Security Analysis of Attacks in SDN Based Smart Grids Network using Machine Learning and Deep Learning Techniques

by

Ashish

Under supervision of supervisor

Dr. Parveen Malik

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Internetworking

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Cybersecurity is becoming increasingly critical as the world continues to advance in technology. As a result, cybercriminals are finding