly complex cyber-attacks [1].

The latter phenomenon has exposed the inadequacies of our traditional cybersecurity strategies, developed at a time when cyber threats were simpler and more static. The strategies, which include firewalls, antivirus software, encryption, and secure socket layers, are known to be ineffective against modern threats, including but not limited to, malware and adware that mutates to avoid discovery, phishing scams that are virtually undetectable, and advanced persistent threats that remain hidden in computer networks for years waiting to be discovered [2].

The limitations of traditional cybersecurity approaches are becoming more apparent. Rule-based algorithms and static defenses lack the flexibility to keep up with the attackers' continuously evolving tactics. Changing the rules that a security program requires manual effort to stay up-to-date, which is inadequate to address the ongoing evolution of cyber threats. Additionally, they are only as good as the latest known threat [3]. New exploits, albeit still using known techniques, are difficult or impossible to detect due to low dependency based on conditions.

The vast challenges above have forced the cybersecurity realm to undergo a more dynamic, intelligent, and adaptive paradigm shift [4]. An emerging subspace of artificial intelligence is deep learning, a series of technologies that mimic the neural networks of the human brain [5]. Deep learning models that process and learn from thousands of samples identify sophisticated items and anomalies that point to cyber risks, a step that human analysts and conventional methods are unable to achieve with precision and speed [6]. This change is not just a significant change of the technical brute strength of cybersecurity defenses; it is also a shift towards proactive and predictive measures. The cybersecurity industry is moving away from a defensive strategy and evolving to one of expectation and prevention, using deep learning to stay a move ahead of threat actors in the endless modern-age digital-arms competition.

Simultaneously, the WSN dataset offers a unique perspective on the security vulnerabilities inherent to wireless sensor networks, presenting distinct challenges that deep learning techniques are poised to address. By integrating deep learning with these datasets, this research aims to not only enhance the detection and mitigation of cyber-attacks but also to contribute to the broader discourse on the future of cybersecurity strategies, where adaptability and intelligence are at the forefront of defending our digital frontiers.

1.1. PROBLEM STATEMENT

The current context of the growing prevalence of the internet and digital infrastructure in various economic, social, and political operations, overall known as the digital age, and increasing sophistication of the cyber threats correspondingly make the potential consequences of the latter increasingly severe. Therefore, the importance of electronic security, or cybersecurity, protecting the networks, devices, and most importantly the data against the highly sophisticated attack, is impossible to overestimate. Most of the traditional cybersecurity solutions such as firewalls, antivirus programs, and encryption mechanisms have been developed to ensure the electronic security of networks and devices. This occurred when cyber threats were far simpler and static as most of them were driven by known signatures [7]. These mechanisms increasingly fail to secure electronic systems and data due to the mutation of malware, better orchestrated and informed phishing attempts, and advanced persistent threats regularly escaping the notice.

The cybersecurity industry is encountering the major problem in the limitations of traditional systems' ability to adequately confront the dynamic and complex dangers. Conventional systems are generally not flexible enough and lack the foresight to pre-emptively stem new dangers from emerging – such as zero-day attacks, which are attacks based on previously unrecognized software vulnerabilities. Conventional systems are often forced into passive defensive positions that always put them behind the attacker in the current dynamic security landscape. Particularly, conventional systems lack the ability to quickly expect new assault vectors and threats, meaning they should react to threats and fetch data from central administration for changes and amendment in their configuration – all of which introduce irrevocable delays that can be manipulated by attackers. Consequently, the difference in real-time latency and capacity for intelligent, autonomous cybersecurity resistance in security solutions empower deep learning formations. Given the limitations of the conventional method outlined above, it is evident that the above challenges demonstrate a robust case of the need for more evolutionary reforms in cybersecurity. This thesis will explore deep learning methods to change the way to identify and repulse network attacks. Through investigating Knowledge Discovery and Data Mining and Wireless Sensor Network datasets, this demonstrates how deep learning can redefine cybersecurity immunity into a more dynamic, smarter force that reduces vulnerabilities in the current model.

1.2. RESEARCH QUESTIONS

- How effective are deep learning models, specifically ANN, LSTM, BiLSTM, and CNN, in detecting anomalies and attacks within Wireless Sensor Networks (WSNs)?
- What is the performance of deep learning models on the KDD dataset for network intrusion detection, and how do they compare in identifying a broad spectrum of cyber threats?
- How do deep learning models' performances vary across the WSN-specific dataset and the KDD dataset, and what insights can be derived from this comparative analysis?

1.3. RESEARCH OBJECTIVES

- To Evaluate the Efficacy of Deep Learning Models in Anomaly and Attack Detection within Wireless Sensor Networks (WSNs): This objective focuses on the application and assessment of various deep learning models—ANN, LSTM, BiLSTM, and CNN—specifically tailored for the context of WSNs. It aims to explore how effectively these models can identify and classify anomalous behavior or potential security threats within sensor network data. Given the unique challenges presented by WSNs, such as resource constraints and the need for real-time processing, this objective addresses the critical task of enhancing security mechanisms through advanced computational techniques.
- To Benchmark Deep Learning Model Performances on the KDD Dataset for Network Intrusion Detection:The next objective involves evaluating the deep learning models previously selected on the KDD dataset, received as a standard benchmark within the network intrusion detection system. It examines how well these models detect a variety of cyber threats given the threats represented in the dataset. The objective aims to identify these models' capacities for learning with different cybersecurity data settings and threat landscapes and gather insights into these models' generalizability relative to other cybersecurity applications.
- To Conduct a Comparative Analysis of Deep Learning Models Across Different Datasets:he final objective brings together the results of WSN and KDD dataset analysis and compare the deep learning performance across differing datasets settings. This comparative analysis helps in identifying which models work best and how when certain types of data and anomalies are being identified. The objective aims to propose the best criteria for deploying these models across differing cybersecurity settings and lead to the contribution of optimal deployment wisdom within the field of Deep learning anomaly and attack detection.

1.4. SIGNIFICANCE OF THE STUDY

This study is significant because it has the potential to significantly contribute to the field of cybersecurity across Wireless Sensor Networks and all network systems via Deep learning models. Our research ensures that existing knowledge of Deep learning models performance via ANN, LSTM, BiLSTM, and CNN DL models derived from both WSN-specific and KDD dataset are well-explored. The study helps to bridge the existing gap to establish how these technologies can be optimized to assure good performance in the improved secure assured systems. This is important due to the advancement of technology through the Internet of Things where WSN is employed in data collection and transmission and thus making it open to cyber-attacks.

Our study is driven to evaluate which of the four DL models perform best in making decisions about threats in real-world scenarios and data sets. The comparison is essential for developing the best mean on how to implement Deep learning in cybersecurity . This is because the study participants will have a practical guide that they can use to configure the DL model to perform best in their specific DL model in combating incoming threats to their systems. Additionally, performance in simulated scenarios is essential for physics experimentation and the learning model in adapting to future threat changes. This is essential because cybersecurity is a rapidly evolving field. Thus the employment of DL models will ensure a better, resilient and smartly designed approach in the battlefield. By delivering this, I offer hope to the network WSN but also in general the implementation of Artificial Intelligence and machine learning.

PART 2

BACKGROUND AND LITERATURE REVIEW

2.1. OVERVIEW

This chapter has presented a foundational overview and a thorough literature review specific to the field of cybersecurity and the detection of attacks. Specifically, starting from the overview, the chapter has proven the subtleties of the big picture of cybersecurity and the theoretical framework of our defense strategies. In this context, the concept of WSN was discussed, and the importance of IDS was considered, and the concept of several threats to digital infrastructure was analyzed and dissected . Although this, additionally, key concepts of Machine Learning were studied: specifically, fundamental concepts and multiple types of Deep Learning models that reflect the most recent field developments in automating the detection and preventions of threats. Finally, the chapter also presented a literature review that classified the deviation of recent approaches in detail and documented their effectiveness. Ultimately, the final summary captured the essence of the literature review set the stage of future research or discussion.

2.2. INTRODUCTION TO CYBERSECURITY AND ATTACK DETECTION

It is more important than ever in the digital world to protect our virtual presence. The guardian of our virtual world is cybersecurity, protecting computers, networks, applications, and data from ill-intended cyber behaviors, unauthorized intrusions, or virtual harm of any level. The importance of cybersecurity is due to the prevalence of the internet in all aspects of our life and the everyday life of people. In addition, cybersecurity is essential not only to protect confidential data but also to enable the normal functioning of our digital life [8] . The field of cyber threats is extensive, evolving, and continually evolving, creating a persistent challenge to cybersecurity. This includes malware that disrupts or destroys the attack system and clever social engineering techniques to deceive individuals, as well as an attack that floods the system with excessive traffic, such as a Denial of Service and Distributed-Denial-of-Service attack, which makes the system not useable and legitimate user denial [9] . The level of sophistication and seriousness of these threats is increasing. The essence of cybersecurity is proactive detection and the alert to potential attacks. This is accomplished by monitoring the behavior of digital environments to recognize any deviations that might represent a risk [10] . Detection is critical since it enables the incident's impact to be reduced and the digital environment to continue operating. Nevertheless, as cyber criminals become more sophisticated and find new ways to bypass conventional safeguards, these techniques must evolve as well. Thus, information might be circulated to pattern analysis engines, which can then use rule-based reasoning and machine learning techniques to detect early indications of threat, failure, or compromise [11] . To keep current in the continuing digital monitoring, cybersecurity strategies must also evolve to address new threats. Advanced technologies are required, but so is a mentality of continuous education and improvement among cybersecurity professionals. Sharing information and knowledge across sectors and boundaries to enhance overall digital security are also crucial. Additionally, sharing insights and training ordinary people to use the internet safely are required as the first line of defense. To create a more secure digital future for people around the U.S. and the population of the world population, we all have to join our forces [12] . . Maintaining secure our virtual world demands relentless observation, innovative solutions, and a cohesive strategy. Our shared initiative to make greater cybersecurity is not limited to safeguard data or computers; it means safeguarding our digital lives.

2.3. THEORETICAL FRAMEWORKS IN CYBERSECURITY

Cybersecurity is a rich area composed of theoretical frameworks, models, policies, standards, and regulations that contribute unique ways of framing and solving the problem of protecting the digital framework in our contemporary environment. These foundational principles are not only vital tools of conceptualizing and mitigating risk but also guide about the culture, development, and practice of information systems. The Open Systems Interconnection model is a central theoretical support system for networking and communications security. The OSI model is a seven-layer outline that precisely explains how data is transmitted across networks from the physical implementation of hardware connection to the specific application context . By isolating network communications into seven layers -the physical layer, the data link layer, the network layer, the transport layer, the session layer, the presentation layer, and the application layer- the OSI model makes it easier to conceptualize and understand how networks work and how to secure them. This framework enhances the power to diagnose network problems, extend security measures, and ensure that information is maintained confidential and intact as it spreads through the digital landscape [13].

The DREAD risk evaluation model complements the OSI model by providing a sophisticated, process of evaluating and ranking threat levels in cybersecurity. DREAD stands for Damage Potential, Reproducibility, Exploitability, Affected users, and Discoverability [14].

In addition to these theoretical models, a plethora of policies, standards, and regulations are developed to define the best practices and legal requirements for information-security. ISO/IEC 27001 plays a crucial role, recommending a risk-based approach for ISM. It requires organizations to establish, implement, maintain, and continually improve their information security management systems to cultivate a culture of security awareness and resilience. ISO/IEC 27001 can be applied to all types of organizations and industries and has become the de facto benchmark for cybersecurity [15].

The National Institute of Standards and Technology offers a wide range of frameworks, the most popular of which is the NIST Cybersecurity Framework. It is a universal mechanism that helps organizations to align and implement a set of industry standards and best practices to help manage their cybersecurity risk and improve the authenticity, categorize, and availability of their information systems. These frameworks will help implement cybersecurity policies and laws, safeguarding organizational information in mainframes and ensuring business continuity [16].

In the regulatory field, the General Data Protection Regulation revolutionized the enforcement of personal data protection and privacy laws. Put in place by the EU, GDPR obliges organizations involved in data collection, storage, and management to take strict measures when dealing with personal data of subjects. Issued by the EU, GDPR has an international effect, as even those nations beyond the EU boundaries and their associations must adhere to it if they collect and process the data of EU citizens. In a period when the digital revolution is redefining the data eco-system, GDPR heralds the arrival of a new global privacy age=: a digital age [17].

2.4. WIRELESS SENSOR NETWORKS (WSNs)

Wireless Sensor Networks represent a crucial technology of the digital era, enabling a vast array of applications from environmental monitoring and smart grids to healthcare and defense. While the structural composition of these networks is instrumental for their successful operation and efficiency, their quality and characteristics depend on the choice of sensor nodes, topology, and communication protocol [18].

Sensor nodes are the primary elements of WSNs, with the simplest model including sensors of various readings. Such devices detect temperature, sound, vibration, pressure, pollutants, and other physical or environmental conditions. These components may be unifunctional or multipurpose depending on the desired sphere of application. Network topology, including star, tree, or mesh types, is also an essential feature that influences the parameters of WSNs like reliability, energy consumption or deployment convenience. Finally, communication protocols are developed for the implementation of the peculiarities of the wireless mode, energy saving, and effective data transfer [19].

The main challenges pertaining to deploying and managing WSNs are the ones linked to scalability, energy, and maintenance. Inability to scale blocks the deployment of a maximum number of new sensor nodes due to the comparatively small organizational capacity. Energy wastage is unavoidable as sensor nodes are situated in hard-to-reach locations and cannot be replaced or charged regularly. Maintenance might become impossible in the case of harsh external environment or a significant number of rendered nodes. Hence, it is necessary to develop the self-repair or maintenance-continuous functions. In this regard, not only the architectural but the operational aspects should be taken into account while developing and deploying WSNs. The global perspective for overcoming these challenges is promising as technological advancements continue driving the development of more efficient and sustainable WSNs. These innovations will support the further expansion of WSNs driving essential breakthroughs in data collection and real-time monitoring, ensuring the establishment of the connected digital world[20].

2.5. INTRUSION DETECTION SYSTEMS (IDS)

Intrusion Detection Systems are a critical component of network security architecture, a meticulously crafted mechanism that sifts through network traffic to identify potential digital threats and unauthorized actions that could jeopardize the integrity of a digital ecosystem [21]. The lineage of IDS is an evolutionary one, rooted significantly in the 1980s with work by Dorothy E. Denning, whose seminal paper “An Intrusion-Detection Model” played a vital role in conceptualizing IDS [22].

This academic initiative eventually led the creation of the Intrusion Detection Expert System by the Stanford Research Institute , one of the very first innovative systems that leveraged statistical-based anomaly detection together with user- and system-based profiling to detect malicious network activities. As the digital industry evolved over the years since the beginning of the 2000s, it gave way to more sophisticated cyber threats that transformed IDS from a unique innovation to an integral facet of cybersecurity best practices. The development of SQL injections, cross-site scripting and other similar sophisticated digital assaults by organizations was significant during this period.

Outpaced by the accelerating sophistication of cyber threats, IDS technologies have dramatically evolved, incorporating an array of intricate approaches significantly expanding beyond the rudimentary principles of their early predecessors. Anomaly detection, greatly enriched by modern artificial intelligence and machine learning , has represented a qualitative leap for IDS capabilities, providing the capacity to differentiate subtle alterations in network behavior that may indicate looming threats[23].

Based on data analysis for identifying irregular patterns, the application of this technique has allowed for the proactive detection of various critical security aspects, from infrastructure vulnerabilities to innovative cyber-attack strategies. Operating in tandem with anomaly detection, signature-based detection was and remains a central IDS method, facilitating the expedited identification of recognized threats by comparing network data to a depository of precompiled threat signatures [24].

Although powerful, this technique displays extensive shortcomings based on the fact that it can only address established vulnerabilities, rendering it relatively ineffective against novel threats.

Behavior analysis has emerged, serving as a vital new element within advanced IDS. Utilizing AI, machine learning and statistical analysis, behavior analysis delves deep into the minutiae of network data patterns to pinpoint not just identified malicious signatures, but primarily those behavioral anomalies which may indicate the onset of an approach. This approach reflects a shift from the perception of detection to predictive patterns analysis [25].

In conclusion, the journey of IDS from its origin to the advanced incorporation of AI and machine learning techniques represents a massive step in the field of network security. Additionally, these achievements are not just technical milestones but critical in the war against cyber threats. With terrifying statistics drawn from sources such as the 2017 Symantec Internet Security Threat Report, which captured millions if not billions of zero-day attacks, and the revelation that billions of data records have been compromised since 2013. The use of the IDS core techniques has never been more crucial. These facts attest to the hard work done fighting back cyber threats across the globe. In addition, they demonstrate the creativity and resilience fighting back the ghost of unauthorized invasion and maintained the integrity of the digital future.

2.6. ATTACK VECTORS AND TYPES

An attack vector can be defined as the collection of weapons systems or entry points relied upon by cyber adversaries to breach systems or networks to transmit malware, access confidential information, or disrupt vital services. According to Shamovskyet al. , these vectors are the weapons of choice for cybercriminals who conscientiously leverage system weaknesses and activate networks or computing devices without authorization. Indeed, the infection and masterful exploitation of vulnerabilities depend with some exceptions on the use of advanced recall vectors, which enable the attacker to by-pass security systems to achieve malicious objectives.

The domain of cyber threats partitions attack vectors into two broad classes: passive attacks and active attacks [26]. Passive attacks can be understood as the assailant's secret reconnaissance of a system, as he methodically scans fort open ports of security weakenesses without unnecessarily influencing the system's data or resources. These scaping techniques aggressively accumulate critical information about the target, at the expense of data confidentiality due to the lack of overt access. The silent nature of these assaults makes them difficult to detect because it leaves no residual evidence [27].

In comparison, active attacks are open campaigns designed to disrupt, weaken, or "break" an organization's system resources depriving it of its proper functioning. These aggressive strategies starve known vulnerabilities, implement denial-of-service measures to overload the system, rely upon users' weak passwords, or infect systems via malware and phishing efforts. Some of the attack vectors include phishing emails, compromised websites, deceptive pop-up ads, and deceitful instant messages. There are also unpatched software vulnerabilities and unsecure network protocols that may be attacked by cybercriminals. An attack surface is a broader concept that includes all of the abovementioned attack vectors yet further contains all available attack routes a criminal may exploit to launch an attack and send data or unlawfully access a system. This includes technical infrastructure as well as human factors within an organization, the field is susceptible to exploitation [28].

2.7. MACHINE LEARNING

Machine learning constitutes a critical component of artificial intelligence, which is "the practice of teaching computers how to learn from data and make decisions. It can be just mentioned as learning without already being programmed ,as noted in [29].

It is basically the development of algorithms that have the capacity to process and analyze data, and eventually act upon it. There are mainly three types of machine learning, specifically supervised learning, unsupervised learning and reinforcement learning, as shown in Figure 2.1. Among these types, supervised learning is the most traditionally used one. With this method, a training set into which the correct answers are "programmed" is offered to a learning algorithm. The algorithm makes predictions and then gets penalized for his mistakes. For instance, it is commonly used for "tasks ranging from fraud detection to spam filtering – to most fundamental tasks like predicting certain weather conditions" [30].

Unsupervised works with "input data that does not have the corresponding output results". The model has to develop the patterns on its own, which makes it particularly effective for when "one does not have any preconceived "ground truth" in the data." The method is used for example for clustering "similar customers by their buying patterns" or searching for the anomalies [31].

Reinforcement learning, in turn, involves "an agent that is trained through trial and error in some environment". The environment provides the agent with rewards for making the right decision and punishments for the wrong one. The method is used for example in robotics, video games or systems steerage [32].

Thereby, all of the aforementioned types of machine learning offers their unique techniques and possibilities and can be utilized in different applications, depending on the problem, the kind of data, and the learning goals. Being flexible and highly effective, they present an almost universal resource that can be used in research, analytics or industry.

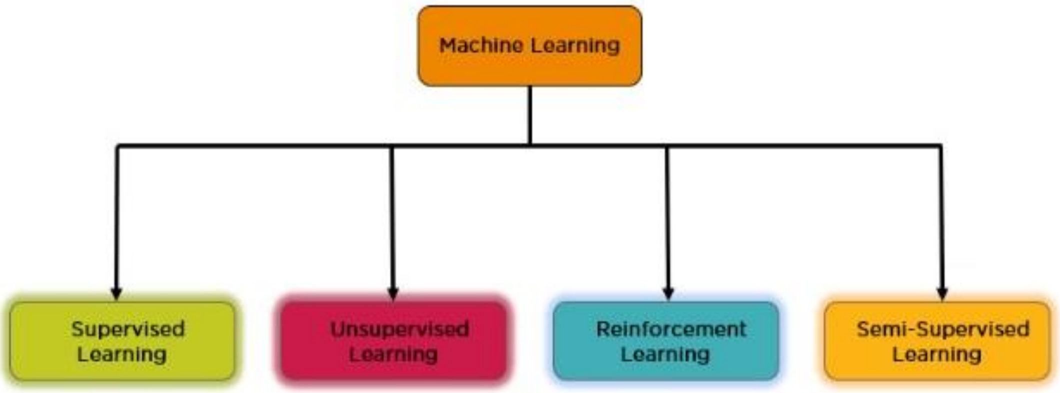


Figure 0.1. ML types.

2.8. DEEP LEARNING

Deep Learning (DL), the views of which could be called an extremely skilled part of Machine Learning (ML) that has been playing one of the main roles in the development of Artificial Intelligence (AI), is the concepts machines can derive conclusions as well as decisions from data [33]. DL capacity to process and analyze huge amounts of the unstructured data by means that have been designed like complicated neural networks is one of the distinguishing factors of DL. Contrary to the traditional ML techniques which rely upon the human identification of data features for the algorithmic processing and other machine learning tasks, a DL systems automatically learns the recognition of these features through successive layers of analysis. Artificial neurons in those layers once again mirroring the real brain's structure and functionality allow the system to learn from a large data set mostly without any manual guidance.

The history of the DL revolution started with McCulloch-Pitts neuron model [34] which axiomatically gives the basis of the neural networks that paved the way for the first Deep Learning demonstrations in the early 1950s. In particular, the figure of speech in the latter half of the twentieth century was particularly marked, especially with regard to the development of a backpropagation algorithm in the 1980s and 1990s, with the internal parameters being solved for better accuracy being refined [35]. Whilst these developments was led through the computational limitations and the lack of large data bases, nevertheless it remained ripe for the future. The DL revived in 2000s, as a result of the rise of large datasets and the introduction of advanced computing technologies, especially theGPUs processing units which made it efficient to process data at a higher speed.

Evidently, this upsurge was marked by cutting-edge advancements, whereby Geoffrey Hinton's deep belief networks [38] and Convolutional Neural Networks (CNNs) were the unrivaled choice among the developers in image recognition efforts [39]. AlexNet, a deep CNN emerged winner in 2012 ImageNet challenge as a golden reference among DL models with a capability of classifying complex visual tasks at high scale [38].

Thus, DL has made headway since then with RNNs being relatively new in the line up which has an expressed role in sequential data analysis and GANs for producing new data instances [39].

2.8.1. Fundamental Concepts of Deep Learning

Deep Learning (DL) is defined as the most advanced and prominent type of Machine Learning (ML) based on the initial principles of design strongly connected with artificial neural networks (ANNs)[32]. These nets create a similar architecture to human brain and possess many of its power. ANNs are made up of "neurons" units and are a collection of neuron layers that are interconnected through "synapses" which gain or lose strength over time as signals are transmitted among the neurons[40]. The architecture is responsible for providing ANNs with a skill similar to that of the brain, which implies that it can process and comprehend complex data patterns very proficiently.

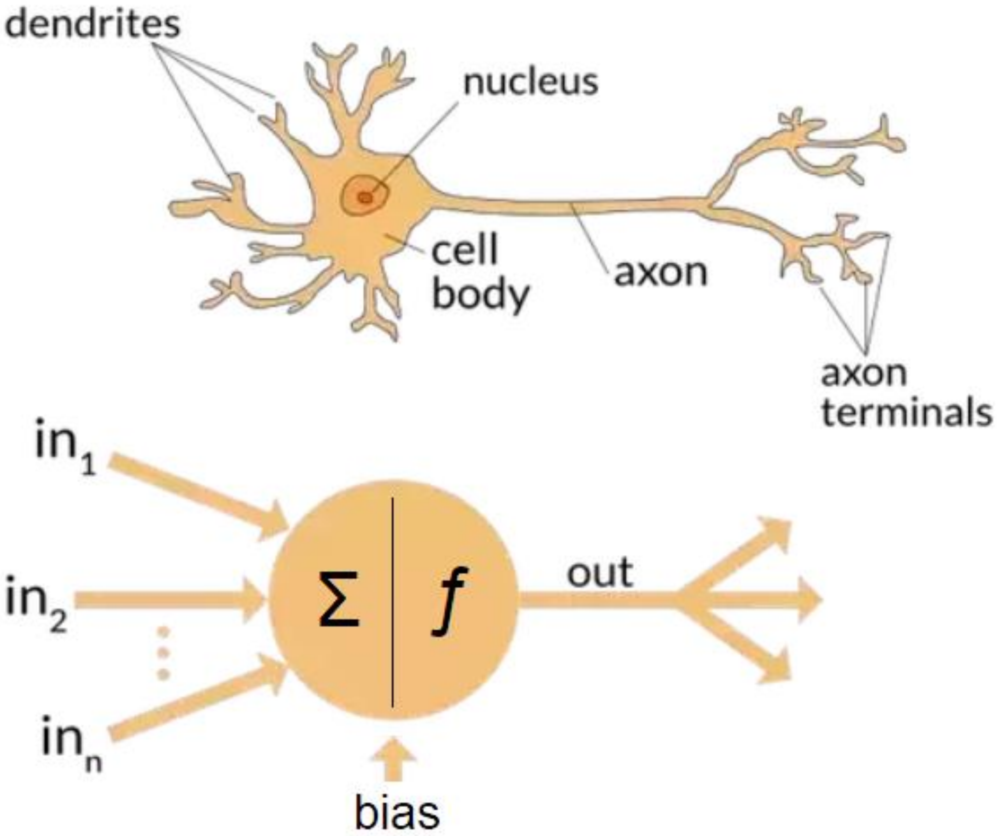


Figure 0.2. Artificial and biological neuron analogy [42].

A standard neural network is structured around three key layers: the first one is the input layer, which functions by absorbing the initial data; continued by the hidden layer/s, where there are array of interconnected neurons to make calculations; and

lastly, the output layer, in which the final output or prediction is produced [43].

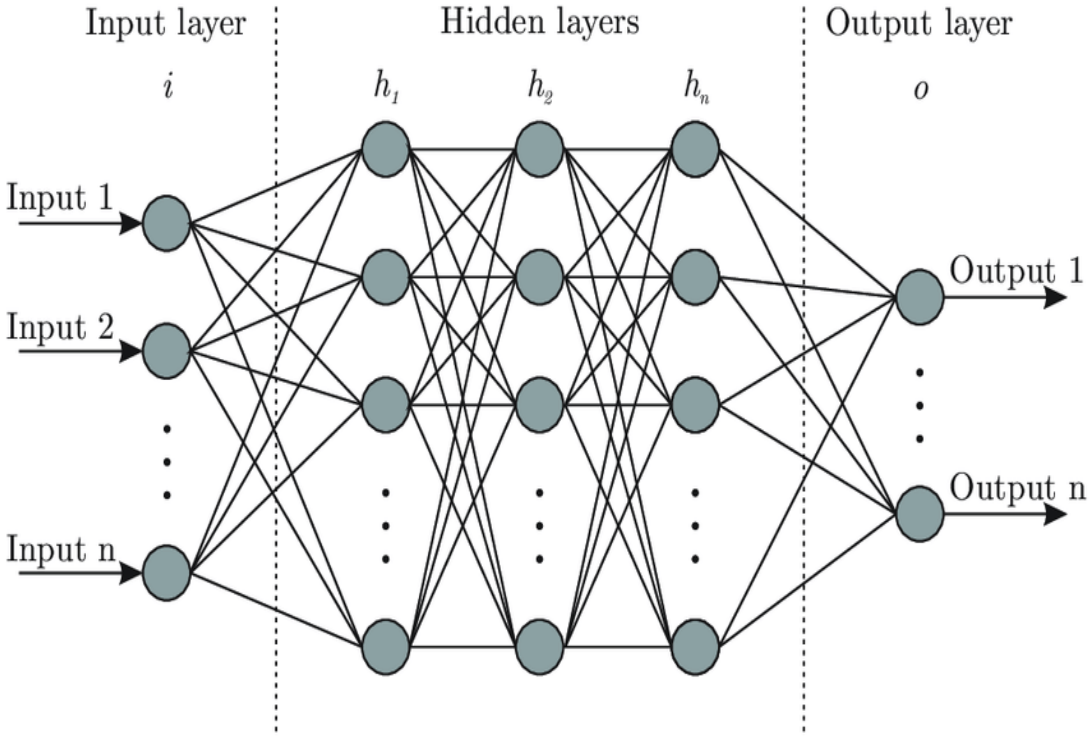


Figure 0.3. ANN structure [52].

A standard neural network is structured around three key layers: the first one is the input layer, which functions by absorbing the initial data; continued by the hidden layer/s, where there are array of interconnected neurons to make calculations; and lastly, the output layer, in which the final output or prediction is produced [43].

The neural network concepts calls for each layer to have its own role to play. DL involves layers of different types (convolutional layers, for instance, in CNNs and recurrent layers, for example, in RNNs) which are being used for different jobs [13]. Convolutional layers possess the ability to effectively deal with grid-structured data like images since passing through convolutional filters they can locate patterns and features within the data. Due to their structure, they are adjusted for the features of the sequential data, like text or time series, they leverage the input information to store the memory of the prior sequence elements.

Activation functions are the two indispensable element of neural networks, they are the ones who introduce non-linearity to the neural network [41]. Among the activation functions most commonly used are Rectified Linear Unit (ReLU), Sigmoid function and the Hyperbolic Tangent (Tanh). The ReLU's popularity is primarily due to two advantages: ease of computation and advantage in alleviating vanishing gradient issues which makes it better than other older functions like Sigmoid and Tanh. The Sigmoid function is used in a class of two output ranges namely'binary 'tasks because of the output range of 0-1 and the Tanh function is used in hidden layers because it helps standardize the input features with its output range of -1 to 1.

2.8.2. Types of Deep Learning Models

2.8.2.1. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNN), a must-have variant of Artificial Neural Networks (ANN), are dimensioned to process structured data, including images and audio signals. Different from classical ANNs, CNNs employ the practice of convolution in some or all layers of their internal structure instead of the frequent use of the matrix multiplication operations in those levels [44]. This phenomenon allows CNNs to learn the hierarchical characteristics of the input data at the same time, and the applications therefore are at the cutting edge such as in the field of Computer Vision (CV) and Natural Language Processing (NLP).

Figure 2.4 portrays CNN that has several phases, each of which has a purpose which is targeting specific actions on activated data. The layers employed here are the CN, PL, and, finally, FC layers. The core of CNNs is formed by the convolutional layers that are based on filters or kernels that multiply and add elements slipstream the selective input data in the fixed-size topology. What it does is that it generates a feature map to emphasize the prime data values. Of the underlying convolution operation and output feature map dimensions configuration, the filter size, stride and input padding are determinants.

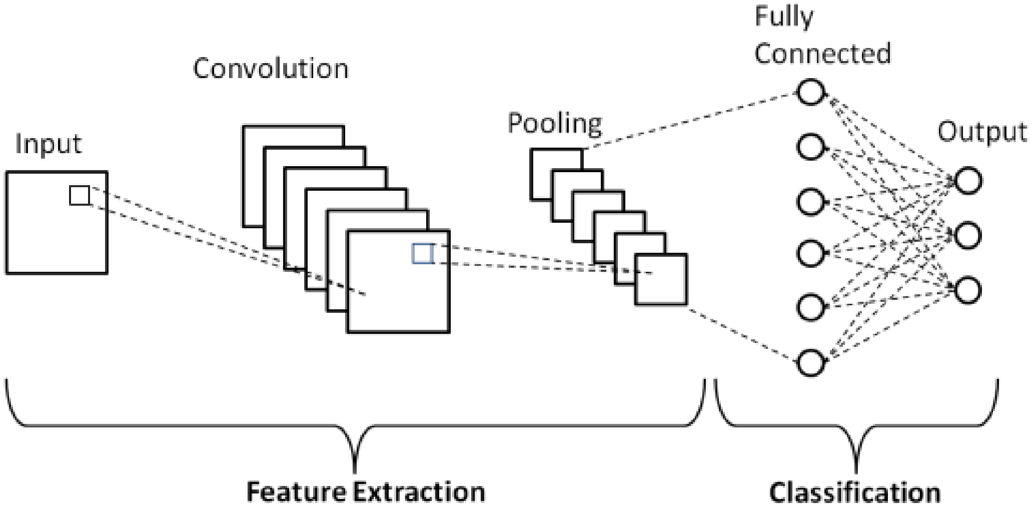


Figure 0.4. CNN architecture [45].

The convolutional layer is essentially a filter used by CNNs to identify the features contained in the input data, e.g. edges or textures Then, this means that pooling layers do the job of downsample the feature map results. The max-pooling technique is an example of how dimensionality of the feature maps is made reduced. With this technique the network remains robust against slight shifts on the position of the input and the computational burden is also reduced accordingly. Is an effect, that the pooling has on the feature maps, determined by a factor of the dimensions of the pooling region and the stride that is used for the pooling process.

Finally, in the end of the network, the fully-connected layers step in as a bridge for converting and classifying the received data. They summarize the entirety neuronal network building connections between every layer of consecutive neurons that as a result of these linear and non-linear transformations output the ‘final decision’. This kind of neural network architecture is inspired by the hierarchy of the visual system in humans, which helps identify and parse visual patterns into semantically meaningful representations at each level of the hierarchy, with the final target being either analyzing or classifying the whole input data.

The CNNs have been a crucial piece for quite a lot of fields, from the LeNet-5's that made a landmark in digit recognition to AlexNet that forwarded deep learning for computer vision, and the Transformer's whose major contribution we have for the new natural language processing. Therefore, this process clearly reflects the weight role of CNNs' ability to tackle and learn from subtle forms of data patterns.

2.8.2.2. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks, in which the researchers Hochreiter and Schmidhuber introduced in 1997 as a specialized form of Recurrent Neural Networks (RNNs) capturing long-term dependencies in sequential data. They were developed to tackle numerical difficulties in the classical RNN which were described in the form of vanishing and exploding gradients. The main trait of LSTMs is their complicated structure that also comprises of a memory cell that has an ability to store and update data for a long time. This structure is supported by three types of gates: input, forget, and output gates that perform as the sigmoid neural networks. These gates perform these functions perfectly as timing of the erasing, storing, and exchanging the information [45].

An LSTM unit's design incorporates four main elements: the input gate (i_t), the forget gate (f_t), the output gate (o_t) and the memory cell (c_t) as presented in Figure 4. The role of the input gate in the cell solely depends on the current input (x_t) and the previous hidden state (h_(t-1)). The forget gate attunes the provision of data from the earlier cell state (c_(t-1)). The output gate determines the appropriate amount of the current cell state (c_t) to be outputted as a hidden state vector (h_t). The status of the memory element is changed within the operations of addition and multiplication only.

The operations within an LSTM unit are governed by the following equations:

$$\mathbf{i_t} = \boldsymbol{\sigma}(\mathbf{W_i x_t} + \mathbf{U_i h_{t-1}} + \mathbf{b_i})$$
$$\mathbf{f_t} = \boldsymbol{\sigma}(\mathbf{W_f x_t} + \mathbf{U_f h_{t-1}} + \mathbf{b_f})$$
$$\mathbf{o_t} = \boldsymbol{\sigma}(\mathbf{W_o x_t} + \mathbf{U_o h_{t-1}} + \mathbf{b_o})$$
$$\widetilde{\mathbf{c_t}} = \mathbf{tanh}(\mathbf{W_c x_t} + \mathbf{U_c h_{t-1}} + \mathbf{b_c})$$
$$\mathbf{c_t} = \mathbf{f_t} \odot \mathbf{c_{t-1}} + \mathbf{i_t} \odot \widetilde{\mathbf{c_t}}$$
$$\mathbf{h_t} = \mathbf{o_t} \odot \mathbf{tanh}(\mathbf{c_t})$$

In this context, σ symbolizes the sigmoid function, and tanh refers to the hyperbolic tangent function. The element-wise product is shown with the notation of $\odot$. The characters W, U and b stand for weight matrices and bias vectors, respectively. The capacity of LSTMs to identify the intricate sequential patterns and retain information over long intervals helped them become a mainstream use with a leading edge performance in most of the applications.

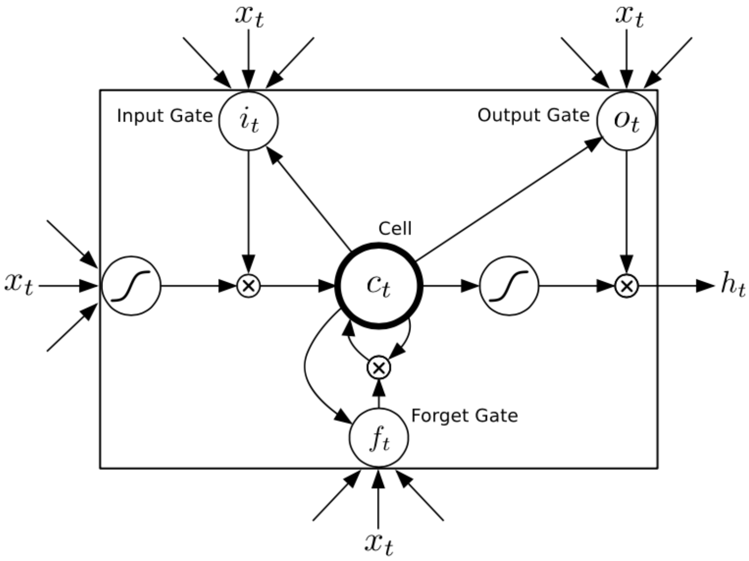


Figure 0.5. LSTM architecture.

2.8.2.3. BidirectionalLong Short-Term Memory (BiLSTM)

BiLSTMs are neural network structures based on advanced principles of sequence modelling and prediction, which is a dynamic representation of data. As illustrated in Figure 2.6, a BiLSTM network is structured with two distinct layers: a forward layer and a backward layer and an input layer to output layer as well. Contrarily, a forward layer reads the sequence from start to finish but a backward layer reads the sequence in the opposite direction from end to start, which helps to get information from both past and future states of the sequence [47].

The dual-layer architecture, together with the capability of capturing contexts, is what makes the BiLSTM different and superior to conventional unidirectional LSTMs. Another instance is the ability of the forward layer to remember the beginning of a sentence and the backward layer to retain infomration about the end of it thus providing more concise information regarding the meaning and structure of the sentence. Each neuron in the forward and backward layers is connected via weighted connections, this weight is symbolized as ω in the figure, with varying weight assigned to different layers, so as to reflect the unique directionality of the information flow in those layers.

The sequence data that enters the input layer is being transmitted by the forward and backward paths sections. The forward pass and the backward pass are independent of each other, they do not share the directional context. At each time stamp, the network blends structures from both directions by typically concatenating or summing the outputs from both layers to pass to the output layer. This, thus, yields a holistic impression of the trend that unfolds along the whole time interval enriched by the data from the beginning and from the end.

Combining the result of the previous two layers yields a multi-way vector, which is then further in depth processed by the third layer that outputs the final predictions or classifications. The illustration sleekly conveys what the working of BiLSTM neural network is, where the circular nodes represents neurons, the arrows signifies the direction of data flow, and the ω symbols communicates the weight values that modulate the data as it is passed through the network.

BiLSTMs are widely used in the tasks where a full grasp of a sequence leads to more accurate recognition, for example, speech recognition, text translation, and time-series analysis, every data point is affected by the surrounding elements which impact the quality of a result or decision. By Amassing of the data both the past and moving toward the future direction, BiLSTMs supply a distinction that it is unfounded and exceptional for its counterparts who only have the access to the unidirectionality.

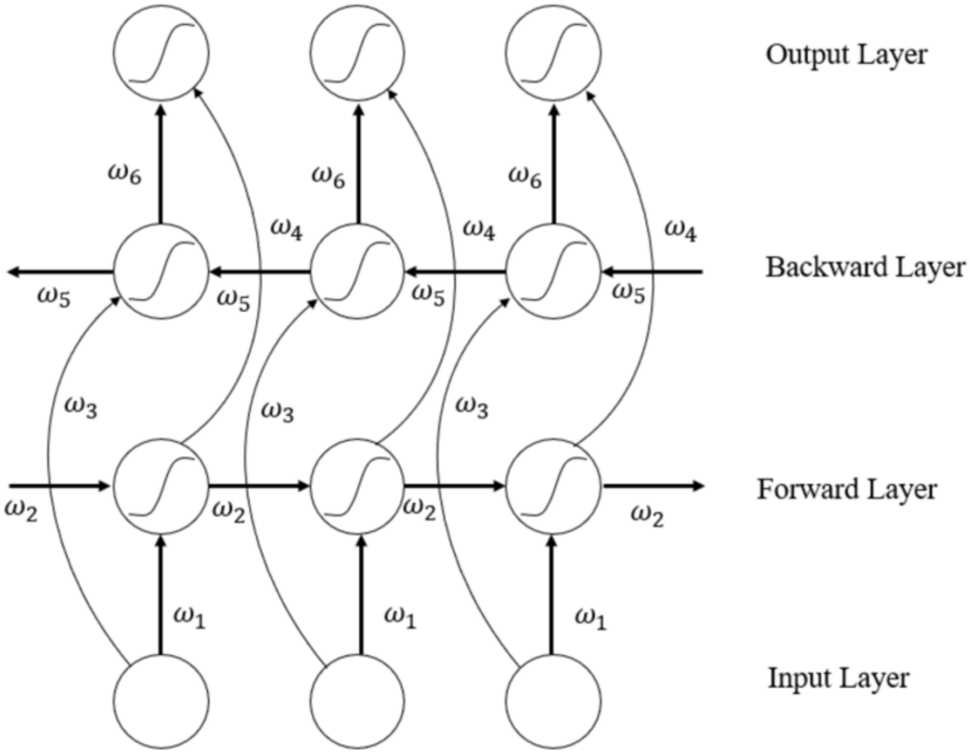


Figure 0.6. BiLSTMarchitecture [46].

2.9. LITERATURE REVIEW

The skyrocketing reliance on the internet and the subsequent expansion of the range of remote services have brought about the pressing need for advanced network security measures which have become shinning examples of atonement for escalating cyber dangers. With the advent of networking vulnerability, the network intrusion detection area has started encountering enough developments over the time that are targeted at strengthening the system of internet security of the wireless sensor networks and other vital infrastructures.

Saying that [48] makes a reference to the BAT-MC approach, which combines BLSTM and comes with an attention mechanism for catching network attacks, this architecture bringing no demand for manual feature design. The model is built up on attention to highlight pertinent traffic vectors that were produced via BLSTM and utilizes multi-layer convolution to extract traffic data features at local level. The main feature of designed solution as an end-to-end one, it will be performed the traditional engineering feature and it will extract the key points hierarchically. In terms of performance testing on a ground-truth dataset with other models, BAT-MC is at the peak in terms of refined anomaly detection.

In the meantime, this research paper [49] considers the use of AI in the domain of cybersecurity by providing an outline for network intrusion detection between ML and DL. Here, two procedures are described for an Artificial Neural Network (ANN) and a Recurrent Neural Network (RNN) which are also IG-based selectors of features. The cognitive style of the second method based on grain ratio (GR) and correlation attribute (CA). This methodology employ was approved through NSL-KDD dataset, in which the results showed that RNN can outperform ANN, and other machine learning classifiers to detect network related attacks and intrusions.

In addition, the article presents the current problem of network intrusion detection by employing the deep learning (DL) potential. The study proposes a non-symmetric deep auto-encoder architecture which is meant to refine the detection of network attacks. The model's functionalities and performance are specified followed by validation using the KDD CUP'99 dataset. This DL-based network intrusion detection system is implemented by the TensorFlow library and GPU framework. It has an impressive accuracy of 99.65% and appears to be a strong candidate for application in network security research and DL-based detection and classification systems.

Moreover, the work [51] argues that NIDS should be enhanced because of the augmenting need for a human element and the decline in the detection accuracy in existing systems. It proposes a novel intrusion prevention mechanism relying on deep learning, particularly a Symmetric Deep Autoencoder (SDAE) for an autonomous feature extraction. In addition, a complex research classification model utilizing stacked SDAEs is suggested. Such a model has been implemented and evaluated by using the NSL-KDD and CICIDS 2017 datasets via the TensorFlow GPU package, and the experimental results are relatively good and made progress compared with the conventional NIDS solutions.

In another instance, the paper [52] looks at the spread of IOT technology and the increase in security breaches that occur as it grows. It suggests the application of DNN

(Deep Neural Networks) for rapid and accurate detection of malicious activities within the IoT networks. The research highlights the importance of good quality datasets for the deployment of the intelligent intrusion detection system. Along with that it determines the model of DNN performance using well-known datasets such as KDD-Cup'99, NSL-KDD and UNSW-NB15, which shows that the proposed method reaches at least 90% accuracy across these datasets.

The article [53] is focused on security vulnerability of sensor networks (WSNs) which include lightweight and restricted nodes along with their complex deployment strategies. The report demonstrates the necessity of devising efficient techniques to trail access controls so as to boost security of WSN. Yet, Machine Learning techniques (ML) are usually chosen by many network intrusion detection systems (NIDS) designers, yet they do not scale well to dynamic environment with imbalanced attacks. The implementation begins a Neural Network Deep (DNN) -based network intrusion detection scheme, which uses a cross-correlation process for feature seleciton purposes optimizing its performance. This DNN organization, unlike the conventional ML techniques such as Support Vector Machine, Decision Tree and Random Forest, appears to perform better in detecting attacks, by the result of the experiments.

On the other side, paper [54] also figures out an advanced IDS method for WSNs that are susceptible to different kinds of attacks because of the unsecured way of transmitting data. This algorithm have an advanced mechanism for choosing the CH that looks at considering the remaining energy of sensor nodes and includes other factors including delay and transmission distance. Our newly developed self-improved sea lion optimization (SI-SLnO) method is positively applied during the course of this selection process by means of a two-tiered multidimensional hierarchical trust model that controls the two-fold trustworthiness of the CHs and network nodes based on the integrity, sincerity, and interaction trust. Attaining deep learning outputs is NN-based method using an input dataset weighted by the SI-SLnO technique. This approach's reliability is seen by contrasting it with the typical available ways.

The journal [55] pinpoints security issues as one of the most important ones in a world of today in which data mining is firmly entrenched and discusses some previous research into data mining-based intrusion detection systems. It admits that the techniques are far from being perfect. It then suggests a new intrusion detection system that is capable of identifying intruders in wireless networks much more efficiently. On the other hand, the system under consideration is equipped with a unique attribute selection algorithm, that includes conditional random field as well as linear correlation coefficient which eliminate the most relevant features. Following that, such features are passed into an available CNN. The performance of system was evaluated using tenfold cross-validation, and the experiment resulted in the high overall detection accuracy of 98.88%.

Research [56] considers the main features of ubiquitous computing and the growing reliance on wireless sensor networks (WSNs). The issue of security threats becomes a most concerning problem in WSNs of different types of attacks including the Denial of Service (DoS), Black hole, Gray hole attack, Flooding, and TDMA attacks, due to the distributed and decentralized nature of the WSN. The project consists of a survey of today's cutting-edge security solutions and proffers an intrusion protection architecture based on deep learning. Moreover, it would describe the proposed system's performance in terms of existing solutions and deliberate on the implications of the results and perspectives in future research directions.

In the article [57] the author focus on the growing trend of the people who relies heavily on internet and the increase of virtual services which underline the fact that network security and fast attack detection are becoming more important as the cyber threats are increasing. It criticizes the prevalent machine learning-based intrusion detection system models in WSNs that heavily rely on a single detection layer. The computationally heavy algorithms cannot be used to scan for suspicious activities due to limited resources. The authors come up with the solution of multiple layer intrusion detection system for WSNs that takes the defense approach of two detection layers. The first layer is situated at the network edge with distributed sensors which employ a Naive Bayes classifier for immediately packet inspection . The subsequent layer that is hosted on the cloud is given to a Random Forest multi-class classifier for a comprehensive packet analysis. As verified by the simulation results, the model displays excellent performance for metrics such as TPR, TNR, FPR, FNR, and

Precision, which indicate its good functionality against normal, flooding, scheduling, grayhole, and blackhole attacks.

2.10. SUMMARY

This chapter continues with a more descriptive approach, looking into both the context and the relevant documents of the current security environment. It starts by providing a wider definition for cybersecurity which play crucial role in attack detection; then, it develops theoretical frameworks as a core of practice. The curriculum, as proposed then, will portray the nuance of Wireless Sensor Networks (WSNs) and the problematic that they face, leading to a lecture and analysis of Intrusion Detection System (IDS) and the attack vectors that they are subjected to. The fundamental concepts and model types in Deep Learning will be looked critically after doing a comprehensive analysis of how Machine Learning contributes to cybersecurity. The careful review section which follows the results section studies and analyzes the recent advances in the field, laying emphasis on the new approaches and comparing them with the traditional ones. Lastly, a conclusion reintegrates the chapters overarching ideas, depicting a coherent understanding of the existing evidence, leading up to the research that lies ahead naturally.

PART 3

METHODOLOGY

2.11. INTRODUCTION

In this part of our study's methodology, we discuss the specific steps we completed to address the critical challenge of detecting and classifying suspicious events and incidents of network security. This section encompasses a primary component of this thesis, laying out a comprehensive blueprint of our research process that goes from finding data to the final correlation of the outcomes generated by our models. The approach is designed to provide a step-by-step commentary about how we chose and preprocessed our datasets, the particular procedures used to process the data, the creation and training of more sophisticated learning models, and the systematic appraisal that was used to validate the quality of their predictions. Within these sections, we outline the methods that surface as the theoretical underpinning of our addition to the cybersecurity domain.

2.12. PROPOSED APPROACH

To enhance network security by identifying anomalies and potential threats, we have prepared a se together within this complete system some advanced deep learning methods and intensive analysis of the WSNBFSF and KDD data accumulations as evidenced in the workflow presented in Figure 3.1.

First, one must prepare the data works and steps for each collection. This preparation must complete the data modifications to ensure that each set is trained and test-ready and include work, such as normalization and encoding and balancing the data where a given point data contains an equal number of occurrences of each scenario, allowing the models to train without bias.

This refers to using a variety of deep learning designs. Each model is developed to recognize a different point and shape within the data. These models include artificial neural networks, ANNs, which are the simplest and best when recognizing obvious patterns; long short-term memory, LSTM, networks that recognize patterns over time; BiLSTMs, which read out and backward simultaneously to give them a wider picture, while convolutional neural networks are excellent at recognizing spatial data.
After, the models are trained and put into prediction, with the WSNBFSF data focusing on categorizing traffic into Normal, Flooding, Blackhole, and Forwarding. The KDD dataset allows the models to be condensed into recognizing normal and anomaly categories, simplifying to a two-point data.

This strategy exposes the versatility ad robustness of deep learning in tackling cyber security by remaining areas dissident between an innocent and a potential bad outcome across numerous settings. Our plan aims to develop a reliable and proactive approach to the discovery and prevention of cyber threats.

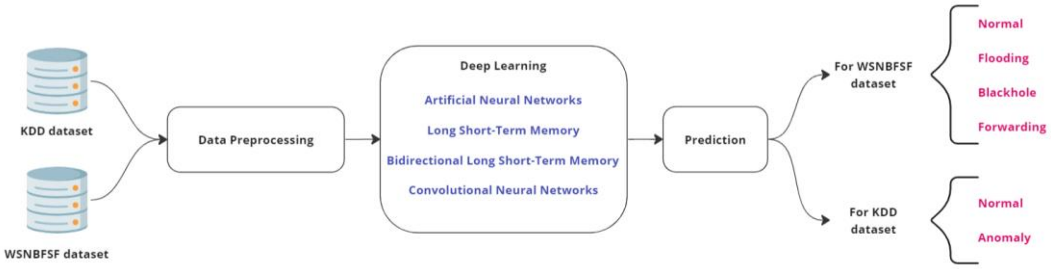


Figure 0.1. Proposed Approach.

3.2.1. Datasets

The WSNBFSF collection is a comprehensive dataset that imitates numerous scenarios of a Distributed Denial of Service attack that may occur in the context of a Wireless Sensor Network. The collection features three types of attacks: Blackhole, Flooding, and Selective Forwarding. The dataset consists of 18 unique attributes and contains over 312,106 records, which is sufficient for detailed analysis. The dataset has four traffic types namely Blackhole, Flooding, Selective Forwarding attack traffic and Normal traffic. The dataset's multivariate WSNBFSF situations to which the intrude lead to the seek is a perfect situation for estimating the accuracy at which IDSs operate in varied situations. This is the essence of a good dataset since it real and based on exact data in real-time and correctly categorized. Therefore, this dataset is good for testing the performance of deep learning algorithms applies in uncovering security flaws in WSN. Figure 3.2 below shows the percentage distribution of the four traffic types in the WSNBFSF datasets.

The horizontal axis divides arranges the traffic into four categories according to patterns: ‘normal,’ ‘Flooding,’ ‘Blackhole’, and ‘Forwarding.’. The vertical axis visualizes how often each pattern in the dataset. It is evident from this graph that the dominant pattern is ‘normal,’ with 262,851 counts. As a result, it is clear that most network activities do not threaten it. Meanwhile, the major pattern of attacks is ‘Flooding’ with 29,844 counts. The second is ‘Blackhole’ with 11,766 counts. The ‘Forwarding’ pattern is the most rare in this dataset with 7,645 counts.

The pattern of distribution for the WSNBFSF dataset is imbalanced; attacks are distinct and represent a small sample size of the whole. Similar biases can be observed in machine-generated datasets, which can negatively affect the machine learning model’s performance.

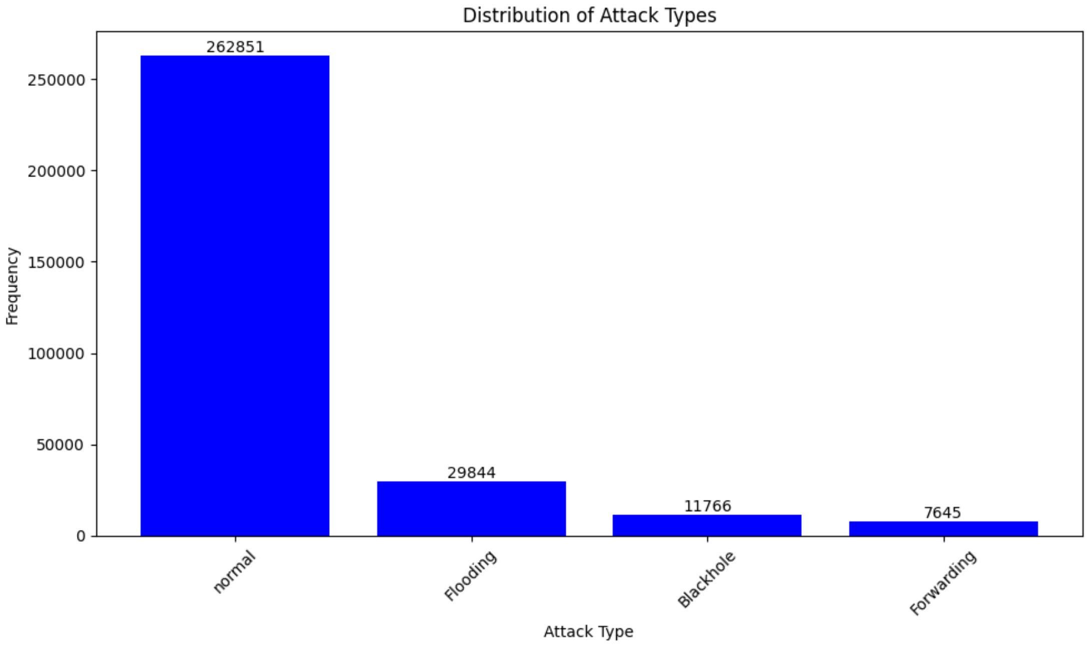


Figure 0.2. Distribution of Attack Types of WSNBFSF dataset.

One of such widely recognized standards to test network intrusion detection systems is the KDD dataset. This dataset includes an extended set of network traffic information, providing a basis to design and test systems capable of identifying cybersecurity risks. It is encapsulated in a dataframe called ‘df_final’ with 137,823 records. Each of them includes 43 exact data types describing various aspects of network behavior and interaction.

This includes basic information about the type of protocol used, service accessed, and connection length, as well as more complex information like secret data that was sent and received , connection status , and the information containing how the connection’s state bars from the mean, for example, when there are indicators of capturing ‘land’, ‘wrong_fragment’, and ‘urgent’. Moreover, the dataset includes features that describe connection’s state and content, such as a metric called ‘hot’ that is a connection to significant or dangerous websites, and another one, ‘logged_in’ that contains information about whether a user was logged into the network while communicating.

Furthermore, KDD publishes information about the other interconnections: behavior features describe how the connection-video behaves on the network, including metrics like ‘count’ and ‘srv_count’, how many of them incur errors and how often they repeat an error on dataset , and how many of their traffic interaction is on the same servers and on different servers . Another set of measures is host-based detail, how many connections the host had, and how many of them were to the same server.
Illustrated in Figure 3.3, this dataset is specially valuable since it labels each of the connections as ‘normal’ or ‘anomaly’ connection. The amount of classes is detailed in. It shows that the classes are quite balanced, with ‘normal’ class occurring 69,495 times and 68,328 – ‘anomaly’. Such balance is critical for machine learning in intrusion detection, as overly imbalanced data will skew the model’s results towards the higher-occurrence class.

Generally, making one class appear more in training data than others leads to the model being biased towards this class. Such behavior usually leads to model non-performance, as in case of anomaly detection the said model is expected to predict the class with fewer examples. However, in this dataset, this is not the case, allowing for more accurate modeling and predictions.

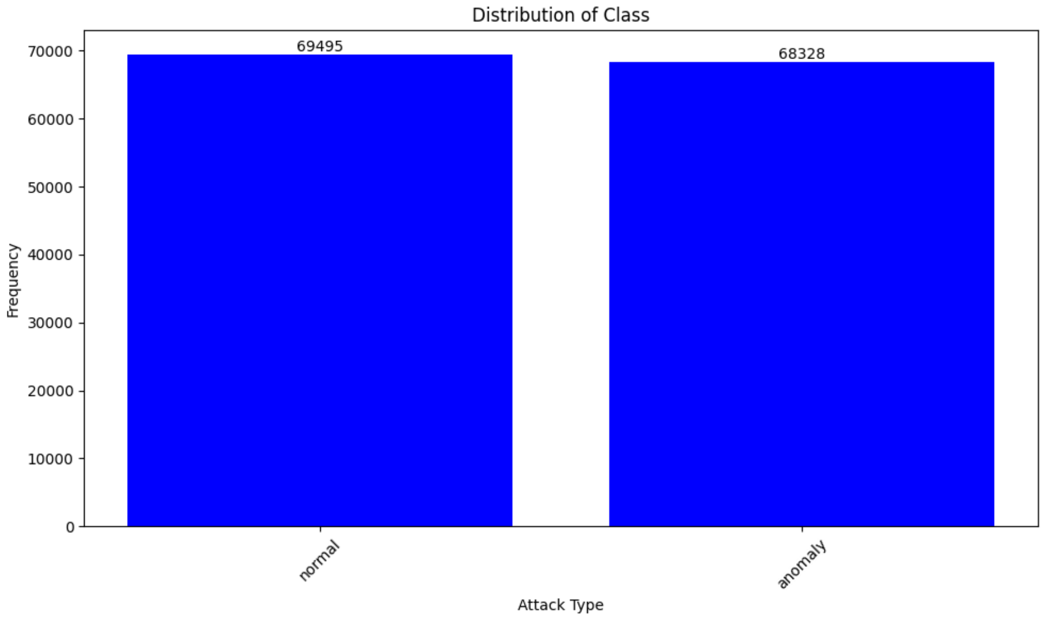


Figure 0.3. Distribution of Attack Types of KDD Dataset.

3.2.2. Data Preprocessing

Before you can successfully train a machine learning model, especially one aimed at identifying unusual patterns or security breaches in network traffic, it's essential to properly prepare your data. This preparation process, known as data preprocessing, ensures that the datasets are in the best possible shape for the models to learn from them effectively. Both the WSNBFSF and KDD datasets undergo several preprocessing steps to ensure they are suitable for the subsequent analysis.

For the WSNBFSF dataset, one of the primary concerns is the balance among different classes to prevent model bias towards the more frequent classes. As illustrated by Figure 3.4, the data balancing technique involves downsampling the dataset to equalize the number of instances in each class, which helps ensure that the models trained on this data do not overfit to the majority class. This is achieved by first determining the smallest class count, and then using the resampling method from the scikit-learn library, with the `resample` function being applied to each attack type subset without replacement, targeting the count of the smallest class, thus achieving a balanced dataset.

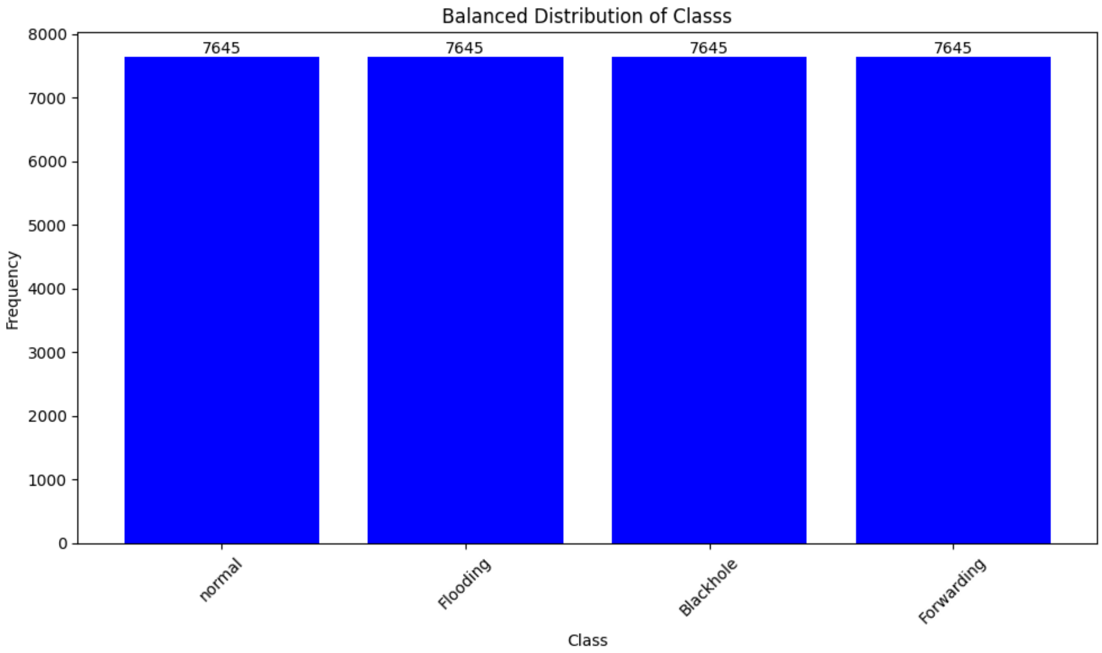


Figure 0.4. Data Balancing for WSNBFSF.

After balancing, the features are standardized using a `StandardScaler` to normalize the data, ensuring that the model is not skewed by the scale of different features. The standardized dataset (X_{scaled}) is thus derived as follows:

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma}$$

where (μ) and (σ) are the mean and the standard deviation of each feature. The dataset is divided into train and test sets while the target is encoded. Notably, one-hot encoding is used to convert categorical class labels into a format that can be passed into the machine learning model. For the KDD dataset, the categorical variables including 'protocol_type,' 'service,' and 'flag' are encoded with label encoding, whereby a unique integer is assigned to each category. The class labels in this case are binary encoded [0,1] with 'normal' assigned to 0 and 'anomaly' 1. The data is then normalized using MinMaxScaler. This function scales features to 0 to 1 according to:

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

The scaled features, (X_{scaled}), are then separated into training and testing sets at a ratio of 0.2 to check the model's generalization. The target features are also one-hot encoded to ensure the target class representation for each of the algorithms used. After the encoding and scaling, the datasets are partitioned into two: the one that the machine learning model will be trained on, the training set, and the one that will be tested to evaluate the capability of the model under test conditions. This partition is usually randomly selected to merit the test with regard to the unpredictability of the new, unseen data value. Thus, the test set is typically a certain percentage of the dataset. For example, the percentage could be 20% or 30% and so on. The remaining of the percentage say 70-80% is for training the model . The reason for partitioning is for learning as 70-80% of the dataset could be trained, and the remaining proportion is necessary for testing, in this case, the model's generalization capability. Hence the datasets have been transformed to suit the machine learning model, with features normalized and class proportions balanced with encoded labels to signify possible capability levels.

3.2.3. Modeling

Developing powerful predictive models is crucial in the act of cybersecurity, allowing the detection and classification of potential threats. The next section focuses on the complex architectures of the deep learning models that were developed to analyze network traffic in the WSNBFSF dataset and the KDD dataset, known throughout the IT and cybersecurity fields. The Artificial Neural Networks, Long Short-Term Memory networks, Bidirectional Long Short-Term Memory and the Convolutional Neural Networks are built and extensively trained to identify the subtle patterns associated with normal functioning and malicious behavior.

For the WSNBFSF dataset, the topologies are trained over 30 epochs with a batch size of 64, allowing the models to learn adequately and maintain computational capabilities. These values were chosen based on the authors' attentive monitoring of the models' performance during training to ensure convergence without overfitting. The KDD dataset, with its unique feature arrays and distributions, are trained for 10 epochs with the batch of 64. Once again, the choice of the number of epochs is dictated by the complexity and the total size of the dataset, which calls for a different approach to ensure the models are neither under nor overfitted.

3.2.3.1. Artificial Neural Network (ANN)

The Artificial Neural Network model is used as the computational frame in our work to classify network traffic as multiple categories indicating normal or malicious activities. The ANN for the WSNBFSF dataset was well-designed with an input layer of 17 neurons, according to the dataset's 17 features(Figure 3.5).

It was followed by a hidden layer structure with three layers with neurons being 128, 64, and 32 neurons, respectively. All hidden layers used the ReLU activation function known for its efficiency and addressing of the vanishing gradient problem. The final output layer of 4 neurons used their own softmax activation function with multiclasses and multiple categories being probabilities output.

The model was compiled using the Adam optimizer on the suggestions of an adaptive learning rate. The categorical cross entropy loss function was used, as with every multi-class classification problem. I used accuracy as a metric to evaluate how it behaved during training.

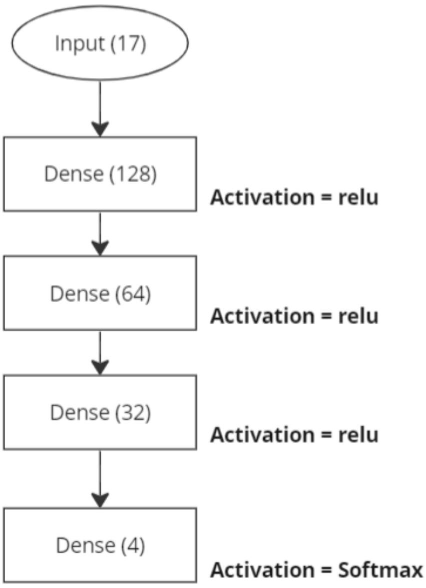


Figure 0.5. ANN architecture for WSNBFSF dataset.

The ANN model’s architecture for the KDD dataset is slightly different due to the nature of the dataset. An input layer with 41 neurons is added, which corresponds to the 41 features in the dataset, and a hidden layer with 64 neurons is included(Figure 3.6). Additionally, I also added a dropout layer with a 50% rate to prevent overfitting, randomly omitting half of the generated feature detectors on each pass used to train the model.

This makes the model more resilient to unknown data. Finally, the output layer of this network contains 2 neurons and also uses the softmax activation function, making it suitable for binary classification.

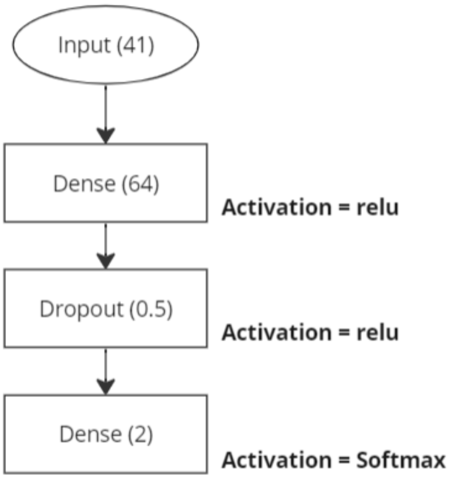


Figure 3.6. ANN architecture for KDDdataset.

Because we are using two completely different dataset, We train both models using the same optimizer and loss function to ensure they adopt a similar approach to learning."However, it is altered to adjust the diverse characteristics of the two datasets and enable a more complex classification of the network traffic to detect anomalies in the wireless sensor networks and identify intruders in typical network environments.

3.2.3.2. Long Short-Term Memory (LSTM)

One of the strategic areas of our work is the Long Short-Term Memory model, which we adopt to capture temporal dependencies and sequential patterns in network traffic data. These aspects are critical for accurate anomaly and intrusion detection.

Figure 3.7 shows the construction of the LSTM model for the WSNBFSF dataset. The LSTM model's building comes with a sequential model architecture that starts with an LSTM layer of 100 units . Additionally, this layer is set to receive input sequences of shape (1, 17), with '1' representing the time-step and '17' being the features.

Both LSTM layers adapted an activation function 'relu' and L2 regularization, which penalized the weights during training to mitigate overfitting. The 'return_sequences' parameter was set to 'True' so that the outputs' sequence can be further directed to the following layer to aid the model in capturing the sequential information imposed on the feature data. Next is a dense layer that has 50 neurons and ‘relu’ activation; it also employs L2 regularization. The dense layer is followed by a softmax output layer utilizing four units corresponding to the four different traffic classes.

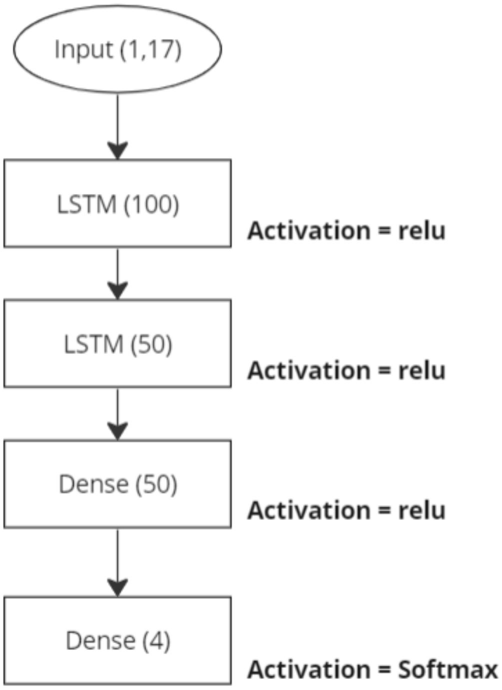


Figure 3.7. LSTM architecture for WSNBFSF dataset.

For the KDD dataset case, the LSTM model is similar to the WSN&BFSF one, differing only the shape of input data that changes to (1, 41) to account for KDD’s 41 features and shown in (Figure 3.8).

This model comprises two LSTM layers that are eventually followed by the second dense layer using L2 regularization to prevent overfitting, and ends with one softmax output layer with 2 units to classify the binary ‘normal’ and ‘anomalous’ traffic.

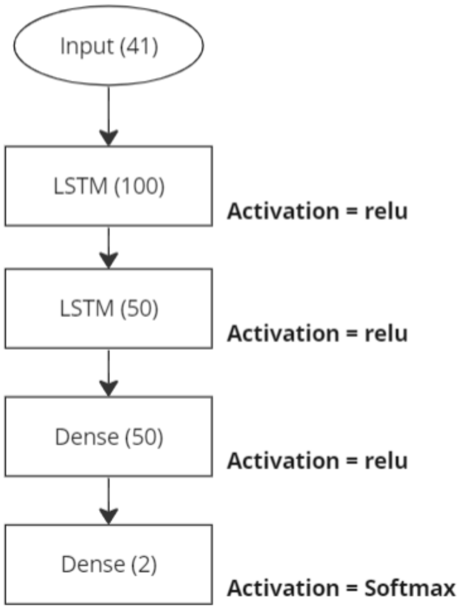


Figure 0.8. LSTM architecture for KDDdataset.

Both LSTM models were compiled with the Adam optimizer, using a categorical crossentropy loss function to allow multi-class classification. Since one of our metrics was accuracy, the inclusion of the reporting metric facilitated a modulo by reducing the ratio of correctly classified instances during the training and validation routines. We also applied L2 regularization to all LSTM and dense unit attempts. Therefore, we enforced some regularity on the model's weights and simplified some aspects of the model's weights, which then stretched the possibility of the learned patterns.

3.2.3.3. Bidirectional Long Short-Term Memory (BiLSTM)

In this work, to take advantage of both the past and future context, we incorporated the Bidirectional Long Short-Term Memory model in the BiLSTM. This is of particular importance for the challenging task of anomaly and attack detection on network traffic.

Starting with the WSNBFSF dataset the model architecture takes a bidirectional LSTM with 100 neurons to process sequences in both the forward and reverse direction as shown in (Figure 3.9). This layer is set to return the full sequence output to the next layer, which is used to maintain the temporal sequence of the characteristics needed to detect complex patterns. The second layer that follows is a bidirectional LSTM with

50 neurons. Both LSTM layers use the ‘relu’ activation and an L2 regularization was thus used to prevent the overfitting problem. Subsequently, there are three dense layers that take 50 neurons each. This set of layers maintain the ‘relu’ activation and L2 regularization. The final layer has a softmax output with four neurons, which are the desired four class of the traffic data which therefore makes the model output a probability distribution over the four categories.

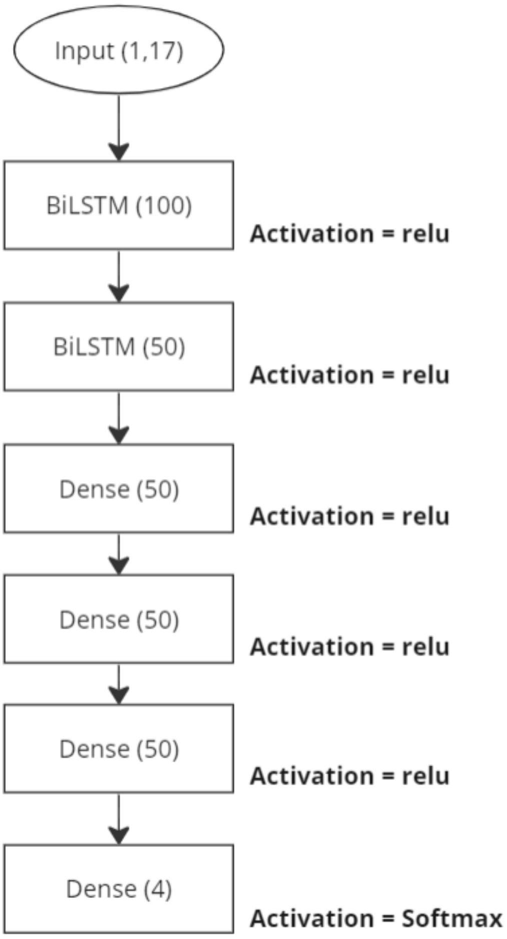


Figure 3.9. BiLSTMarchitecture for WSNBFSF dataset.

The KDD dataset uses a BiLSTM architecture, which is similar to the setup used in the WSNBFSF model, but it's tailored for binary classification tasks. This means the model concludes its process with an output layer that has two neurons. These neurons make use of the softmax activation function to categorize inputs into one of two classes.This structure suits the binary nature of the dataset, where the objective is to distinguish between 'normal' and 'anomaly' classes(Figure 3.10).

Both models are compiled with the Adam optimizer, famed for both the efficiency of computation and the adaptiveness of the learning rate, and the categorical crossentropy loss function, suitable for multi-class classification problems. Then, the ‘accuracy’ metric is introduced that allows an intuitively meaningful understanding of the model’s performance as the proportion of samples in the training-set correctly classified.

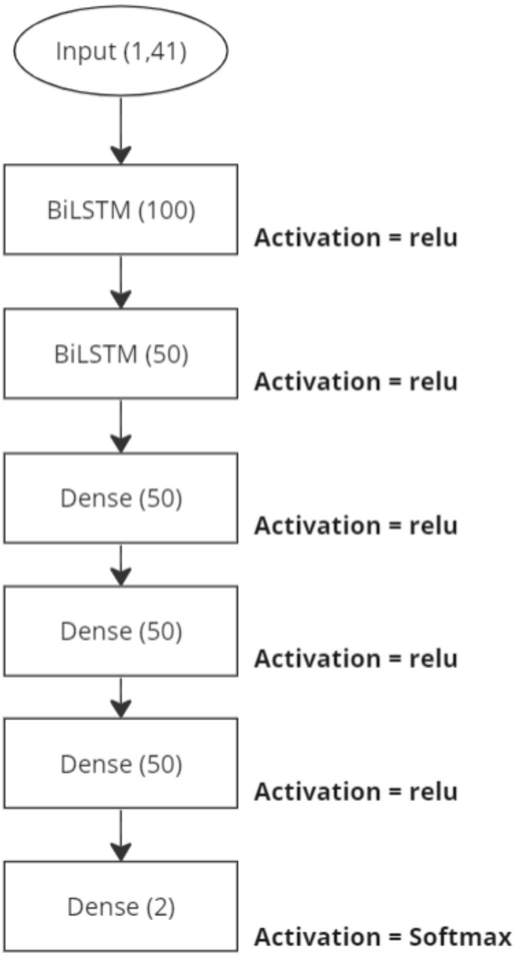


Figure 3.10. BiLSTMarchitectureforKDDdataset.

The repeated application of L2 regularization across both LSTM and dense layers in the BiLSTM models addresses the complexity that often comes with bidirectional processing, helping to ensure that the models do not overfit to the training data and can generalize well to new, unseen data. This bidirectional approach aims to encapsulate the temporal dependencies within the network traffic data more comprehensively, potentially leading to enhanced predictive performance in identifying network anomalies and attacks.

3.2.3.4. Convolutional Neural Network (CNN)

In our research, the Convolutional Neural Network (CNN) model is adapted to process one-dimensional sequence data, providing an innovative approach to analyzing network traffic for the WSNBFSF and KDD datasets.

For the WSNBFSF dataset, the CNN model, denoted by model_conv1d, has three convolutional layers having one-dimensional kernels as presented in Figure (3.11). The first convolutional layer consists of 128 filters, and the subsequent two are 64 filters, all with a kernel size 3. They have an activation function of ‘relu’ and utilize L2 regularization to combat overfitting.

'Same' padding is used to ensure the output of the convolutional layers has the same length as the input, preserving the full information of the sequence. After convolutional layers, the model flattens the data to pass it to the dense layers, where it has a progression of neurons numbered at 100 and 50, both applying 'relu' activation and L2 regularization. The final layer is a softmax activation layer with 4 output units, each representing a class of network traffic, allowing for a probabilistic classification.

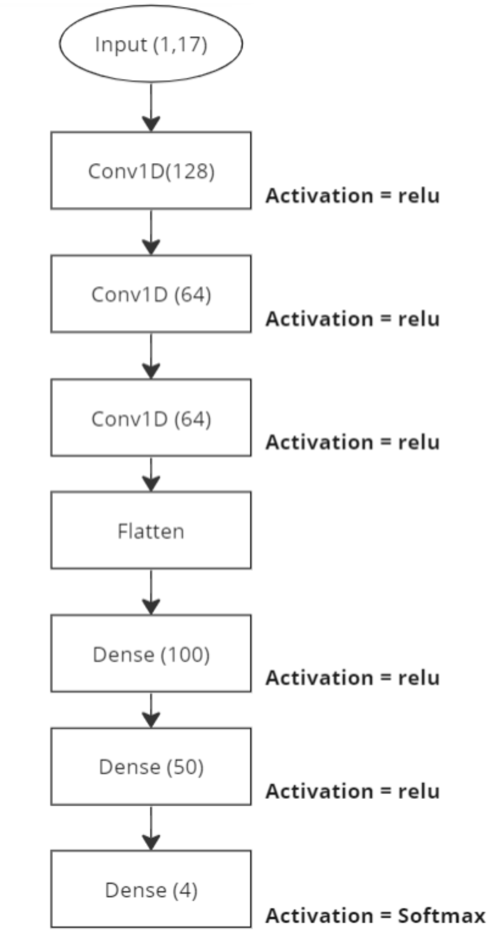


Figure 0.11. CNN architecture for WSNBFSF dataset.

For the KDD dataset, the `model_conv1d` is similarly composed, starting with a convolutional layer of 128 filters, followed by two layers of 64 filters, and using the same padding, activation, and regularization strategies. The flattened output is then fed into two dense layers with 'relu' activation and L2 regularization, followed by a softmax output layer with 2 units appropriate for binary classification (Figure 3.12).

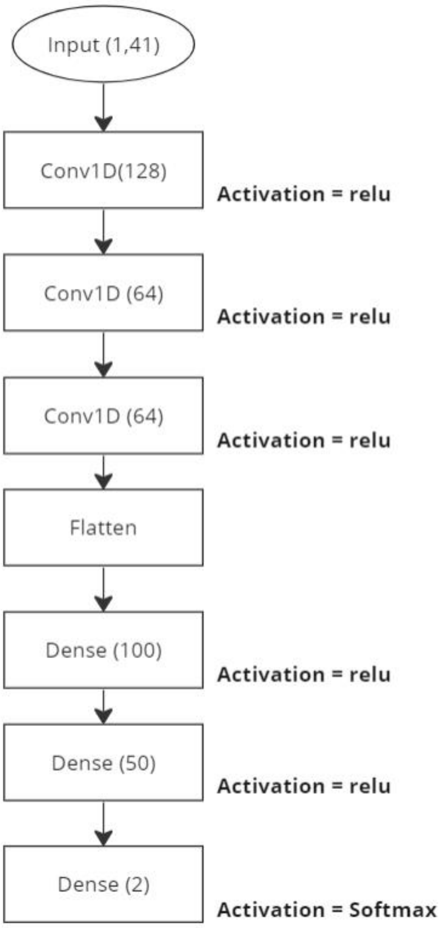


Figure 3.12. CNNarchitecture forKDDdataset.

Both models are compiled with the Adam optimizer and the categorical crossentropy loss function to tune the weights based on the data's multi-class nature. The models are trained over several epochs with batch processing, and a portion of the training data is set aside as a validation set to monitor the model's performance and prevent overfitting.

The primary objective of developing these CNN models is to leverage the spatial feature hierarchy of one-dimensional data to extract low-level and high-level representations of network traffic. However, the actual training pipeline focuses on making a model more sensitive to the distinctions between normal traffic and multiple types of network attacks. One of the central strengths of this architecture is its focus on local dependencies and patterns, which allows for the identification of intricate anomalies in network behavior.

3.2.4. Evaluation Measures

Evaluation measures are certain metrics that are used to evaluate how classification models work. They indicate how a model could realize its prediction task and are calculated from the confusion matrix, which indicates how many correct and incorrect predictions were made in each category.

Confusion Matrix: each row represents the instances of an actual class, and each column : represents predicted instances for a class. It consists of the following:

- True positive (TP) : instances of positive that were correctly predicted
- True Negatives (TN): Correctly predicted negative observations.
- False Positives (FP): Incorrectly predicted positive observations (also known as Type I error).
- False Negatives (FN): Incorrectly predicted negative observations (also known as Type II error).

Accuracy: This measures the ratio of correctly predicted observations to the total observations. It is suitable when the class distribution is similar. The equation is given by:

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

Precision: Also known as the positive predictive value, this measures the ratio of correctly predicted positive observations to the total predicted positive observations. The equation is:

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

Recall (or Sensitivity or True Positive Rate): This measures the ratio of correctly predicted positive observations to all observations in actual class. The equation is:

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score: The F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. It is particularly useful when the class distribution is uneven. The F1 Score is calculated by:

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

Each of these measures provides different insights into the performance of a classification model. Accuracy can be deceiving in and of itself, especially when dealing with imbalanced datasets, it proves beneficial to combine Precision, Recall, and the F1 Score to paint a clearer picture of the model's effectiveness.

3.3. SUMMARY

In conclusion, the methodology section of this study has thoroughly described the roadmap of the research strategy. The presented methodology can be described as methodical and data-driven. We have ventured through the intricacies of dealing with vast amounts of data, each possessing a different set of challenges and nuances, to distill the raw data into a form palatable for learning via neural networks systematically. This meticulousness involving the steps of data preprocessing, model creation, and model evaluation is an excellent representation of the thoroughness required in modern cybersecurity research. Our usage of several neural networks spanning the VAE, LSTM, and denser networks shows the diverse means through which one can identify patterns of network behavior that identify the potentiality of threats and breaches in security. Remaining steps of data balancing, feature scaling, and model creation have all been designed meticulously to work in unison toward the ultimate goal of promoting anomaly detection in cybersecurity.

PART 4

EXPERIMENTS AND RESULTS

4.1. INTRODUCTION

Chapter 4, “Experiments and Results” breaks down the empirical analysis undertaken to test the efficiency of various deep learning models on two different datasets: WSNBFSF and KDD. This chapter is divided into two major parts. Firstly, it is an experimental setting where the readings are tested. In the subsequent sections, the outcomes from utilizing Artificial Neural Network , LSTM, Bidirectional LSTM , and CNN on the WSNBFSF and KDD datasets are elaborated . These sections were discussed utilizing different performance statistics like accuracy, precision, recall and loss. A comparative analysis of the usage of the same model between two datasets is given. This shall be followed by a comparison among different datasets. The results end using WSNBFSF and KDD, with a recapitulation. The summary section integrates the resulting findings and discusses their relevance. This involves a discussion of the capacities of the WSNBFSF and KDD datasets to solve these issues.

4.2. EXPERIMENTAL SETTING

In this section, we discuss the technical as well as the operational details of the environment and tools used in conducting the experiments. Python is the primary programming language because Python comes with rich libraries and frameworks that are suitable for machine learning. Google Colab was used for this experiment, which is a cloud-based platform that allows easy access to high-level computational resources, such as GPUs and TPUs, and allows better collaboration. Moreover, Google Colab is a better platform for the complex deep learning models that have large training data because python has the vast support of diverse libraries, while Google Colab is an easy, flexible platform that can be easily scaled. The probable aspects of this section will probably be the configurations, libraries, and versions used for the study. This information helps replicate results and insights into the resources used in the computational study.

4.3. RESULTS ON WSNBFSF DATASET

4.3.1. Results of ANN Model

The learning curve of the ANN model on the WSNBFSF dataset in the course of 30 epochs presented impressive improvement. In particular, at epoch 1, the model presented an accuracy of 75.60% on the training dataset, which gradually increased. Therefore, the learning curve is good. The validation accuracy started from 79.75% and also gradually increased. Therefore, the model was able to generalize well.

In particular, after 30 epochs, the training accuracy was 92.42% and validation accuracy was 91.80%. The corresponding loss metrics also indicate a positive trend, with the training loss decreasing from an initial 0.6174 to 0.1860, and the validation loss from 0.4790 to 0.2022 by the final epoch.

The convergence of the training and validation accuracy, as depicted in the left of Figure 4.1, signals a harmonious balance in the model's ability to learn from the training data and its performance on the validation set. A slight divergence seen in the initial epochs quickly resolves, leading to a parallel increase which suggests that the model is not memorizing the training data but truly learning the distinguishing features of the traffic types.

The loss metrics, shown on the right, reinforce this notion; both the training and validation loss demonstrate a steady decline, plateauing towards the later epochs. This is indicative of the model reaching its potential in minimizing the classification error. Notably, the validation loss remains closely aligned with the training loss throughout the epochs, a sign that the model is not overfitting the data.

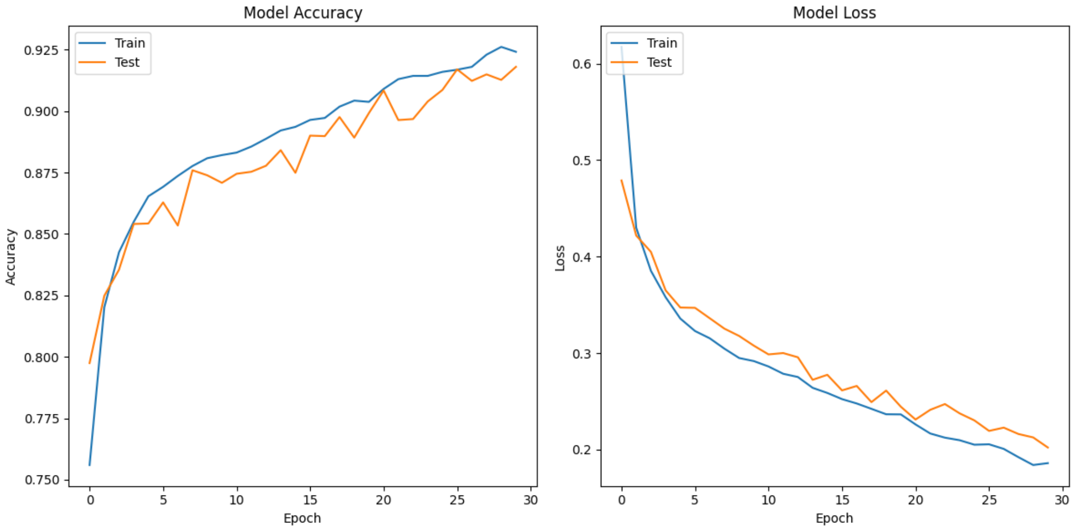


Figure 4.1. Training curves of ANN.

The confusion matrix for the ANN model applied to the WSNBFSF dataset provides a detailed visualization of the model's classification accuracy across different traffic types. The matrix shows the number of correct and incorrect predictions juxtaposed against the actual labels, with each cell representing the counts of predictions for every true label (Figure 4.2).

Looking at the matrix, the model demonstrates a high degree of accuracy for 'Blackhole' attacks, correctly identifying 1,337 instances while misclassifying only a small number as 'Flooding' (61) and 'Normal' (45). Similarly, the model is proficient at detecting 'Flooding' attacks with 1,514 correct predictions, although it did misclassify 34 instances as 'Blackhole'.

For 'Forwarding' attacks, the model performs exceptionally well, with 1,581 correct classifications and only 5 instances mislabeled as 'Normal'. This indicates a strong model capability to distinguish 'Forwarding' attacks from other types.

However, the model shows some challenges in correctly classifying 'Normal' traffic, with 215 instances misclassified as 'Blackhole' and 57 as 'Forwarding', which suggests a tendency of the model to falsely predict normal behavior as malicious to some extent.

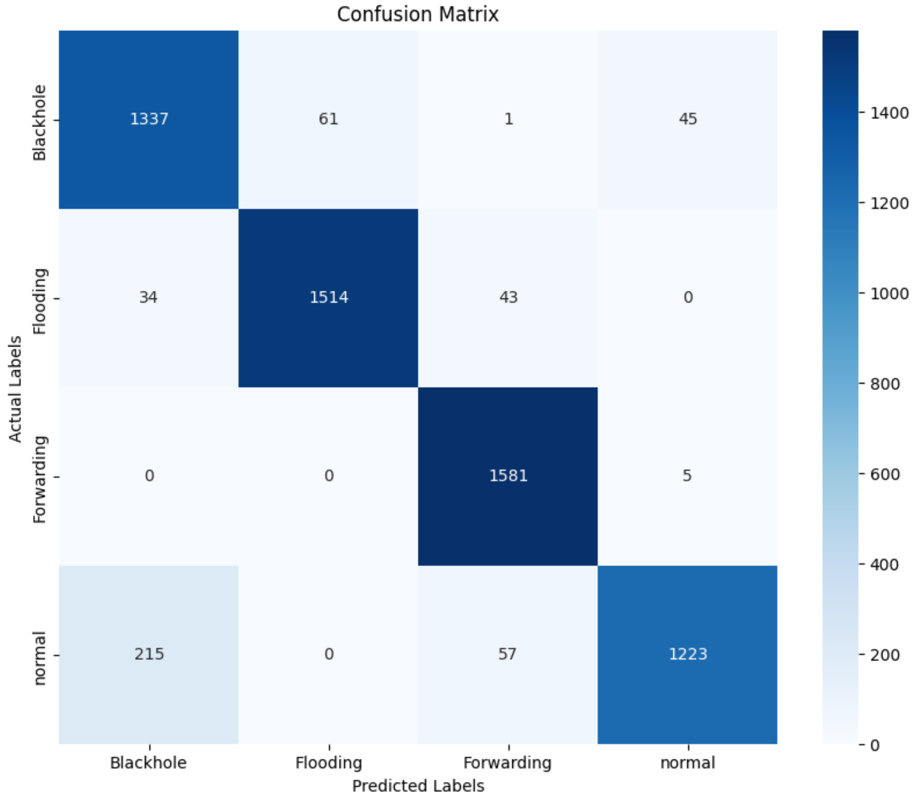
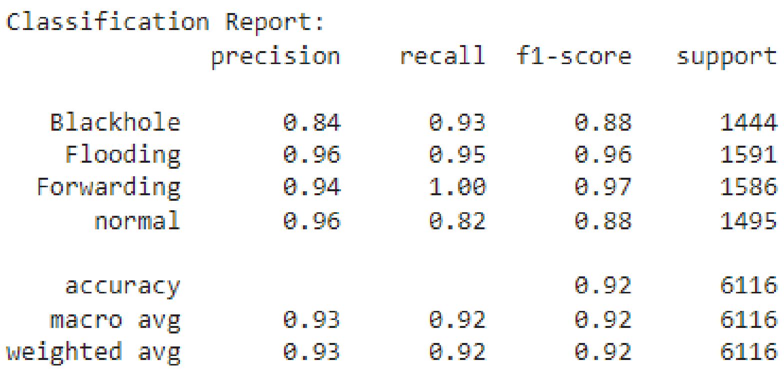


Figure 0.2. CM of ANN.

The results for the ANN model applied to the WSNBFSF dataset reflect a high degree of accuracy in classifying different network traffic types as shown in Figure 4.3. The model showcases strong precision across the board, particularly in identifying 'Flooding' and 'normal' traffic with precision scores of 0.96 each, suggesting a high likelihood that predicted positives are true positives. The recall for 'Forwarding' attacks is perfect at 1.00, indicating the model successfully identified all actual 'Forwarding' attack instances. Notably, the model also achieves a commendable balance of precision and recall (as indicated by the f1-scores) for 'Blackhole' and 'normal' traffic, with f1-scores of 0.88 for both, pointing to a well-rounded performance.

'Flooding' and 'Forwarding' types are recognized with impressive f1-scores of 0.96 and 0.97, respectively, indicating a strong harmonic mean of precision and recall. The overall accuracy of the model stands at 0.92, which is a robust indicator of its overall ability to correctly classify traffic types on a consistent basis. The macro and weighted average for precision, recall, and f1-score is 0.92, which indicated the classification capability range over different classes. The support column in the classification report demonstrates the actual number of occurrences of each label, which portrays how the model’s performance measure variant compared to the size of the dataset.



```
Classification Report:
              precision    recall  f1-score   support

   Blackhole       0.84      0.93      0.88      1444
    Flooding       0.96      0.95      0.96      1591
  Forwarding       0.94      1.00      0.97      1586
      normal       0.96      0.82      0.88      1495

    accuracy                           0.92      6116
   macro avg       0.93      0.92      0.92      6116
weighted avg       0.93      0.92      0.92      6116
```

Figure 0.3. Classification report of ANN.

4.3.2. Results of LSTM Model

The performance of the LSTM model applied to the WSNBFSF dataset indicates a well-defined learning curve, as demonstrated by the pattern in the accuracy and loss graphs provided in Figure 4.4. From epoch 1, the model reveals a training accuracy of about 69.39%, which increases slowly to 91.96% at epoch 30. The relatively smooth motion of the training accuracy indicates that the model correctly recognizes the dynamics of the observations in the data. The validation accuracy equally increases slowly from 74.13% to 91.70% within the same learning period. This relatively smooth incline in the validation accuracy, which follows that of the training accuracy, indicates that the model captures well-generalizable features.

In the loss graphs, the model exhibits a swift decrease in loss for both training and validation sets, starting high at the first epoch and then plummeting rapidly, which is characteristic of LSTM networks' ability to quickly reduce error during the initial learning phase. The training loss decreases from 0.8841 to 0.2960, while the validation loss mirrors this trend, dropping from 0.6584 to 0.2996 by the end of the training process. The close convergence of training and validation loss by the 30th epoch further underscores the model's balanced fit.

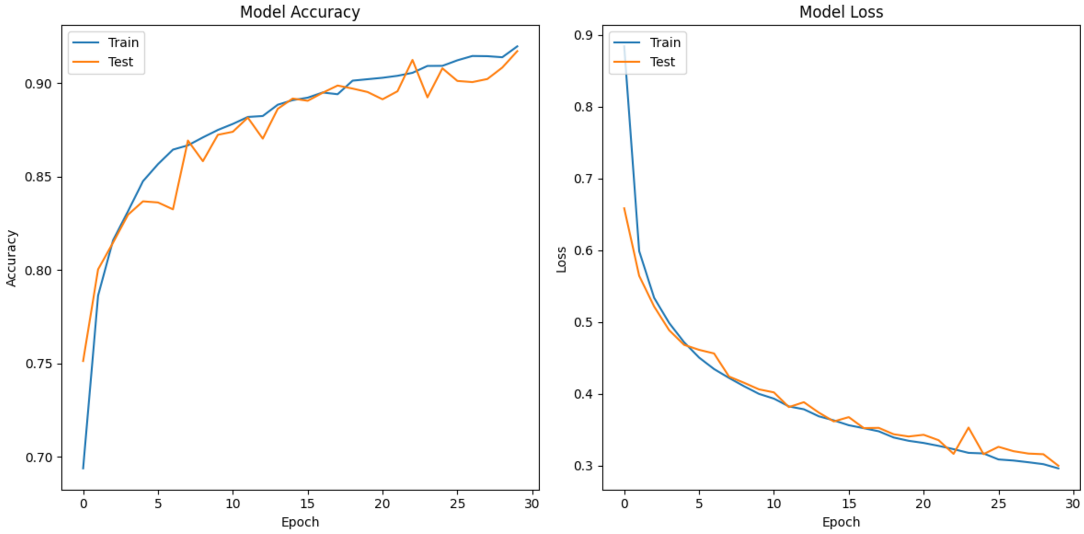


Figure 0.4. Training curves of LSTM.

The confusion matrix for the LSTM model applied to the WSNBFSF dataset illustrates a promising performance across all classes of network traffic (Figure 4.5). For 'Blackhole' attacks, the model achieves a high level of precision with 1340 true positives and a minimal number of false negatives, indicating strong sensitivity in detecting this type of attack. However, there are 78 instances where 'Blackhole' is mistaken for 'Flooding', suggesting some confusion between these two classes.

The model shows remarkable accuracy in identifying 'Flooding' attacks, with 1527 true positives and only 22 instances incorrectly labeled as 'Blackhole'. This demonstrates the LSTM's capability in distinguishing 'Flooding' attacks with high reliability.

For 'Forwarding' attacks, the LSTM model achieves near-perfect detection with 1567 true positives, signifying an excellent true positive rate. A minor confusion is seen with 17 instances misclassified as 'Flooding', which may require further analysis to understand the overlap between these attack patterns.

The performance on normal traffic, while still strong, exhibits some room for improvement, with 1225 true positives against 225 instances misclassified as 'Blackhole', indicating a tendency of the model to be overly cautious, potentially flagging normal behavior as malicious.

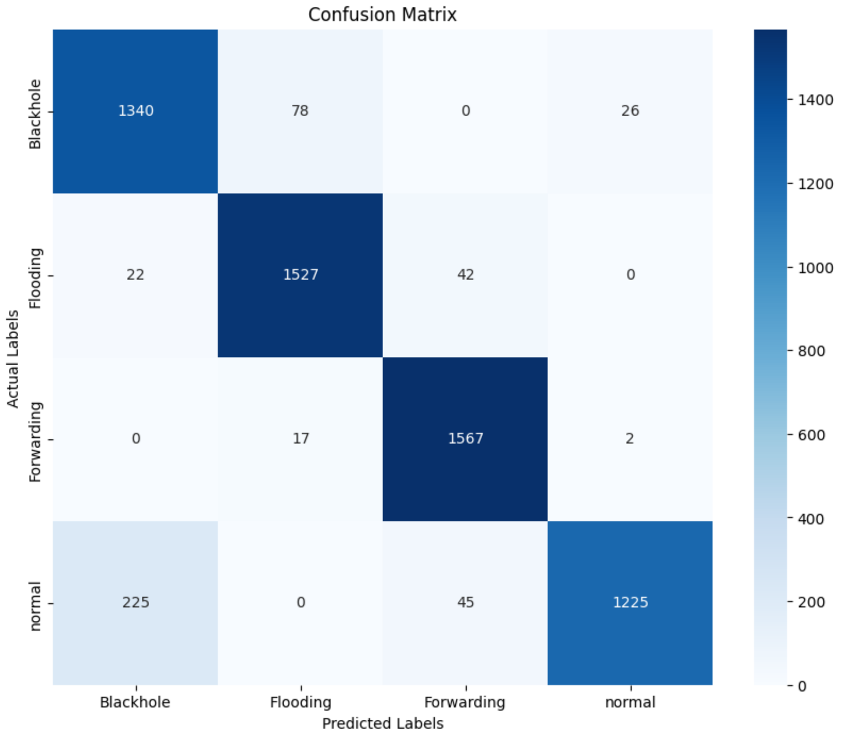
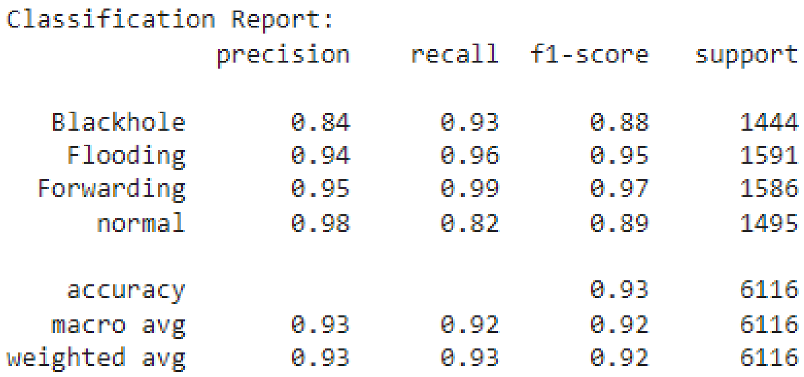


Figure 0.5. CM of LSTM.

The LSTM model's classification report for the WSNBFSF dataset exhibits high precision, recall, and F1-scores across all categories(Figure 4.6). 'Blackhole' detection is precise (0.84) and has a high recall (0.93), resulting in a solid F1-score (0.88). 'Flooding' attacks are identified with even higher precision (0.94) and recall (0.96), reflected in an excellent F1-score (0.95). 'Forwarding' attacks see the best performance with an outstanding precision of 0.95, almost perfect recall of 0.99, and an F1-score of 0.97. Normal traffic detection is the most precise at 0.98, but with a lower recall of 0.82, it has an F1-score of 0.89.

The overall accuracy of the LSTM model stands at an impressive 92.53%, demonstrating its strong capability to classify different types of network traffic effectively. The macro and weighted averages for precision, recall, and the F1-score all hover around the 0.92 to 0.93 range, signifying a balanced performance across the classes despite variations in class distribution.

This performance, combined with a test loss of 0.2859, confirms that the LSTM model is both accurate and robust in predicting network traffic types, managing to balance the trade-off between precision and recall effectively. The high-test accuracy underlines the LSTM's ability to capture temporal dependencies and sequence patterns essential for anomaly and intrusion detection in network traffic data.



```
Classification Report:
              precision    recall  f1-score   support

   Blackhole       0.84      0.93      0.88      1444
    Flooding       0.94      0.96      0.95      1591
  Forwarding       0.95      0.99      0.97      1586
      normal       0.98      0.82      0.89      1495

    accuracy                           0.93      6116
   macro avg       0.93      0.92      0.92      6116
weighted avg       0.93      0.93      0.92      6116
```

Figure 0.6. Classification report of LSTM.

4.3.3. Results of BiLSTM Model

The performance of the BiLSTM model on the WSNBFSF dataset over 30 epochs demonstrates substantial learning and generalization capabilities(Figure 4.7). The model begins with a lower accuracy of 67.24% on the training set and a validation accuracy of 76.35%, signifying initial learning stages. As epochs progress, both training and validation accuracies improve, indicating that the model is effectively capturing the underlying patterns in the data without overfitting, as evidenced by the convergence of training and validation lines. The accuracy peaks at 93.54% for training and 91.80% for validation, while the test accuracy reaches a notable 93.00%, which suggests that the model has achieved a commendable level of predictive power. This is further reinforced by the decreasing trend of the loss curves for both training and validation, with the model experiencing a test loss of approximately 0.276, pointing to a good fit. The consistent improvement and stabilization of accuracy and loss metrics over epochs underscore the model's robustness in classifying network traffic types effectively.

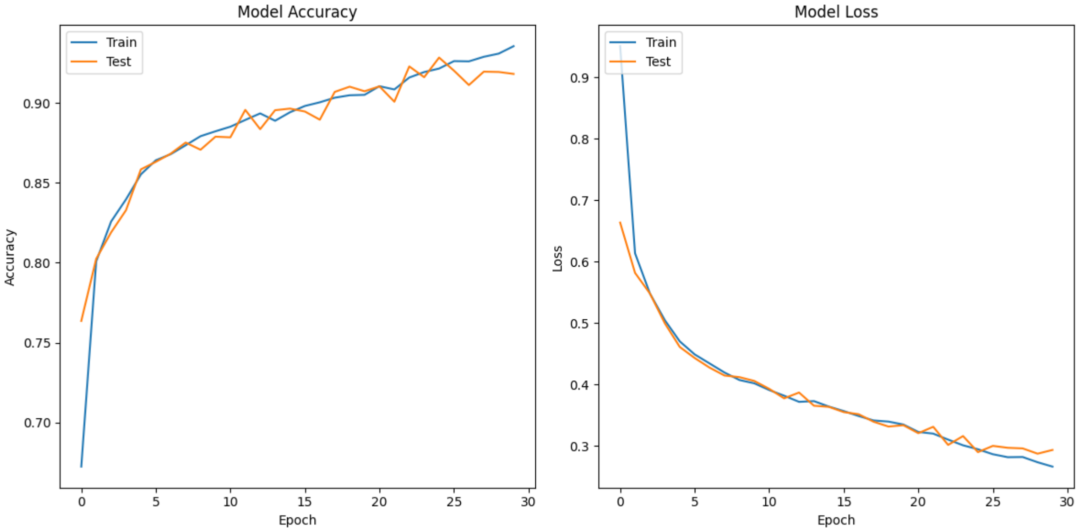


Figure 4.7. Training curves of BiLSTM.

The confusion matrix in Figure 4.8 for the BiLSTM model applied to the WSNBFSF dataset presents an insightful depiction of the model's classification accuracy across different types of network traffic. For the Blackhole category, the model exhibits a high true positive rate, correctly identifying 1,175 instances, but also misclassifies 79 as Flooding and notably mislabels 190 as Normal, indicating a potential area for improvement. Flooding attacks are well-recognized with 1,528 true positives; however, there are still 58 instances mistaken as Forwarding, and 5 as Blackhole, suggesting a high precision but slightly less recall. Remarkably, the model achieves perfect classification for the Forwarding category, with all 1,586 instances correctly identified, displaying strong sensitivity towards this type of attack. The Normal traffic shows some confusion with 31 instances incorrectly labeled as Blackhole and 65 as Forwarding, but it successfully classifies 1,399 as Normal, which is quite robust. These results suggest that while the BiLSTM model is adept at distinguishing most of the attack types, there is a tendency to falsely categorize some attacks as Normal traffic, which is an aspect that could be further examined and optimized in future iterations.

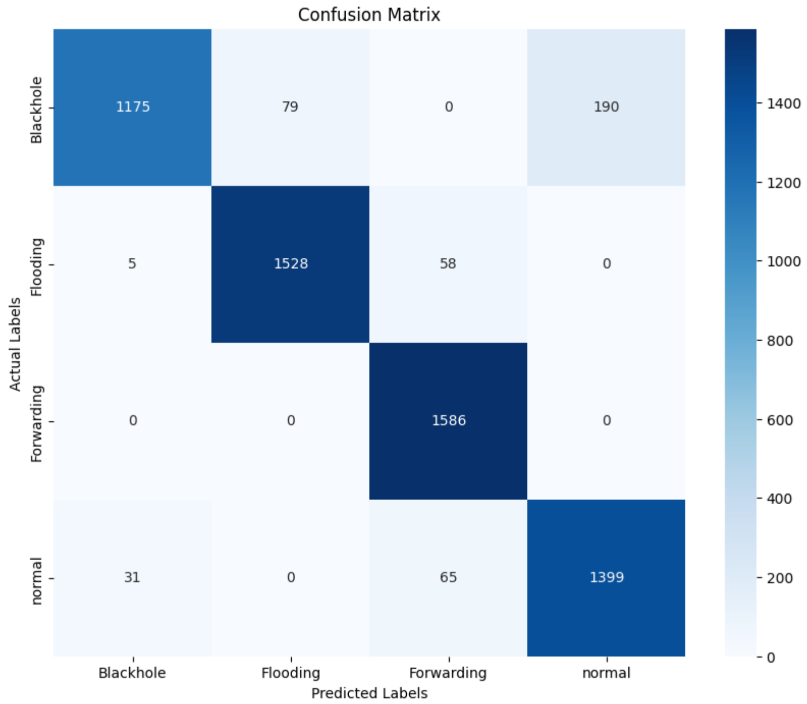
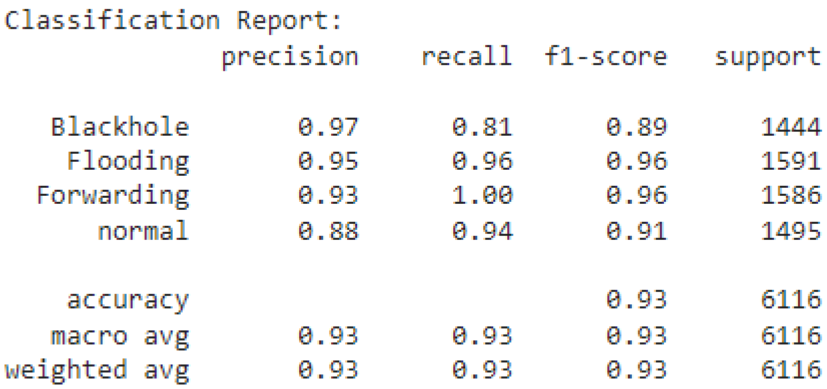


Figure 0.8. CM of BiLSTM.

The results for the BiLSTM model on the WSNBFSF dataset demonstrate strong performance across various metrics, showcasing the model's proficiency in classifying network traffic (Figure 4.9). With precision scores ranging from 0.88 for normal traffic to 0.97 for Blackhole attack traffic, the model exhibits high accuracy in identifying true positives for each category. Recall scores are also impressive, with a perfect score of 1.00 for Forwarding attacks, indicating that all instances of this attack type were correctly identified.

The f1-score, which is a balanced measure combining precision and recall, is consistently high across all classes, peaking at 0.96 for both Flooding and Forwarding attacks, signifying an excellent balance between precision and recall for these types. The support values reflect the number of actual occurrences of each class in the dataset, allowing for an understanding of the distribution and volume of each traffic type.

The model achieves an accuracy of 0.93, and the macro and weighted averages for precision, recall, and f1-score are all congruent at 0.93, indicating that the model is both accurate and consistent in its predictions across the different classes.



```
Classification Report:
              precision    recall  f1-score   support

   Blackhole       0.97      0.81      0.89      1444
    Flooding       0.95      0.96      0.96      1591
  Forwarding       0.93      1.00      0.96      1586
      normal       0.88      0.94      0.91      1495

    accuracy                           0.93      6116
   macro avg       0.93      0.93      0.93      6116
weighted avg       0.93      0.93      0.93      6116
```

Figure 0.9. Classification report of BiLSTM.

4.3.4. Results of CNN Model

The CNN model's training on the WSNBFSF dataset over 30 epochs has yielded impressive outcomes(Figure 4.10). Initial steps began with a loss of 0.7827 and an accuracy of 73.89%, which through consistent learning, the model improved to a final training loss of 0.2379 and an accuracy of 93.82%. On the validation side, the model commenced with a loss of 0.5789 and an accuracy of 80.11%, reaching a noteworthy validation loss of 0.2302 and an accuracy of 94.01% by the last epoch.

These results reveal the model's capability to adeptly learn and distinguish between various types of network traffic, including both normal behavior and malicious threats. The significant convergence seen in the loss graphs and the parallel rise in accuracy scores between the training and testing datasets denote that the model is not overfitting. This model is generalizing well to unseen data, which is crucial in its real-world application in the detection of network intrusions. The observed performance, especially in the final epochs, suggests that the CNN model is capable of classifying network traffic accurately. It could, therefore, be applied to cybersecurity to identify any anomalies in network behaviors.

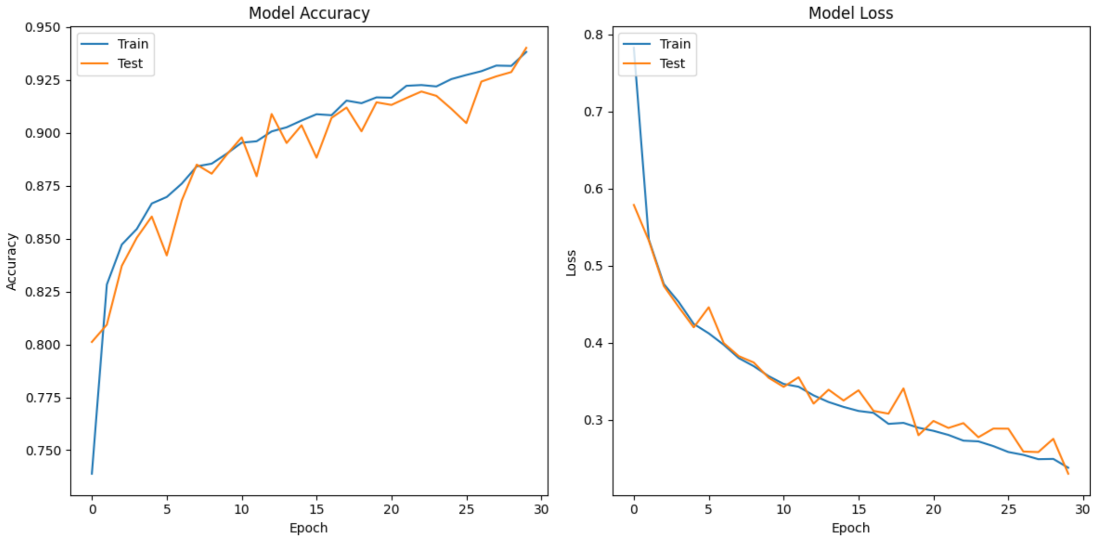


Figure 0.10. Training curves of CNN.

Figure 4.11 below illustrates the Confusion Matrix for the CNN model of the WSNBFSF dataset. The matrix gives insights into the model's classification performance based on various traffic types: Blackhole, Flooding, Forwarding, and Normal. The most notable observation from the matrix is the significantly high values along the diagonal grid: 'Blackhole' has 1377 true positives, Flooding has 1554, Forwarding has 1583, and Normal has 1303. This shows that the model is highly competent in accurately identifying most instances within the majority of these categories. As presented in the off-diagonal grid, the misclassification values are relatively low. The matrix indicates that 'Normal' is the most misclassified traffic type, with 151 instances misclassified as 'Blackhole' and 41 as 'Forwarding'. 'Flooding' also has 19 instances misclassified as 'Blackhole' and 18 as 'Forwarding', although the numbers are relatively small compared to the correctly predicted instances. In essence, the absence of misclassification occurs at Forwarding as Flooding and Flooding as Forwarding, which means that the model clearly learned the difference in these attack types. It is confirmed by the dominance of diagonals across the matrix axis, indicating the model's confidence in its capacity of classifying different traffic patterns.

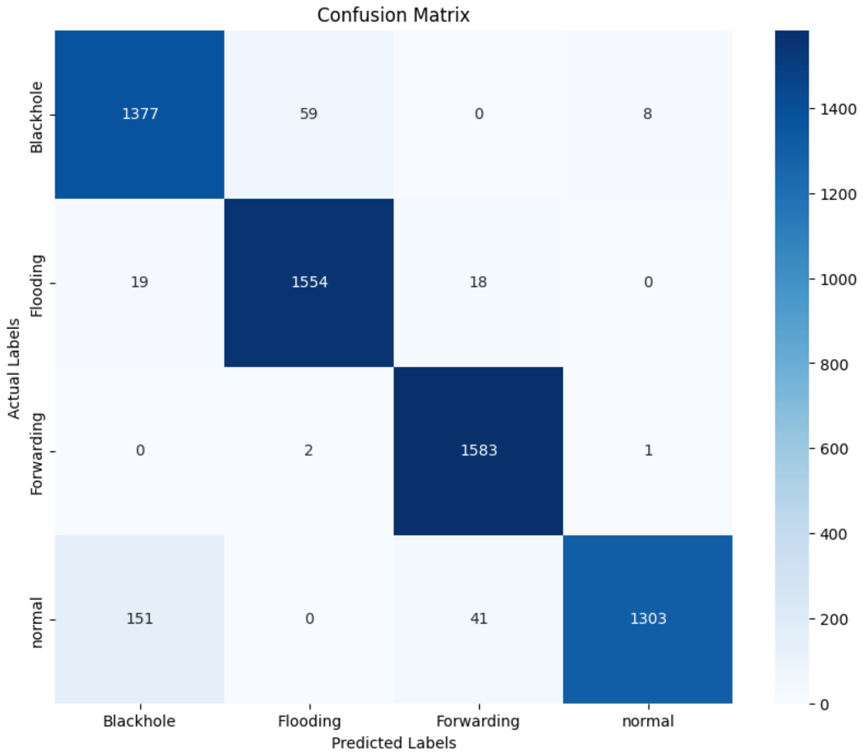


Figure 0.11. CM of CNN.

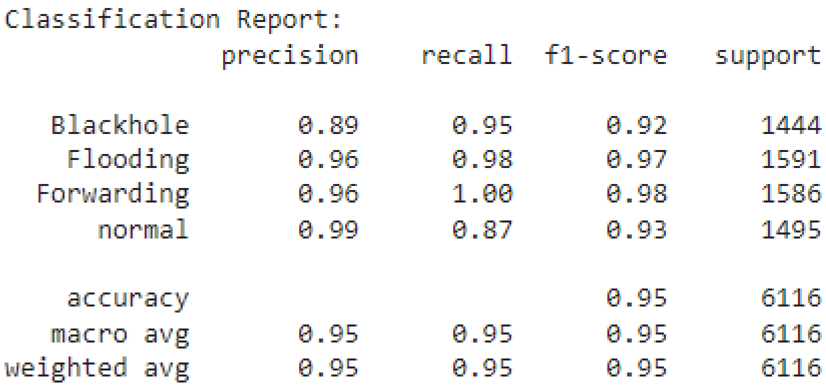
The CNN model being evaluated on the WSNBFSF dataset exhibits excellent performance, as demonstrated by the classification report. The precision scores, which show the model's ability to accurately retrieve positive instances correctly identified among all identifiable instances, are as impressive. High in all classes, the 'Normal' instance attains the highest score with 0.99 precision. This outcome suggests the model's improved capability of identifying normal traffic away from any potential threats.

The recall scores, which indicate the model's ability to retrieve all the accurate positive instances, are equally outstanding. 'Forwarding' attains a perfect score as 'Flooding' receives a 0.98, followed by Blackhole at 0.95 and 'Normal' at 0.87, the latter being slightly lower but robust.

There is a high F1-score, the harmonic mean of both precision and recall, in all classes, signaling a consistent performance between precision and recall. This balance is essential in the network security domain due to the numerous implications of false negatives and false positives.

The general accuracy level was 0.95, which is quite impressive. The measure implies that the model predicts the class label correctly in 95% of the test set instances. Therefore, with this accuracy level and the precision, recall and F1-scores presented in their respective individual precisions show CNN models are indeed effective in classifying various instances of network traffic and can, therefore, be used as a vital tool in identifying and responding to diverse network security threats.

The test loss reported at 0.2102 also corroborates the accuracy metrics. The reported loss shows that the model's predictions were on average closed to actual class labels and, therefore, portray a reliable capability of generalizing outside the training data. The comprehensive performance indicators demonstrate the CNN model potential for use in actual network intrusion detection applications.



```
Classification Report:
              precision    recall  f1-score   support

   Blackhole       0.89      0.95      0.92      1444
    Flooding       0.96      0.98      0.97      1591
  Forwarding       0.96      1.00      0.98      1586
      normal       0.99      0.87      0.93      1495

    accuracy                           0.95      6116
   macro avg       0.95      0.95      0.95      6116
weighted avg       0.95      0.95      0.95      6116
```

Figure 0.12. Classification report of CNN.

4.3.5. Comparison

The deep learning models perform differently, and the results obtained by comparing Figures 4.13 reveal several insights regarding each architecture's performance. The CNN model is clearly the best performer with an accuracy of 0.95, demonstrating that it is very good at identifying the spatial hierarchies and features found in my dataset. The BiLSTM model is very close with an accuracy of 0.93, and this could imply that the model comprehends my sequence data well in both forward and reverse orders, helping it establish the required context. The LSTM is also at 0.93, equaling the BiLSTM. This indicates that my LSTM processing sequences is good, but without the help of backward information; the accuracy drops a Despite the model also having applications in unstructured sequence data like image and handwriting recognition, it has not shown any improvement due to the dataset or lack of back propagation. At 0.92, the ANN is still good, although the slightly lower accuracy shows that it is not very good at identifying the existing patterns. The loss of each model also shows how well each model reduces the quantity of errors during the learning period. However, it is surprising that the ANN should hold the lowest loss of 0.18, showing a better fit during training. The CNN, with my best accuracy, has the lowest loss of 0.21, which is also higher than for the LSTM and BiLSTM. The two models have an error rate of between 0.28 and 0.29, which is high compared to their good correction fit. This could be an indication that the models are good and are catching the normal patterns for accuracy, but they are not good at converging to the minimal error solution as CNN or ANN for my dataset.

These results collectively highlight that while CNNs might be the most accurate for the WSNBFSF Dataset, ANNs seem to have an edge in terms of model loss, indicating a potential for better generalization. The choice of model could thus be influenced by the specific requirements of the task, whether it is the highest predictive accuracy or the best generalization capability.

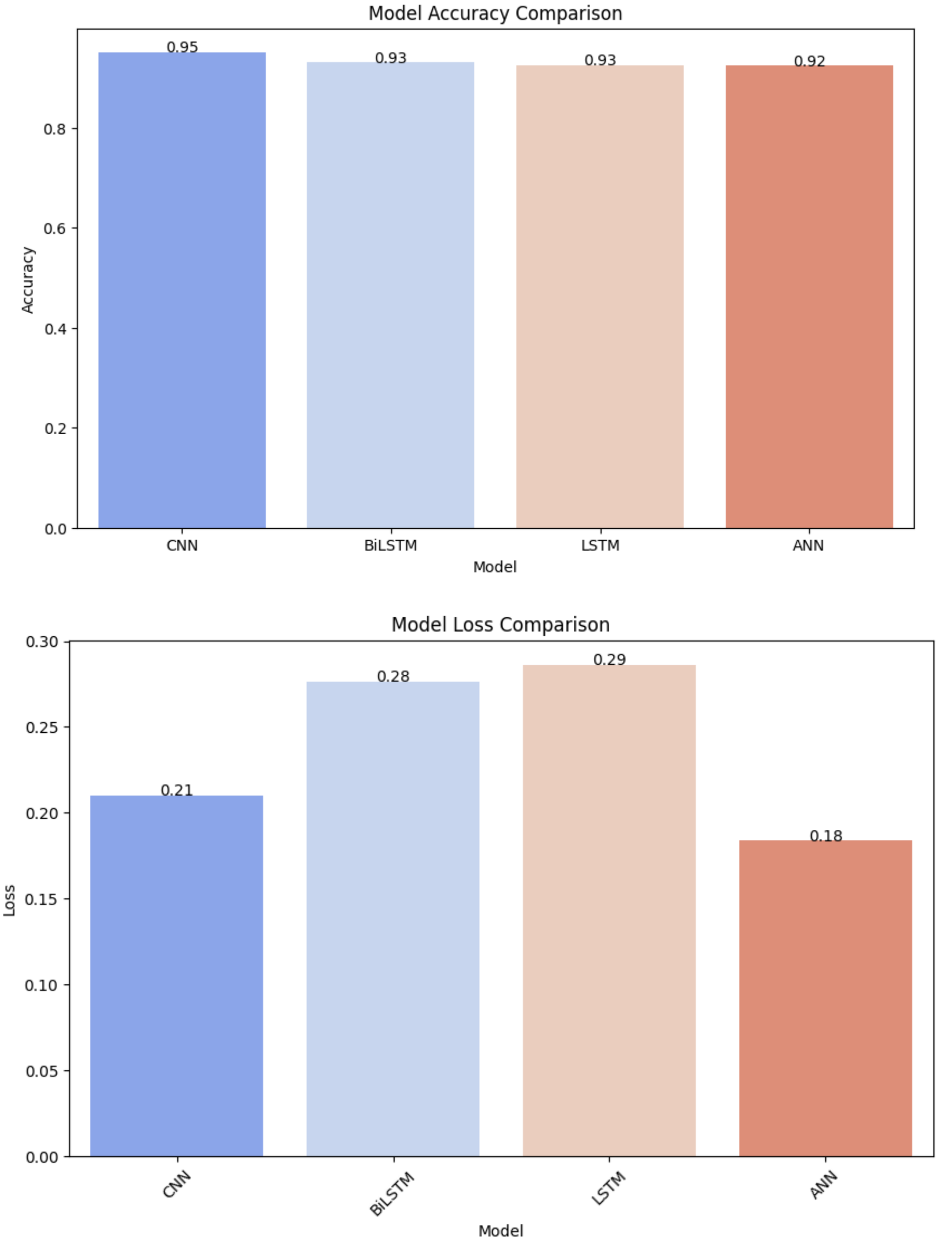


Figure 0.13. Model accuracy and loss comparison.

The performance of various deep learning models on the WSNBFSF Dataset is further elucidated through their Precision, Recall, and F1-Score metrics (Figure 4.14). The convolutional Neural Network (CNN) demonstrates an excellent balance across all three metrics with a constant score of 0.95. The equal levels of Precision, Recall, and F1-Score are an indicator that not only is the CNN highly accurate in its predictions, but also it holds a balanced trade-off for false positives and negatives while performing, which is the desired outcome for a reliable job in a security application.

The Bidirectional Long Short-Term Memory 0.93 and Long Short-Term Memory models display the same levels of all three indicators. Once again, these expressions signify the high ability of the models to capture temporal-based relations and contexts in the studied dataset, which are vital in spotting elaborate patterns of attack. Thus, the matching indicators of CNN and BiLSTM across all indicators used while considering the performance reveal a similar capability of predictive work in this dataset.

The artificial Neural Network (ANN) is slightly lagging in the results, resulting in similar Precision, Recall, and F1-Score of 0.92. The small gap from the ANN results as compared to other models indicates that despite being not as sensitive to the dataset's nuances, it is a competitive model for Anomaly-based network security.

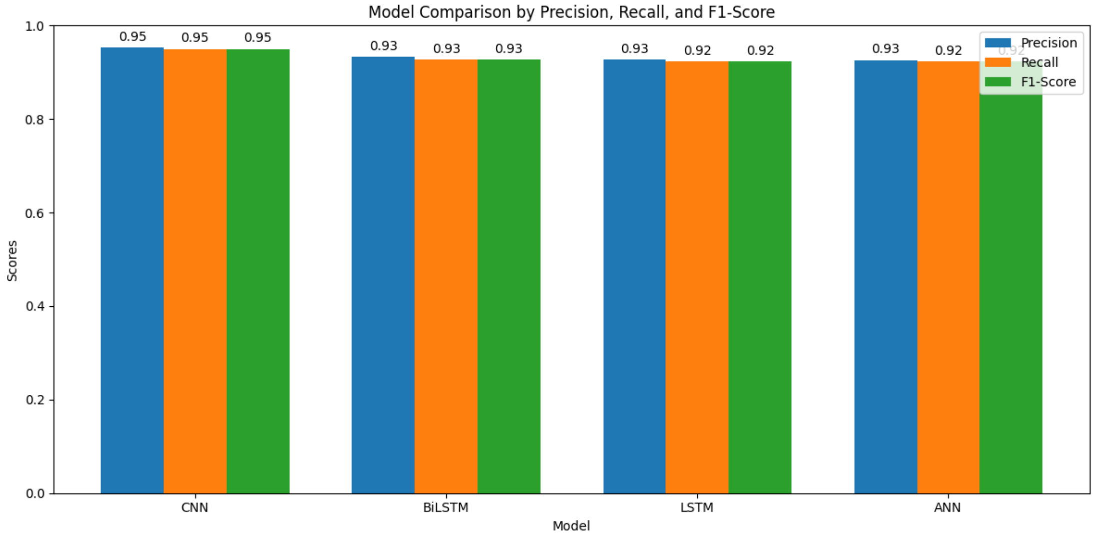


Figure 0.14. Performance of various deep learning models on the WSNBFSF Dataset through Precision, Recall, and F1-Score metrics.

Figure 4.15 shows a visual comparison of the performance metrics of the deep learning models on WSNBFSF Dataset. It is presented in shades of blue, from the paler ones indicating the lowest score to the darkest ones for the highest scores. CNN always achieves the highest scores uniformly in all five metrics: Accuracy, precision, recall, F1-score, and sensitivity at 0.95. This finding implies that CNN does not only classify the traffic accurately but also has a high true positive rate and correct ratio of true positive results over a true positive and false negative results.

The LSTM and BiLSTM models, which are identical in terms of all the performance metrics, indicate equal capability performance. Each scores 0.93, with the exception of LSTM, which receives a 0.91 in both recall and F1-score. These findings illustrate that the two models are good at handling sequential data, such as the one that occurs in network traffic time series analysis. Lastly, the ANN model performed a 0.93 precision result which means the model can label positive cases correctly.

However, other metrics have lower scores from CNN, BiLSTM, and LSTM at 0.92. This finding means that ANN is good at finding true threats, but it has a slightly lower consistency with other performance metrics. CNN and BiLSTM scored 0.95, LSTM 0.93, and ANN 0.92 in sensitivity, which indicated that CNN and BiLSTM were slightly better at finding true positive.

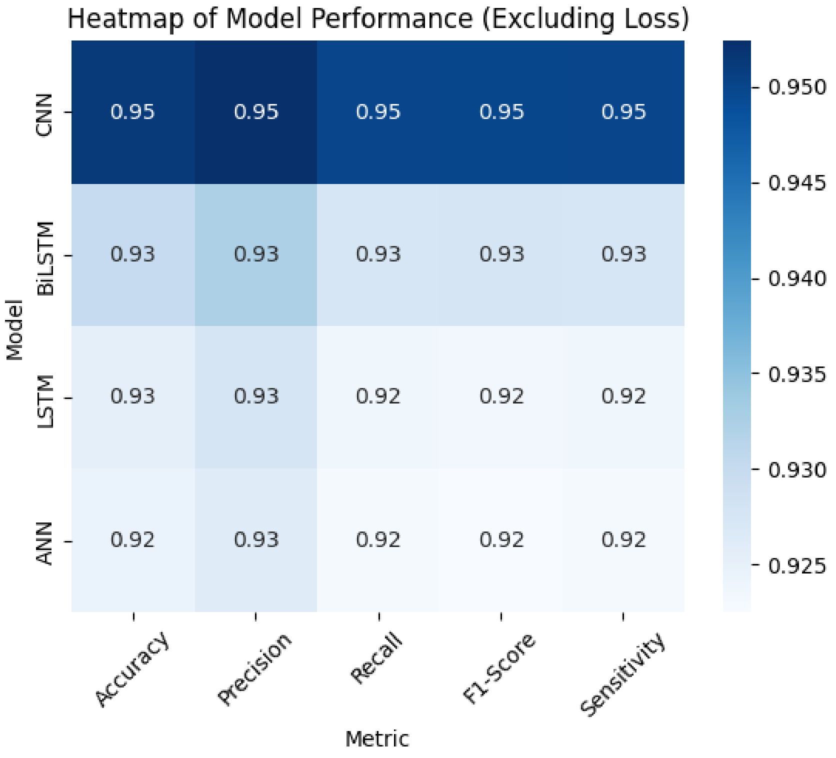


Figure 0.15. Heatmap of Model Performance.

The radar chart in Figure 4.16 illustrates the performance of different deep learning models on the WSNBFSF Dataset across several metrics: Accuracy, Precision, Recall (Sensitivity), F1-Score, and Loss. Each axis represents a metric, and the closer the plot is to the outer edge of the radar chart, the better the performance in that metric.

The Convolutional Neural Network (CNN) appears to have the outermost plot consistently across all metrics except Loss, indicating superior performance in Accuracy, Precision, Recall, F1-Score, and Sensitivity. This suggests that the CNN model is highly effective in correctly identifying true positives (high Precision), covering a majority of actual positive cases (high Recall and Sensitivity), and maintaining a balance between Precision and Recall (high F1-Score), all while making correct predictions (high Accuracy).

The Bidirectional Long Short-Term Memory (BiLSTM), Long Short-Term Memory (LSTM), and Artificial Neural Network (ANN) models show overlapping plots that are quite close to the CNN's, suggesting that their performances are comparable across these metrics. However, the ANN does display a slight inward dent on the Recall axis, indicating a lower true positive rate compared to the other models.

Notably, Loss is not plotted for these models, which would typically appear on an inverse scale with higher performance associated with a point closer to the center. However, given the absence of this metric on the chart, we cannot comment on how well the models minimized errors during training.

The radar chart conveys that while all models are strong performers across the key metrics of model evaluation, the CNN demonstrates a marginally better overall performance, which could make it the preferred choice for this particular dataset in network security applications.

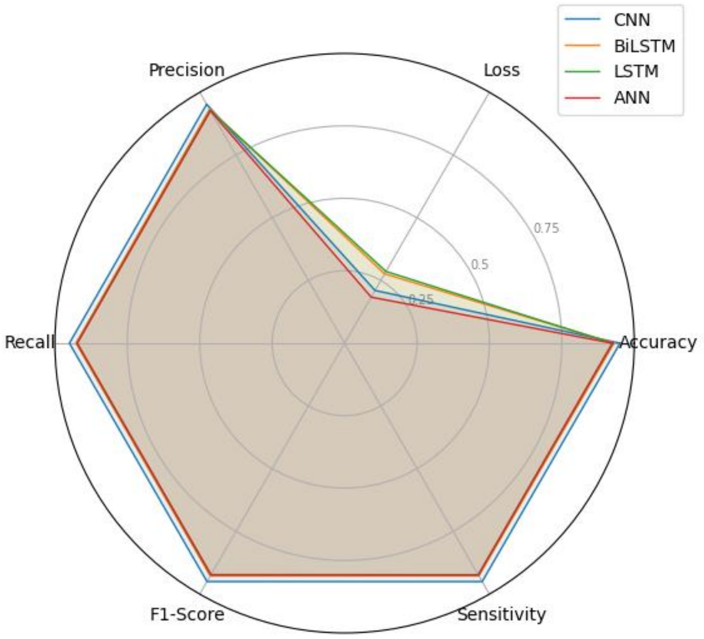


Figure 0.16. Radar chart about the performance of different deep learning.

The scatter plot in Figure 4.17 depicts the relationship between loss and accuracy for four different machine learning models tested on the WSNBFSF dataset. The Convolutional Neural Network (CNN) is marked by a dark blue dot and stands out with the highest accuracy, above 0.95, and the lowest loss, around 0.18. This suggests that the CNN is highly effective at making correct predictions and has a strong ability to generalize from the training data, thus likely to perform well on unseen data.

The BiLSTM and LSTM models are marked by teal and green dots respectively. Although the models carry out when exposed to the test set approximately the same, approximately 0.93, the BiLSTM model has slightly smaller loss compared to the LSTM. This is in the sense that on the x-axis, is located at a position closer to the left. In other terms, although the accuracy level the models is the same, the BiLSTM model is minimizes errors during the training process more due to its lesser loss.

The ANN is represented by the orange dot. At a value of loss higher than the CNN but lower than the LSTM models, and an accuracy level slightly closer to the LSTM models, about 0.92. From the position of the dot, though less accurate compared to the CNN or LSTM models, it has a relatively low loss. This implies that the ANN model best generalizes.

In summary, the CNN model is the best-performing model on the dataset in both high accuracy and low loss. The BiLSTM and LSTM models also exhibit high performance although they are less perfect compared to the CNN model. The ANN model has the least accuracy across all models, but it maintains just the right amount of loss compared to the other models.

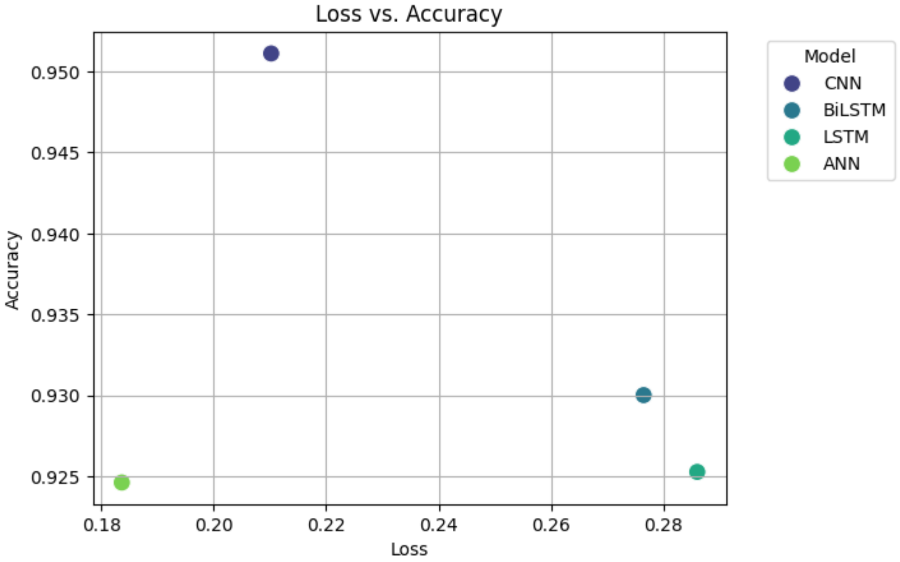


Figure 0.17. Scatter plot depicts the relationship between loss and accuracy.

4.4. RESULTS OF KDD DATASET

4.4.1. Results of ANN Model

The Artificial Neural Network model trained on the KDD dataset shows consistent progression over the 10 epochs in both the accuracy and loss graphs, as illustrated in Figure 4.18.

In the initial epoch, the training accuracy stands at 92.40% with the model's loss at 0.1993. The validation accuracy is marginally higher 95.03% with the validation loss of 0.1313. With each subsequent epoch, there was a pattern of increasing accuracy and diminishing loss, indicating improvement in the model's ability to learn from the training data. By the epoch's end, the training accuracy significantly improved to

97.49% with the loss standing at 0.0736. The validation accuracy is similarly high in both the validation loss and accuracy, attaining a high of 98.20% and 0.0613, respectively.

Similarly, the loss graph indicates the model's increased ability to generalize, as the gap between the training and the validation loss narrows over time. The accuracy graph shows the model's prediction capability, as the validation accuracy trends closely to the training accuracy throughout the training. It seems that the learning took place appropriately without developing a high bias during the training.

This implies a well-fitting model on the training data that maintains its generalization capability. Therefore, the model's results should be reliable when validating new, unseen observations from the dataset.

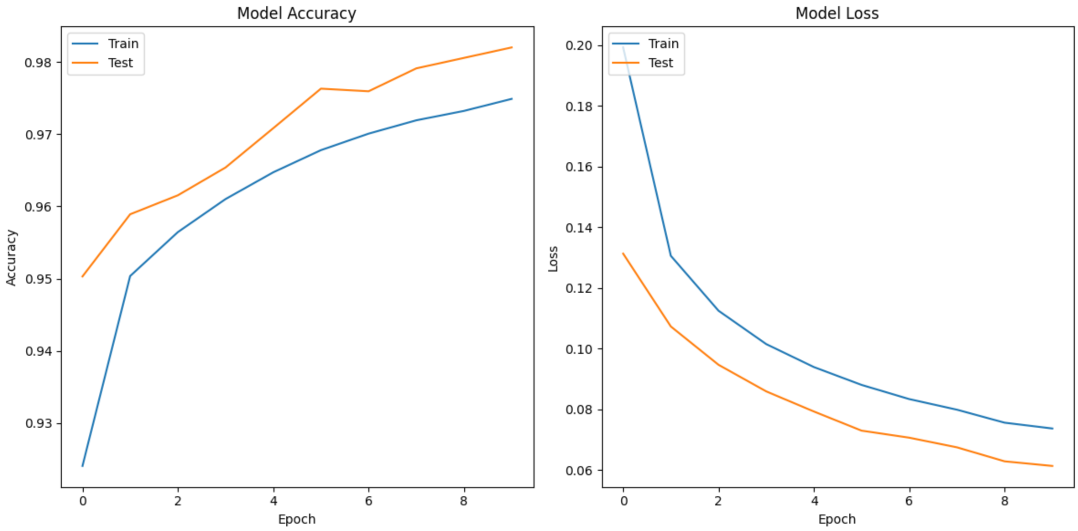


Figure 0.18. Training curves of ANN.

A confusion matrix depicting the Artificial Neural Network's performance in classifying the KDD dataset is presented in Figure 4.19. As could be seen, the matrix include a relatively large number of true positives (TP) and true negatives (TN) , which reflect the ANN correctly identifying, respectively, 26,989 cases of "anomaly" and 26,989 cases of "normal." At the same time, the matrix include 576 false negatives (FN) , which refer to "anomaly" that were indeed classified as "normal," and 576 false positives (FP) , which concern "normal" that were classified as "anomaly". Despite these misclassifications, the high values on the diagonal indicate that the ANN has performed very well in differentiating between 'anomaly' and 'normal' classes.

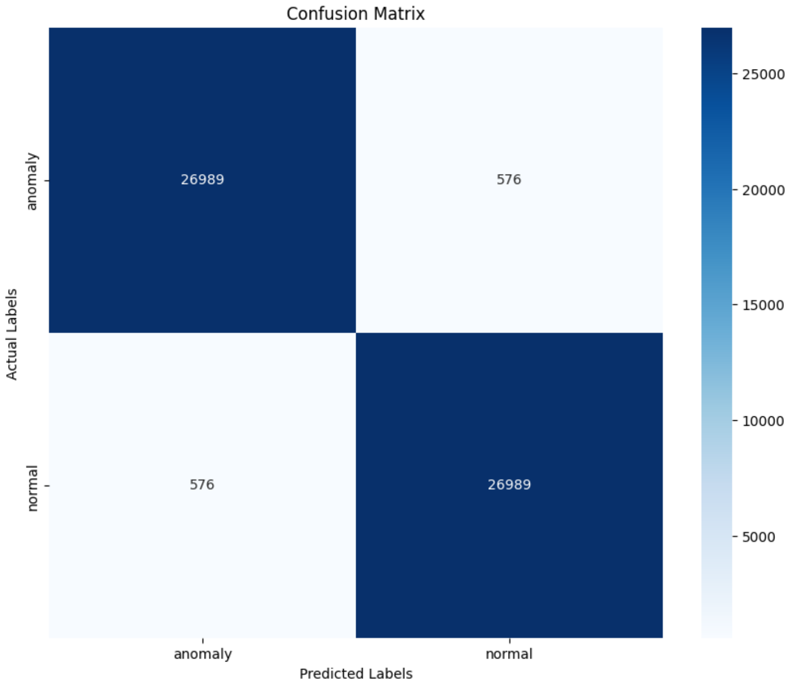
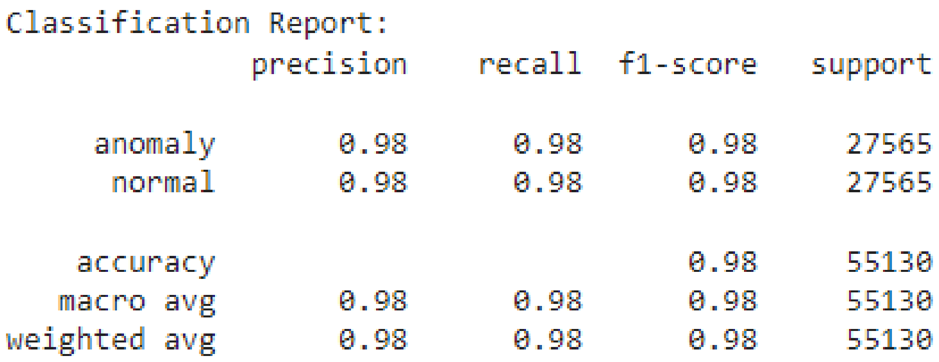


Figure 0.19. Confusion matrix of ANN.

The classification report in Figure 4.20 provides precise numerical values for the model's precision, recall, and F1-score, all of which are 0.98 for both classes, corroborating the high accuracy seen in the confusion matrix. 'Support' refers to the number of actual occurrences of each class in the dataset, which is evenly distributed with 27,565 instances each. The overall accuracy of the ANN is 0.98, which is exceptionally high. The macro, weighted, and accuracy averages are all consistent at 0.98, further demonstrating the ANN's effective classification performance on the KDD dataset. These results indicate that the ANN has a strong predictive ability and is highly reliable for this classification task.



```
Classification Report:
              precision    recall  f1-score   support

     anomaly       0.98      0.98      0.98     27565
      normal       0.98      0.98      0.98     27565

    accuracy                           0.98     55130
   macro avg       0.98      0.98      0.98     55130
weighted avg       0.98      0.98      0.98     55130
```

Figure 4.20. Classification report of ANN.

4.4.2. Results of LSTM Model

The pair of graphs displays the training progress of a Long Short-Term Memory (LSTM) model on the KDD dataset over 10 epochs, illustrating trends in model accuracy and loss(Figure 4.21).

In the accuracy graph, both training and test (validation) accuracies start at high levels — training accuracy at approximately 93.7% and test accuracy at 95.15% — and both improve steadily over time. By the 10th epoch, training accuracy has reached 97.54%, while test accuracy has attained 97.24%. The curves' convergence indicates that the model generalizes well without significant overfitting, as the test accuracy tracks closely with the training accuracy.

The loss graph shows a rapid decline in both training and test loss during the initial epochs, with training loss decreasing from around 0.23 to below 0.10, and test loss diminishing from approximately 0.17 to around 0.10. The test loss experiences a slight increase after the 5th epoch before stabilizing, which could suggest the beginnings of overfitting or an area where the model's learning rate and regularization parameters might need adjustment.

The epochs corroborate the trends seen in the graphs, with both training and validation loss decreasing and accuracy increasing as the epochs progress. The model demonstrates strong performance by the end of the training, with a high validation accuracy of 97.24% and a low validation loss of 0.1040 by the 10th epoch. The training process shows that the LSTM model has learned effectively from the KDD dataset and suggests that it could be a reliable classifier for the problem at hand.

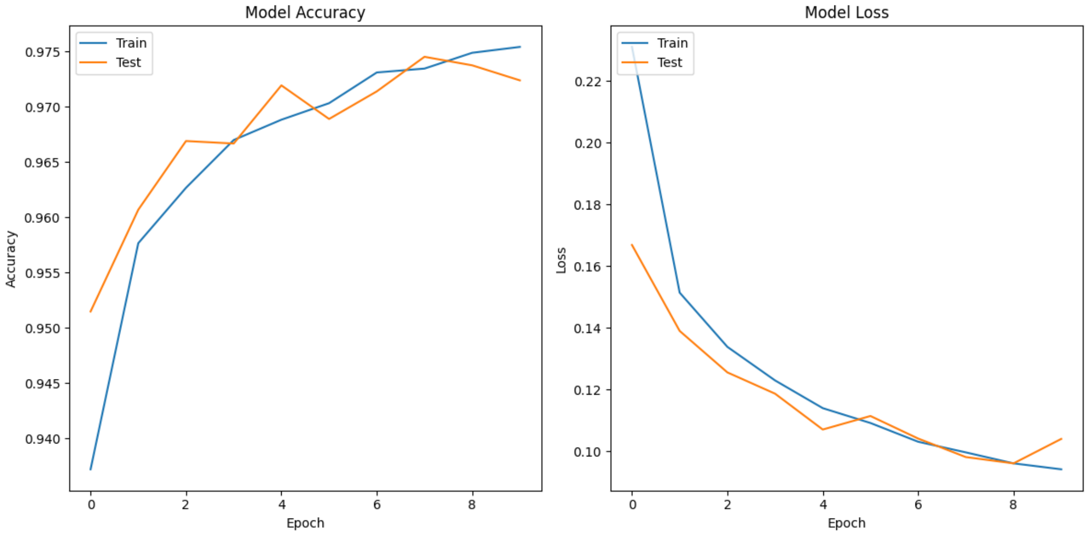


Figure 0.21. Training curves of LSTM.

For the confusion matrix in Figure 4.22, it shows the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by a Long Short-Term Memory (LSTM) model when classifying data from the KDD dataset into 'anomaly' and 'normal' classes. The model correctly identified 26,778 anomalies (TP) and 26,778 normals (TN)(Figure 4.22). However, there were 787 instances where normal behavior was incorrectly classified as an anomaly (FP) and 787 instances where anomalous behavior was classified as normal (FN). Despite these errors, the large numbers on the matrix's diagonal indicate that the model has high accuracy in its predictions.

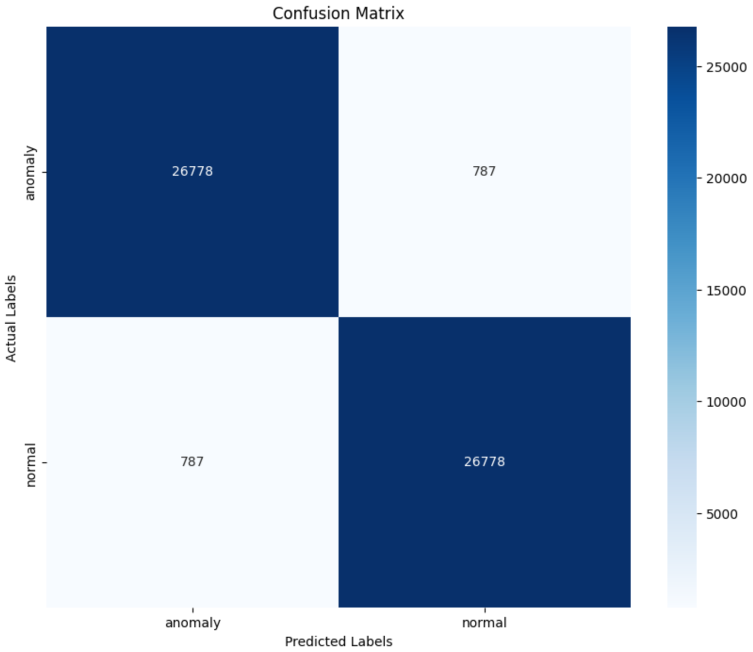
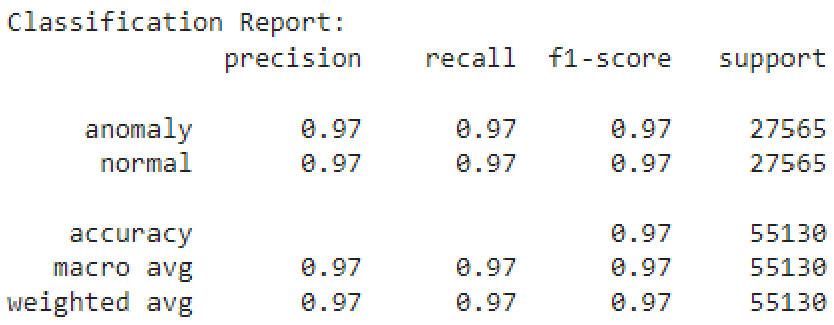


Figure 4.22. CM of LSTM.

In the classification report, the LSTM model achieves a precision, recall, and F1-score of 0.97 for both classes, which is quite high (Figure 4.23). Precision is a measure of the model's accuracy regarding false positives, recall (or sensitivity) measures the accuracy with regard to false negatives, and the F1-score provides a balance between precision and recall. The 'support' represents the actual number of occurrences for each class in the dataset, and these are evenly distributed. The overall accuracy of the model is also reported as 0.97. The consistent high performance across all these metrics suggests that the LSTM model is a strong performer for this classification task, able to distinguish well between normal and anomalous behaviors in the dataset.



```
Classification Report:
              precision    recall  f1-score   support

     anomaly       0.97      0.97      0.97     27565
      normal       0.97      0.97      0.97     27565

    accuracy                           0.97     55130
   macro avg       0.97      0.97      0.97     55130
weighted avg       0.97      0.97      0.97     55130
```

Figure 0.23. Classification report of LSTM.

4.4.3. Results of BiLSTM Model

Figure 4.24 illustrates the performance of a Bidirectional Long Short-Term Memory (BiLSTM) model during training and validation on the KDD dataset.

From the accuracy chart, we observe that the model's training and validation accuracies start above 94.5% and show a generally upward trend throughout the epochs. There's a notable increase in validation accuracy between epochs 1 and 2, and while there are some fluctuations, both accuracies plateau near the end, with the training accuracy slightly higher than the validation accuracy. This could indicate the model is beginning to overfit the training data, as the gap between the training and validation accuracy is widening.

The loss chart shows that both training and validation loss decrease sharply at the start, which is typical as the model begins to learn from the data. By the end of the 10 epochs, both losses have leveled off, with the validation loss showing slight volatility but remaining below the training loss throughout. This can sometimes be a sign of the model performing better on the validation set than on the training set, which may be due to the regularization effects or the model benefiting from the bidirectional nature of the LSTM layers.

The training log confirms these observations, with initial high improvements in accuracy and reductions in loss that begin to stabilize towards the later epochs. By the final epoch, the model achieves a high validation accuracy of approximately 97.75%, and the validation loss settles at around 0.0882, which is consistent with the trends observed in the loss chart.

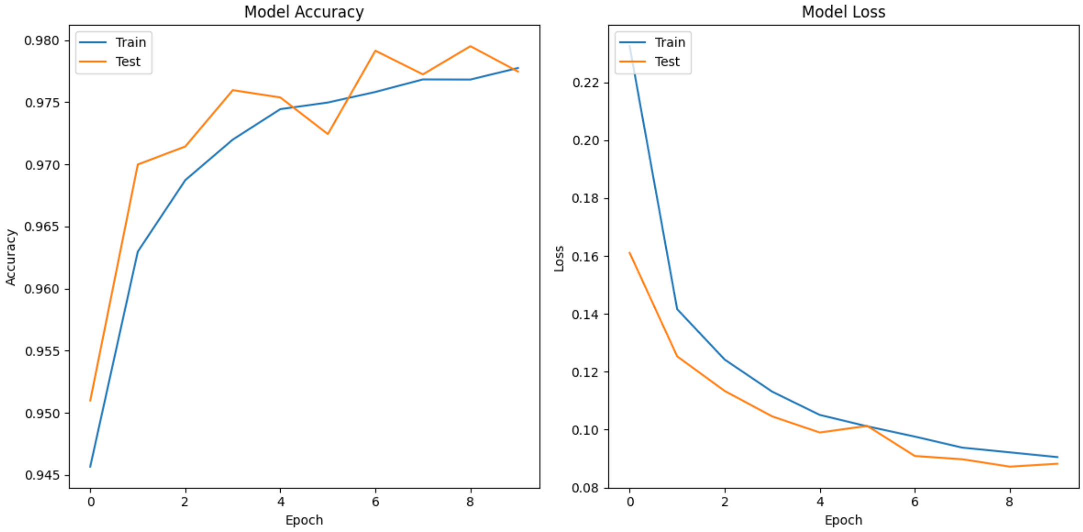


Figure 0.24. Training curves of BiLSTM.

The confusion matrix for the BiLSTM model indicates a high level of accuracy in distinguishing between 'anomaly' and 'normal' classes in the KDD dataset (Figure 4.25). The model has correctly predicted a vast majority of the samples with 26,895 true positives (anomalies correctly identified as anomalies) and 26,895 true negatives (normals correctly identified as normal). However, there are 670 false positives (normals incorrectly labeled as anomalies) and 670 false negatives (anomalies incorrectly labeled as normal), which are relatively small compared to the number of true classifications.

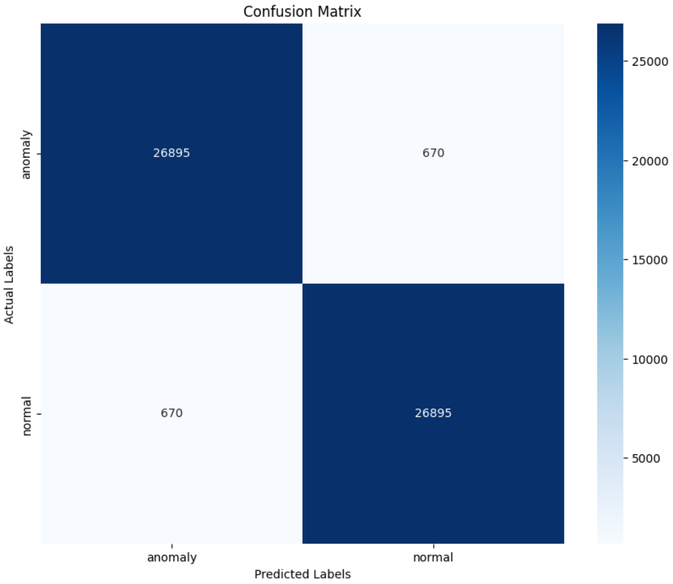
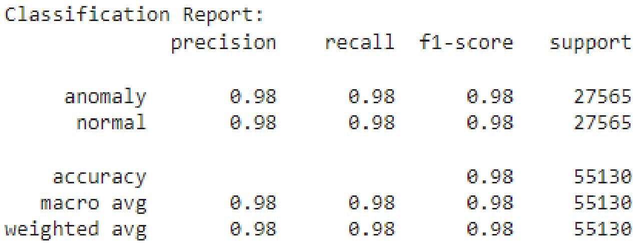


Figure 0.25. CM of BiLSTM.

The classification report supports the high accuracy observed in the confusion matrix, with a precision, recall, and F1-score of 0.98 for both 'anomaly' and 'normal' classes(Figure 4.26). The overall accuracy of the model is also reported to be 0.98, which is very high. These scores suggest that the BiLSTM model is not only good at classifying anomalies but does so with a balanced performance in terms of both precision and recall.

In the context of network security, where it is critical to correctly identify as many anomalies as possible while minimizing the false alarms, the BiLSTM model's performance is exemplary, as indicated by the near-equal true positive and true negative rates. This balance is crucial for practical applications, as it minimizes the chances of missing genuine threats and reduces the workload associated with investigating false alerts.



```
Classification Report:
              precision    recall  f1-score   support

     anomaly       0.98      0.98      0.98     27565
      normal       0.98      0.98      0.98     27565

    accuracy                           0.98     55130
   macro avg       0.98      0.98      0.98     55130
weighted avg       0.98      0.98      0.98     55130
```

Figure 0.26. Classification report of BiLSTM

4.4.4. Results of CNN Model

The training performance charts in Figure 4.27indicate the progression of a Convolutional Neural Network (CNN) model's training on the KDD dataset.

From the accuracy chart, the CNN model's training accuracy begins at 95.52% and consistently increases over time, reaching an impressive 98.00% by the 10th epoch. The test accuracy closely follows the training accuracy, starting at 97.02% and rising to 98.18%, suggesting that the model is generalizing well without overfitting significantly to the training data.

The loss chart complements the accuracy chart, showing a sharp decline in both training and test loss initially, with training loss dropping from 0.1881 to 0.0802, and test loss decreasing from 0.1223 to 0.0762 by the end of the training. The slight increases in test loss around epochs 6 and 7 could indicate some overfitting or a need for further optimization, but the subsequent reduction in loss indicates that the model managed to recover and improve.

The reported epoch log provides additional detail, with both training and validation loss steadily decreasing and accuracy improving, which is a strong sign of effective learning. The CNN has not only learned well but has done so in a stable manner, as seen by the minor fluctuations in the test loss and accuracy.

By the end of the training process, the CNN model demonstrates high predictive accuracy and low loss on both the training and test sets, indicating it is an excellent candidate for effectively classifying network traffic on the KDD dataset. The results suggest that the CNN model would perform well when applied to similar tasks in practice.

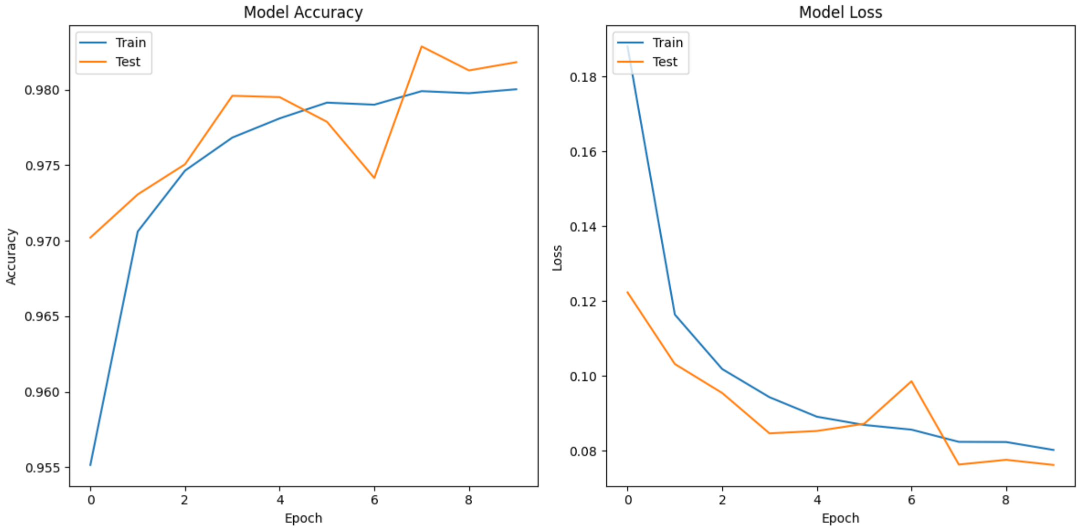


Figure 0.27. Training curves of CNN.

The confusion matrix depicts the performance of the CNN on the KDD dataset (Figure 4.29). The model has accurately predicted 27,025 anomalies and 27,025 normal instances, showing a high number of true positives and true negatives. However, there are 540 instances where normal behaviors were mistakenly classified as anomalies (false positives) and 540 where anomalies were misclassified as normal (false negatives). Despite these errors, the overall high numbers of correct predictions on both classes indicate that CNN has performed with high accuracy.

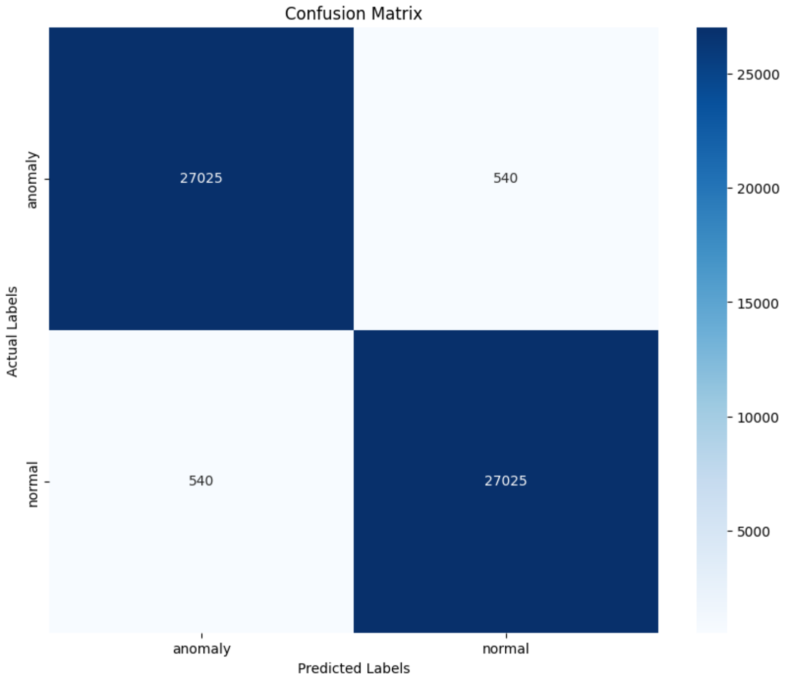
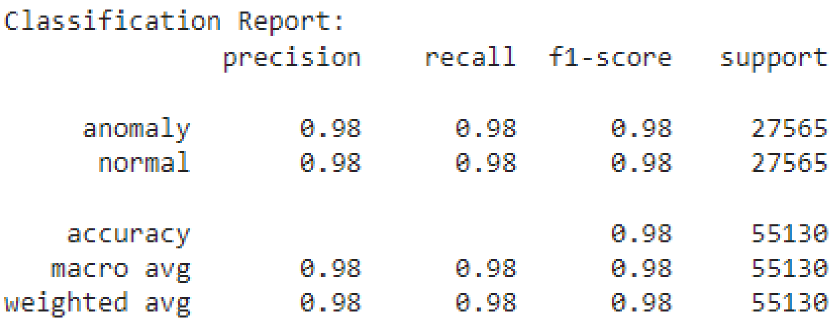


Figure 4.28. CM of CNN.

The classification report complements the confusion matrix with precise metrics: a precision, recall, and F1-score of 0.98 for both classes, which indicates excellent model performance (Figure 4.29). Precision measures the accuracy of positive predictions, recall indicates the ability to find all positive instances, and the F1-score is a harmonic mean of precision and recall, showing a balance between them. The accuracy of the model stands at 0.98, which is consistent with the high values observed in the confusion matrix, underscoring the model's reliability in classifying network traffic accurately. The support column confirms an equal distribution of both classes in the dataset, with each class having 27,565 instances.



```
Classification Report:
              precision    recall  f1-score   support

     anomaly       0.98      0.98      0.98     27565
      normal       0.98      0.98      0.98     27565

    accuracy                           0.98     55130
   macro avg       0.98      0.98      0.98     55130
weighted avg       0.98      0.98      0.98     55130
```

Figure 0.29. Classification report ofCNN.

4.4.5. Comparison

The provided bar charts in Figure 4.30 compare the performance of four different models—CNN, BiLSTM, LSTM, and ANN—on the KDD (presumably mistyped as KNN) Dataset in terms of accuracy and loss.

From the accuracy comparison chart, it is observed that the CNN and ANN models achieve the highest accuracy at 0.98, closely followed by the BiLSTM model at the same mark, and then the LSTM model slightly lower at 0.97. This suggests that all models are performing quite well on the dataset, with CNN and ANN models showing a slight edge over the others.

The loss comparison chart reveals that the ANN model has the lowest loss at 0.06, indicating that it has the best generalization performance among the four models. The CNN model follows with a loss of 0.08, then the BiLSTM with 0.09, and the LSTM has the highest loss at 0.1. While the loss values for CNN, BiLSTM, and LSTM are close, indicating similar levels of model fit, the ANN's lower loss suggests it may be the most efficient at reducing the error between the predicted and actual values.

In summary, the comparisons show that while each model has much to offer, the ANN strikes the best balance between high accuracy and low loss on this particular dataset which may position it as the best option for anomaly detection in network security use-cases. The CNN can also appear as a strong setup for such a task due to a strong balance between High accuracy and low loss.

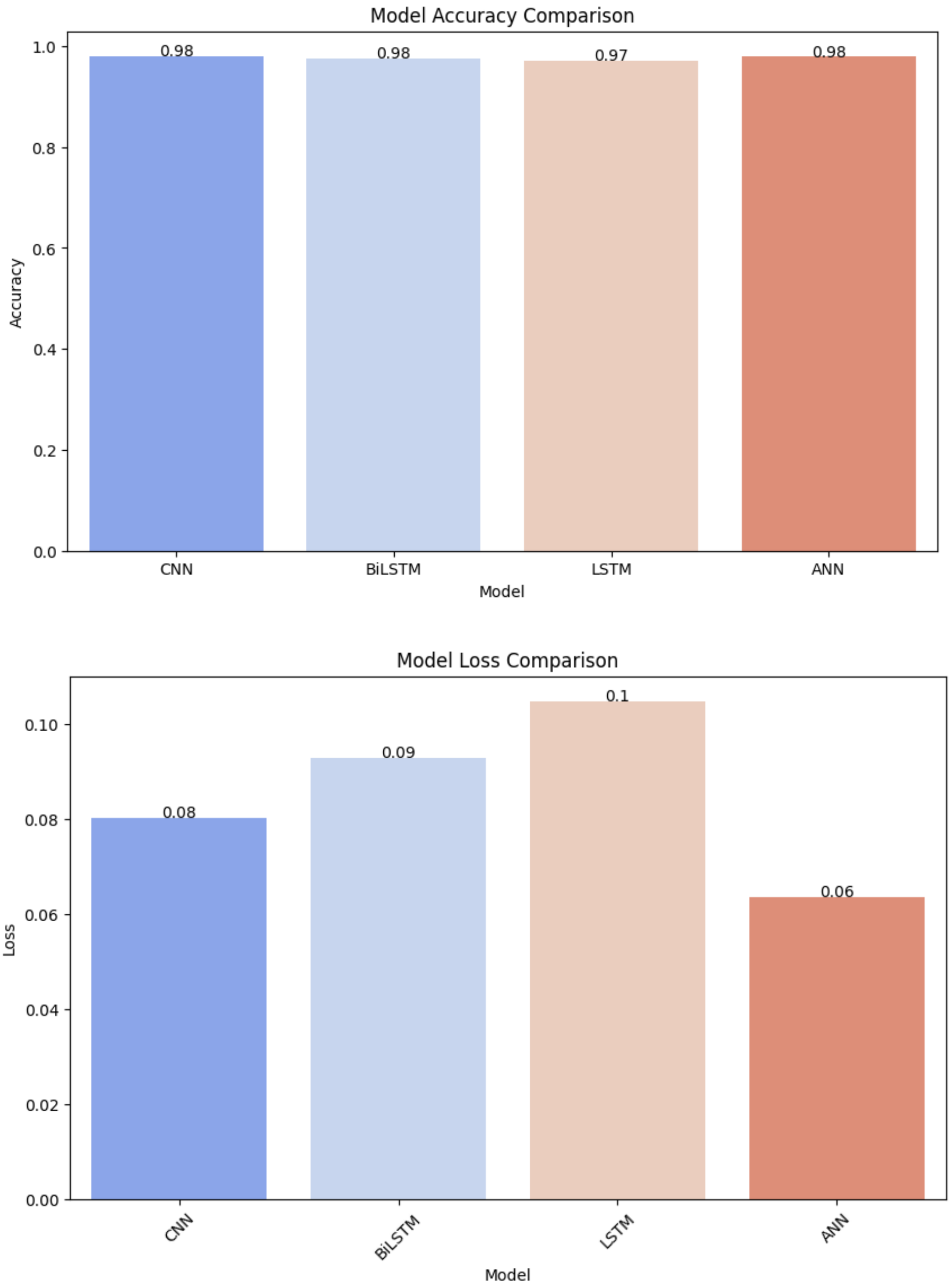


Figure 4.30. Model accuracy and loss comparison on the KDD dataset.

Figure 4.31 presents a bar graph of the comparative analysis of four ML models, namely, CNN, BiLSTM, LSTM, and ANN, in terms of precision, recall, and F1-score on the KDD dataset. High precision, recall, and F1-scores are presented by all models, which implies that each one is effective and can classify the data with minimum false positives and negatives. However, the CNN and ANN have scored slightly more compared to the BiLSTM and LSTM models. The obtained results are equal to 0.98, which means that these two models are slightly better and precise in classifying these data.

The LSTM model shows a slight dip in performance compared to the others, with all metrics at 0.97. Despite this, it's still a high score, indicating the LSTM model also performs well.

The consistent high scores across all three metrics for each model suggest that each model has successfully captured the underlying patterns in the dataset, leading to strong predictive performance. For tasks within the KDD dataset's scope, such as network intrusion detection, any of these models could be a viable option, with CNN and ANN offering marginally better performance.

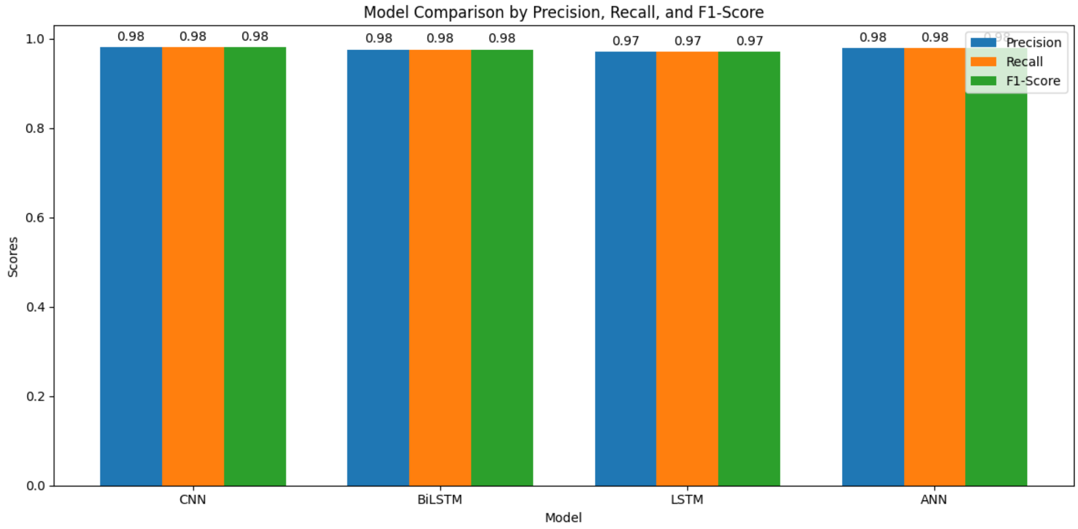


Figure 4.31. Performance of various deep learning models on the KDD Dataset through Precision, Recall, and F1-Score metrics.

The heatmap provides a visual comparison of different deep learning models' performance metrics on the KDD dataset. All models exhibit high metrics across the board—CNN, BiLSTM, and ANN have uniform scores of 0.98 in all categories, while LSTM lags slightly with scores of 0.97. The uniformity and high scores across Accuracy, Precision, Recall, F1-Score, and Sensitivity for CNN, BiLSTM, and ANN models suggest they are all highly capable of delivering reliable and consistent predictions. LSTM's performance, though slightly lower, is still within a high range, indicating it is also a robust model for this dataset.

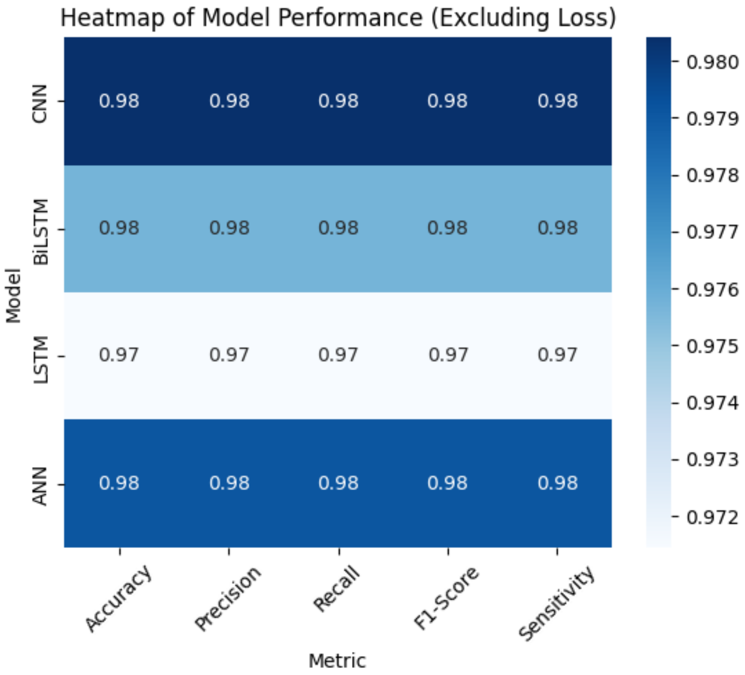


Figure 0.32. Heatmap of Model Performanceon KDD Dataset.

The scatter plot in Figure 4.33 contrasts the loss and accuracy of the models. Ideally, we want a model positioned towards the bottom right, indicating low loss and high accuracy. The CNN, represented by the blue dot, shows the best balance with the lowest loss and highest accuracy, suggesting an optimal performance. The other models are clustered closely with slightly higher loss values, indicating a slightly less efficient but still strong performance.

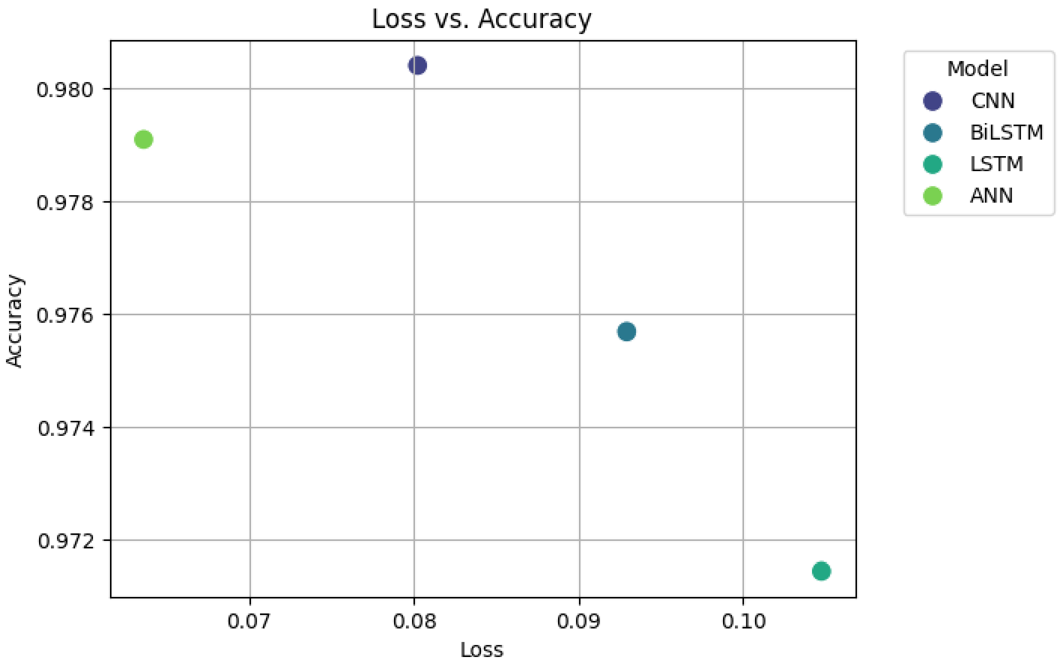


Figure 04.33. Scatter plot depicts the relationship between loss and accuracy on KDD dataset.

The radar chart in Figure 4.34 illustrates the model performance metrics for CNN, BiLSTM, LSTM, and ANN. All models' plots almost entirely overlap and extend close to the edges of the radar chart, confirming their high performance across all measured metrics. This visual reinforces the previous assessments, showing how each model's capabilities in terms of Accuracy, Precision, Recall, F1-Score, and Sensitivity are near-optimal and quite similar across the models, with very slight differences between them. The chart emphasizes that all models perform well, with CNN and ANN being slightly more preferable for tasks involving the KDD dataset.

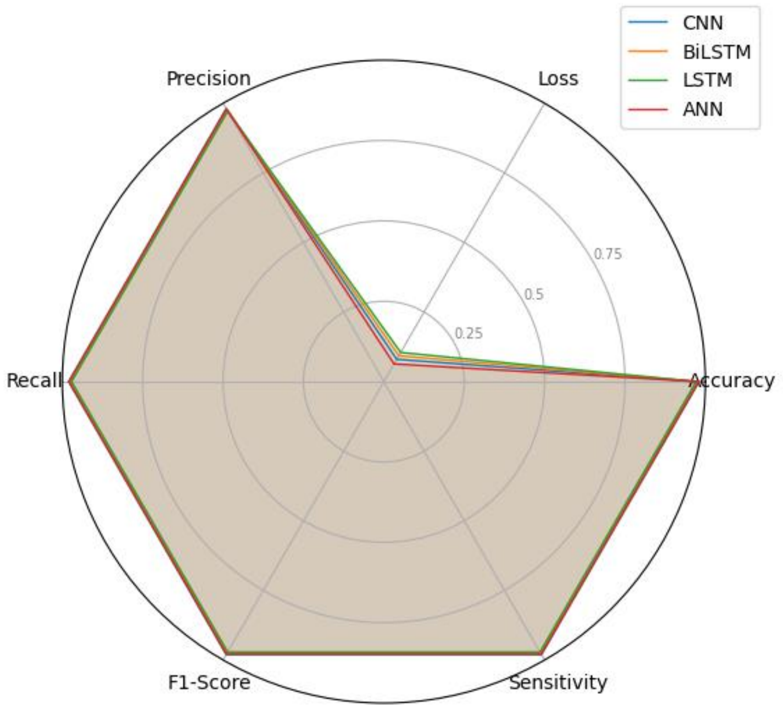


Figure 0.34. Radar chart about the performance of different deep learning on KDD dataset.

4.5. COMPARISON BETWEEN WSNBFSF AND KDD DATASET

The bar chart in Figure 4.35 visualizes the accuracy comparison of four machine learning models (CNN, BiLSTM, LSTM, and ANN) on two different datasets: WSNBFSF and KDD. The performance on the KDD dataset is consistently higher for all models compared to the WSNBFSF dataset, which may suggest that the patterns in the KDD dataset are more distinct or that it may contain fewer complexities for the models to learn and predict accurately.

The CNN and ANN models display the highest accuracy on the KDD dataset, both achieving slightly over 98%. This performance variability indicates that the features of the KDD dataset may be better suited for this type of architecture, which helps them capture the underlying pattern more efficiently. Both the LSTM and BiLSTM models, despite not lagging too much behind the KDD performance, still show a minor drop in performance from the CNN and ANN models, which could be attributed to their sequential data nature that may not be as well-suited for the KDD dataset . On the WSNBFSF dataset, the performance of all models is slightly reduced in comparison to the models' KDD performance. The CNN model is still in the lead but shown to degrade more in terms of the KDD performance. This could indicate that the WSNBFSF is a more challenging dataset, with more noise or more complexity. This performance divergence showcases the significance of dataset-based model performance studies. The similar performance of the CNN and ANN models across both datasets suggests a robustness and flexibility that makes them applicable to a great number of tasks, while LSTM and BiLSTM models show less consistence between the two datasets, possibly requiring more tuning or some data pre-processing to show the best results.

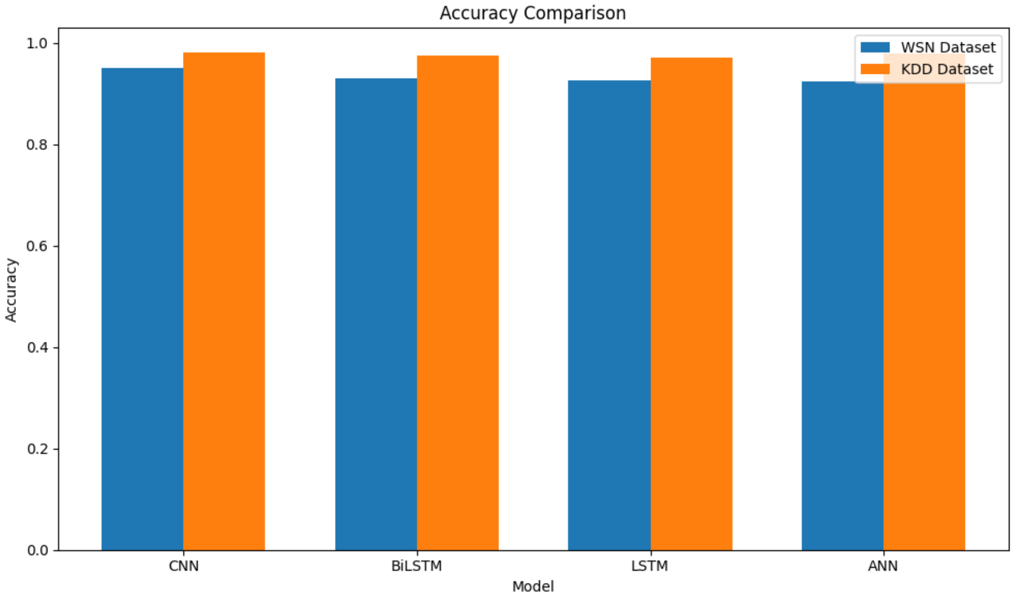


Figure 0.35. Accuracy comparison between WSNBFSF and KDD Dataset.

The Figure 4.36 provided bar chart compares the precision of four machine learning models, CNN, BiLSTM, LSTM, and ANN, based on the WSNBFSF and KDD datasets. Precision measures the extent to which a model is returning relevant results, true positives, or the extent to which it avoids false positives. It indicates the importance of correctly identifying positive cases. The CNN model has the highest precision and records perfect on the KDD dataset 0.980410 and has a slightly lower measure of 0.952387 on the WSNBFSF dataset. The trend implies that CNN has strong feature extraction capability due to the precision determination based on different datasets; thus is reliable in situations that need the positive case to be identified accurately.

The BiLSTM and LSTM models feature excellent precision records on the KDD dataset compared to the WSNBFSD dataset, which have significantly fewer values. The finding implies that the sequential aspect of the BiLSTM and LSTM model fits into the KDD dataset patterns. The ANN model has lots of precision as CNN and records 0.979104 on the KDD dataset and 0.926237 on the WSNBFSF dataset.

The small discrepancy in the precision between datasets for the ANN model suggests that it has a stable performance but may require adjustments or more nuanced feature engineering when dealing with different types of data.

All models perform exceptionally well in terms of precision on the KDD dataset, with CNN and ANN slightly outperforming BiLSTM and LSTM. On the WSNBFSF dataset, all models see a decrease in precision, but they still maintain high performance, indicating their potential effectiveness in various real-world applications where precision is critical. However, the choice between these models for a specific task would likely consider other factors beyond precision, such as recall, F1-score, interpretability, and computational efficiency.

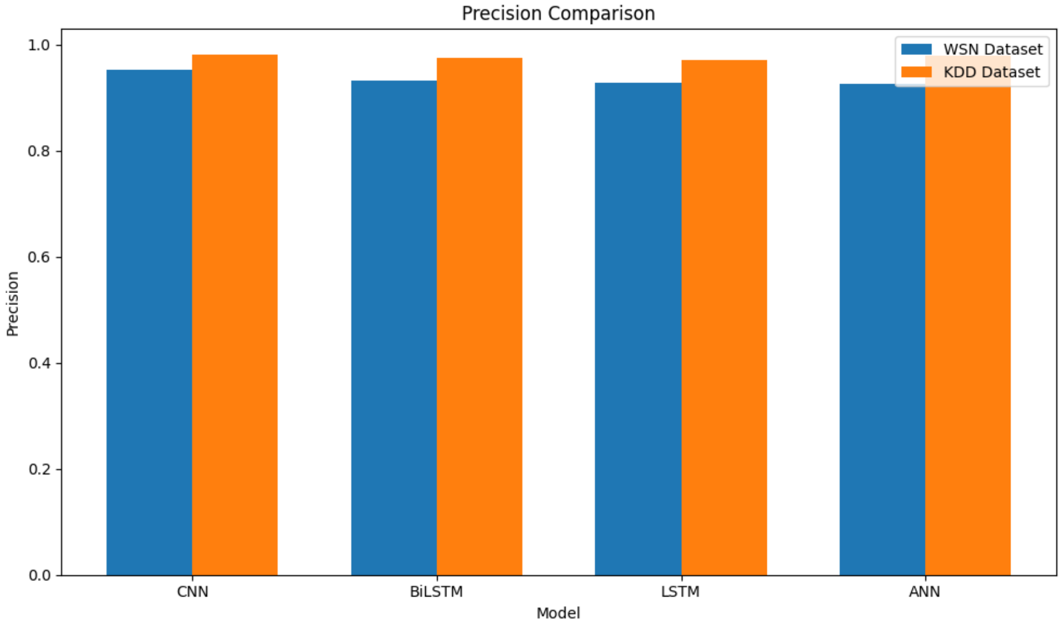
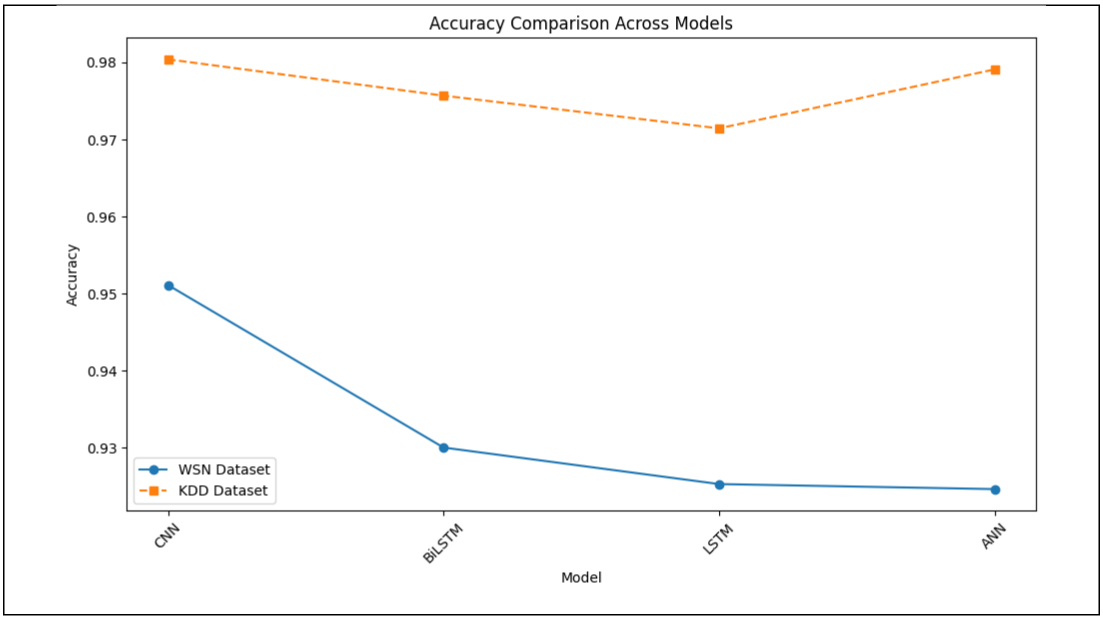


Figure 4.36. Precision comparison between WSNBFSF and KDD Dataset.

The two figures in 4.37 provided appear to give a comparative analysis of the performance of different machine learning models on the WSN and KDD datasets, in terms of accuracy and the trade-off between loss and accuracy.

The first figure presents a line chart comparing the accuracy of four models—CNN, BiLSTM, LSTM, and ANN—on the WSN and KDD datasets. The dashed line for the KDD dataset suggests that the models achieve very high accuracy, all above 97%, with the accuracy on the KDD dataset being consistently higher across all models compared to the WSN dataset. This might indicate that the KDD dataset is either less complex, better cleaned, or that the features in this dataset are more discernible for the models.

The second figure, a scatter plot, plots loss versus accuracy for the models on both datasets. There's a clear distinction between the two datasets: for the WSN dataset, as the loss increases, accuracy decreases, which is a common and expected trend. However, the KDD dataset shows very high accuracy at lower loss levels, implying that the models are more effectively fitting the KDD data with less error.



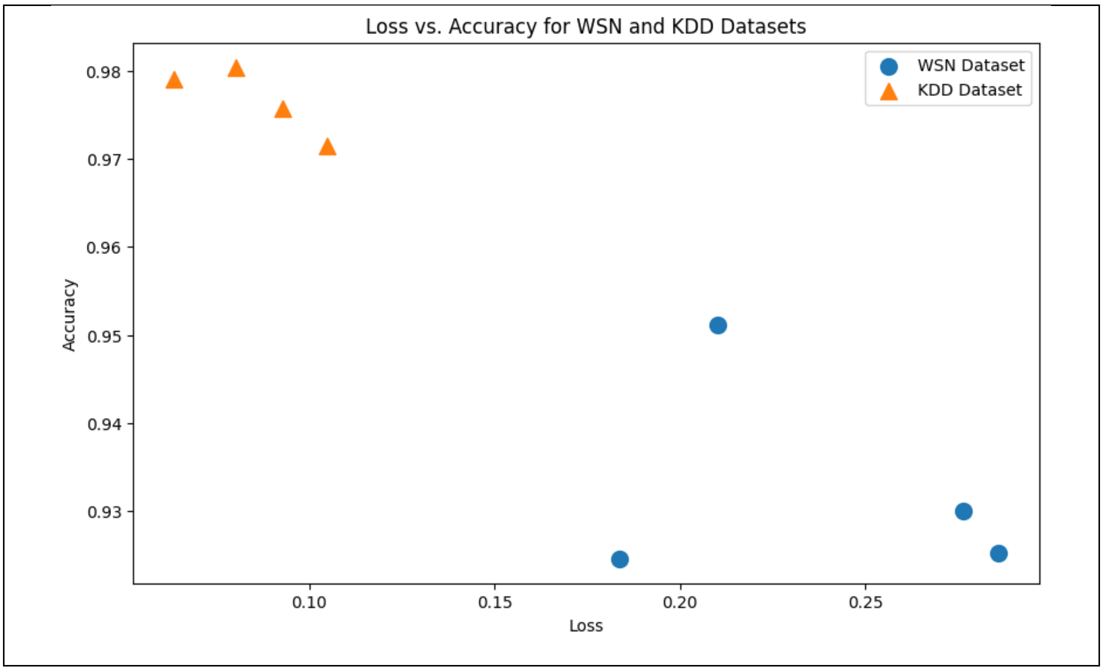


Figure 0.37. Comparative analysis of the performance of different DL models on the WSN and KDD datasets.

This bar chart in Figure 4.38 presents a direct comparison between the best-performing models for the WSN and KDD datasets, with a focus on accuracy. The bar for the WSN dataset (in blue) stands at 0.95, indicating that the best model (CNN) achieves 95% accuracy. For the KDD dataset (in green), the best model's (CNN)accuracy is even higher, at 0.98 or 98%.

The fact that both bars represent the best models for their respective datasets suggests that while both models perform well, the model trained on the KDD dataset is slightly more accurate. This could mean that the patterns in the KDD dataset are more frequently identified by the model or that the dataset itself is less complex or has less noise, as a result of which the model can achieve m0ore accurate predictions. Moreover, both datasets' models can reach such high accuracies also mean that they are well-tuned and can effectively capture the underlying patterns.

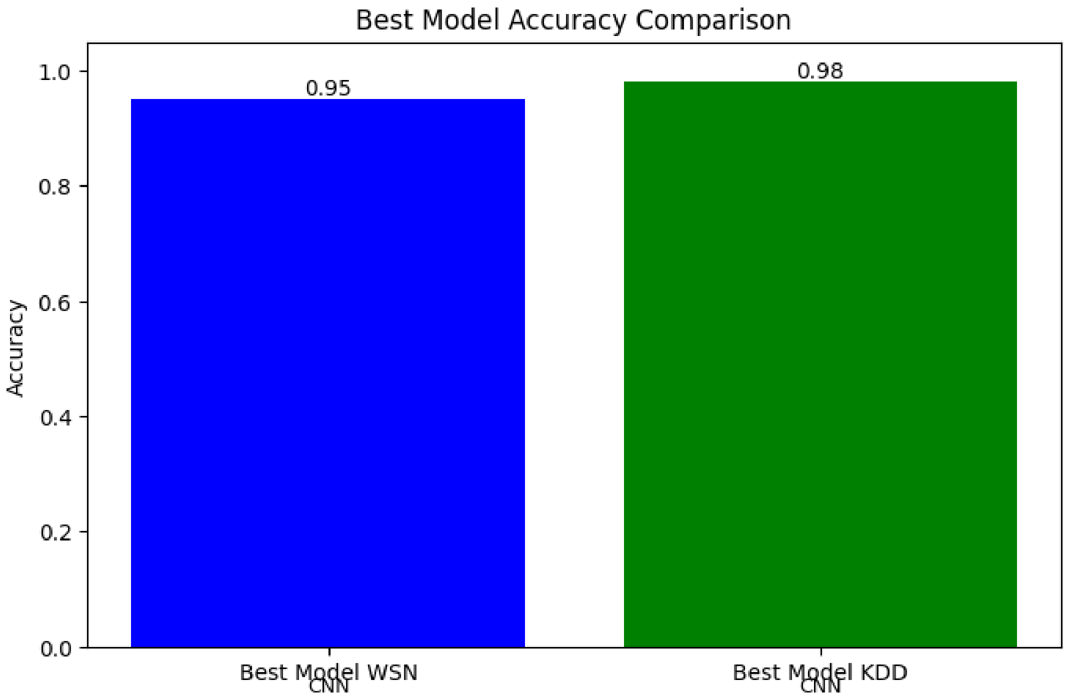


Figure 4.38. Best Model Accuracy Comparison.

PART 5

SUMMARY

This chapter presents a detailed analysis of how the selected deep learning models perform based on two unique datasets, namely WSNBFSF and KDD datasets. Having already established and described the objectives and the experimental setup, the following subsections provide a critical evaluation of the ANN, LSTM, BiLSTM, and CNN models' performance on both datasets. Furthermore, it involves a comparison of models using similar performance indicators, including accuracy, precision, and recall, among others, on each dataset.

The penultimate section synthesizes the information to compare the four models on two datasets and validate the results by interpreting the findings. It also provides an insight into the models' performance strength and weakness to affirm their applicability in real-world applications. The chapter finally ends with a conclusion that summarizes the outcomes of the experiment in form of findings on a comparative basis. Overall, it discusses the performance of the models regarding the desired security threats concerning network environments. The aim is to help one choose the most appropriate deep learning model for anomaly detection and attack recognition in network security applications.

5.1. CONCLUSION AND FUTURE WORKS

5.1.1. Conclusion

To summarize, our thesis charted the course of a vital endeavor improving the security of networks by employing innovative utilizations of deep learning for anomaly and attack detection in network traffic. Thus, we began from the observation of the increasing complexity of cyber threats and the pressing need for their advanced detection. Then, we undertook the diligent collection and preprocessing of two important datasets, WSNBFSF and KDD, making sure that the data that would be used to train our models is sound and representative. With the help of four deep learning architectures Artificial Neural Networks, Long Short-Term Memory networks, Bidirectional Long Short-Term Memory networks, and Convolutional Neural Networks we employed their unique strengths and capabilities in learning on the data intricate patterns and dependencies.

Our experiments, conducted in a convenient and computationally powerful Python and Google Colab setup, provided us with valuable results. The CNN model, able to discern spatial hierarchies and features in the network traffic, proved to be the best model by achieving the highest accuracy in detecting network anomalies on the KDD dataset. The LSTM and BiLSTM, focused on temporal data, prove to be very precise models but slightly less accurate than CNN architectures. The ANN model, despite showing robust performance, appears to be not tailor-made for the intricacies of the temporal data of network traffic.

Overall, the thesis concludes that deep learning is a strong aid in the arsenal of network security, promising high accuracy and precision in the detection of anomalies. The thesis statement opens further avenues for exploring hybrid models or other neural architectures that could offer even more protection in the constantly shifting arena of cyber threats. Moreover, the results point to the usefulness of applying these methods in real use-cases, ensuring more reliable and secure network models.

5.2. FUTURE WORKS

The promising outcomes of our thesis lay the groundwork for numerous pathways in future research. One potential direction is the exploration of hybrid deep learning models that combine the strengths of CNNs with RNNs, such as LSTM or BiLSTM, to enhance the detection of complex anomalies in network traffic. Further studies could also delve into the application of transfer learning, where knowledge from pre-trained models on large datasets could be transferred to improve performance on smaller, domain-specific datasets.

Advancements in unsupervised and semi-supervised learning offer another fertile ground for research, particularly for scenarios where labeled data is scarce or expensive to obtain. These methods could help in the detection of zero-day attacks by identifying subtle, previously unseen patterns. Additionally, the integration of deep reinforcement learning could lead to systems that not only detect threats but also learn and adapt their detection strategies over time.

Another avenue is the refinement of feature selection techniques to reduce dimensionality and computational costs while preserving, or even enhancing, model performance. With the ever-increasing volume of network data, efficient real-time analysis will be paramount.

Moreover, addressing the challenges posed by adversarial attacks on deep learning models themselves is an area that warrants attention. Developing robust models that can withstand adversarial manipulation is critical to ensure the reliability of anomaly detection systems.

Lastly, extending our work to different types of network architectures, such as Internet of Things (IoT) networks and cloud infrastructures, will be essential as these environments become increasingly ubiquitous and integral to modern computing landscapes. Each of these environments presents unique challenges and opportunities for deep learning applications in security.

REFERANCE

[1] Cremer, F., Sheehan, B., Fortmann, M., Kia, A. N., Mullins, M., Murphy, F., & Materne, S. (2022). Cyber risk and cybersecurity: a systematic review of data availability. Geneva Papers on Risk and Insurance. Issues and Practice, 47(3), 698. doi: 10.1057/s41288-022-00266-6

[2] Bada, M., & Nurse, J. R. C. (2021). Profiling the Cybercriminal: A Systematic Review of Research. arXiv, 2105.02930. Retrieved from https://arxiv.org/abs/2105.02930v2

[3] Ahmad, R., Alsmadi, I., Alhamdani, W., &Tawalbeh, L. (2023). Zero-day attack detection: a systematic literature review. Artif. Intell. Rev., 56(10), 10733–10811. doi: 10.1007/s10462-023-10437-z

[4] Sarker, I. H. (2021). Deep Cybersecurity: A Comprehensive Overview from Neural Network and Deep Learning Perspective. SN Comput. Sci., 2(3), 154–16. doi: 10.1007/s42979-021-00535-6

[5] Imamverdiyev, Y. N., & Abdullayeva, F. J. (2020). Deep Learning in Cybersecurity: Challenges and Approaches. International Journal of Cyber Warfare and Terrorism (IJCWT), 10(2), 82–105. doi: 10.4018/IJCWT.2020040105

[6] Chen, Z. . Deep Learning for Cybersecurity: A Review. 2020 International Conference on Computing and Data Science (CDS). IEEE. doi: 10.1109/CDS49703.2020.00009

[7] Sarker, I. H., Furhad, M. H., &Nowrozy, R. (2021). AI-Driven Cybersecurity: An Overview, Security Intelligence Modeling and Research Directions. SN Comput. Sci., 2(3), 173–18. doi: 10.1007/s42979-021-00557-0

[8] Farooq, O., & Martin, I. (2023). Cybersecurity Challenges in the Era of Digital Transformation. *Journal of Emerging Technology and Digital Transformation*, *2*(2), 102-113.

[9] Namanya, A. P., Cullen, A., Awan, I. U., &Disso, J. P. . The World of Malware: An Overview. 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud). IEEE. doi: 10.1109/FiCloud.2018.00067

[10] Zlomislić, V., Fertalj, K., &Sruk, V. (2017). Denial of service attacks, defences and research challenges. Cluster Comput., 20(1), 661–671. doi: 10.1007/s10586-017-0730-x

[11] Handa, A., Sharma, A., & Shukla, S. K. (2019). Machine learning in cybersecurity: A review. WIREs Data Min. Knowl. Discovery, 9(4), e1306. doi: 10.1002/widm.1306

[12] Cremer, F., Sheehan, B., Fortmann, M., Kia, A. N., Mullins, M., Murphy, F., & Materne, S. (2022). Cyber risk and cybersecurity: a systematic review of data availability. Geneva Pap. Risk Insur. Issues Pract., 47(3), 698–736. doi: 10.1057/s41288-022-00266-6

[13] Costa, L. (1998). Open Systems Interconnect (OSI) Model. JALA: Journal of the Association for Laboratory Automation, 3(1), 28–35. doi: 10.1177/221106829800300108

[14] Zhang, L., Taal, A., Cushing, R., de Laat, C., & Grosso, P. (2022). A risk-level assessment system based on the STRIDE/DREAD model for digital data marketplaces. Int. J. Inf. Secur., 21(3), 509–525. doi: 10.1007/s10207-021-00566-3

[15] Culot, G., Nassimbeni, G., Podrecca, M., &Sartor, M. (2021). The ISO/IEC 27001 information security management standard: literature review and theory-based research agenda. TQM Journal, 33(7), 76–105. doi: 10.1108/TQM-09-2020-0202

[16] Krumay, B., Bernroider, E. W., & Walser, R. (2018). Evaluation of cybersecurity management controls and metrics of critical infrastructures: A literature review considering the NIST cybersecurity framework. In *Secure IT Systems: 23rd Nordic Conference, NordSec 2018, Oslo, Norway, November 28-30, 2018, Proceedings 23* (pp. 369-384). Springer International Publishing.

[17] Chassang, G. (2017). The impact of the EU general data protection regulation on scientific research. ecancermedicalscience, 11. doi: 10.3332/ecancer.2017.709

[18] Jaiswal, S. K., & Dwivedi, A. K. .A Security and Application of Wireless Sensor Network: A Comprehensive Study. 2023 International Conference on IoT, Communication and Automation Technology (ICICAT). IEEE. doi: 10.1109/ICICAT57735.2023.10263644

[19] BenSaleh, M. S., Saida, R., Kacem, Y. H., & Abid, M. (2020). Wireless Sensor Network Design Methodologies: A Survey. J. Sens., 2020. doi: 10.1155/2020/9592836

[20] Priyadarshi, R., Gupta, B., & Anurag, A. (2020). Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues. J. Supercomput., 76(9), 7333–7373. doi: 10.1007/s11227-020-03166-5

[21] Kumar, S., Gupta, S., & Arora, S. (2021). Research Trends in Network-Based Intrusion Detection Systems: A Review. IEEE Access, 9, 157761–157779. doi: 10.1109/ACCESS.2021.3129775

[22] Denning, D. E. (1987). An Intrusion-Detection Model. IEEE Trans. Software Eng., SE-13(2), 222–232. doi: 10.1109/TSE.1987.232894

[23] Krivchenkov, A., Grakovski, A., &Misnevs, B. (2024). Anomaly Detection for Intrusion Detection Systems Using Machine Learning: Experimental Study and Feature Reduction Approach. Reliability and Statistics in Transportation and Communication. Springer. doi: 10.1007/978-3-031-53598-7_11

[24] Ioulianou, P., Vasilakis, V., Moscholios, I., & Logothetis, M. (2018). A signature-based intrusion detection system for the internet of things. *Information and Communication Technology Form*.

[25] Youssef, B., Nada, M., & Regragui, B. (2019). Behavioural analysis approach for IDS based on attack pattern and risk assessment in cloud computing. Int. J. Inf. Comput. Secur.

[26] Ee, S. J., Ming, J. W. T., Yap, J. S., Lee, S. C. Y., & Zahra, F. t. (2023). Active and Passive Security Attacks in Wireless Networks and Prevention Techniques. Authorea Preprints. doi: 10.36227/techrxiv.12972857.v1

[27] Tariq, U., Ahmed, I., Bashir, A. K., & Shaukat, K. (2023). A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review. Sensors, 23(8), 4117. doi: 10.3390/s23084117

[28] Dimitrov, W. (2020). The Impact of the Advanced Technologies over the Cyber Attacks Surface. Artificial Intelligence and Bioinspired Computational Methods. Springer. doi: 10.1007/978-3-030-51971-1_42

[29] Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN COMPUT. SCI. 2, 160 (2021). https://doi.org/10.1007/s42979-021-00592-x

[30] Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., Aljaaf, A.J. (2020). A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science. In: Berry, M., Mohamed, A., Yap, B. (eds) Supervised and Unsupervised Learning for Data Science . Unsupervised and Semi-Supervised Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-22475-2_1

[31] Dike, H. U., Zhou, Y., Deveerasetty, K. K., & Wu, Q. (2018, October). Unsupervised learning based on artificial neural network: A review. In 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS) (pp. 322-327). IEEE.

[32] Li, Y. (2022). Reinforcement learning in practice: Opportunities and challenges. arXiv preprint arXiv:2202.11296.

[33] Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. SN COMPUT. SCI. 2, 420 (2021). https://doi.org/10.1007/s42979-021-00815-1

[34] S. Hayman, "The McCulloch-Pitts model," IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339), Washington, DC, USA, 1999, pp. 4438-4439 vol.6, doi: 10.1109/IJCNN.1999.830886.

[35] Damadi, S., Moharrer, G., Cham, M., & Shen, J. (2023, June). The Backpropagation algorithm for a math student. In *2023 International Joint Conference on Neural Networks (IJCNN)* (pp. 01-09). IEEE.

[36] Yuming Hua, Junhai Guo and Hua Zhao, "Deep Belief Networks and deep learning," Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things, Harbin, 2015, pp. 1-4, doi: 10.1109/ICAIOT.2015.7111524.

[37] Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2020). Fundamental concepts of convolutional neural network. *Recent trends and advances in artificial intelligence and Internet of Things*, 519-567.

[38] Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... & Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*.

[39] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306.

[40] Schmidgall, S., Achterberg, J., Miconi, T., Kirsch, L., Ziaei, R., Hajiseyedrazi, S., &Eshraghian, J. (2023). Brain-inspired learning in artificial neural networks: a review. *arXiv preprint arXiv:2305.11252*.

[41] Lederer, J. (2021). Activation functions in artificial neural networks: A systematic overview. *arXiv preprint arXiv:2101.09957*.

[42] Tokime, R., Maldague, X., & Perron, L. (2019). Automatic Defect Detection for X-Ray inspection: Identifying defects with deep convolutional network. *Proceedings of the Canadian Institute for Non-destructive Evaluation (CINDE), Edmonton, AB, Canada*, 18-20.

[43] Bre, F., Gimenez, J. M., &Fachinotti, V. D. (2018). Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, *158*, 1429-1441..

[44] O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

[45] Van Houdt, G., Mosquera, C. &Nápoles, G. A review on the long short-term memory model. ArtifIntell Rev 53, 5929–5955 (2020). https://doi.org/10.1007/s10462-020-09838-1

[46] Sun, T., Yang, C., Han, K., Ma, W., & Zhang, F. (2020). Bidirectional spatial–temporal network for traffic prediction with multisource data. *Transportation research record*, *2674*(8), 78-89.

[47] Du, G., Wang, Z., Gao, B., Mumtaz, S., Abualnaja, K. M., & Du, C. (2020). A convolution bidirectional long short-term memory neural network for driver emotion recognition. *IEEE Transactions on Intelligent Transportation Systems*, *22*(7), 4570-4578.

[48] Su, T., Sun, H., Zhu, J., Wang, S., & Li, Y. (2020). BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset. IEEE Access, 8, 29575–29585. doi: 10.1109/ACCESS.2020.2972627

[49] Chaibi, N., Atmani, B., & Mokaddem, M. (2020). Deep Learning Approaches to Intrusion Detection: A new Performance of ANN and RNN on NSL-KDD. ISPR '20: Proceedings of the 1st International Conference on Intelligent Systems and Pattern Recognition. Association for Computing Machinery. doi: 10.1145/3432867.3432889

[50] Qazi, E.-u.-H., Imran, M., Haider, N., Shoaib, M., & Razzak, I. (2022). An intelligent and efficient network intrusion detection system using deep learning. Comput. Electr. Eng., 99, 107764. doi: 10.1016/j.compeleceng.2022.107764

[51] Moraboena, S., Ketepalli, G., & Ragam, P. (2020, August 01). A Deep Learning Approach to Network Intrusion Detection Using Deep Autoencoder. | Revue d'Intelligence Artificielle | EBSCOhost. doi: 10.18280/ria.340410

[52] Choudhary, S., &Kesswani, N. (2020). Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT. Procedia Comput. Sci., 167, 1561–1573. doi: 10.1016/j.procs.2020.03.367

[53] Gowdhaman, V., & Dhanapal, R. (2022). An intrusion detection system for wireless sensor networks using deep neural network. Soft Comput., 26(23), 13059–13067. doi: 10.1007/s00500-021-06473-y

[54] Kagade, R. B., &Jayagopalan, S. (2022). Optimization assisted deep learning based intrusion detection system in wireless sensor network with two-tier trust evaluation. Int. J. Network Manage., 32(4), e2196. doi: 10.1002/nem.2196

[55] Riyaz, B., & Ganapathy, S. (2020). A deep learning approach for effective intrusion detection in wireless networks using CNN. Soft Comput., 24(22), 17265–17278. doi: 10.1007/s00500-020-05017-0

[56] Pankaj R. Chandre, Dr. P. N. M. (2021). Intrusion Prevention Framework for WSN using Deep CNN. TURCOMAT, 12(6), 3567–3572. doi: 10.17762/turcomat.v12i6.7145

[57] Alruhaily, N. M., & Ibrahim, D. M. (2021). A multi-layer machine learning-based intrusion detection system for wireless sensor networks. *International Journal of Advanced Computer Science and Applications*, *12*(4), 281-288.

RESUME

Hayder Abdulameer Yousif AL-IESSA began his academic journey in Baghdad, Iraq. He pursued his undergraduate studies at Baghdad College of Economic Sciences University in the 2010-2011. In 2021, He moved to Karabuk, Turkey, to undertake postgraduate studies. He enrolled in a Master of Science program in Computer Engineering at Karabuk University.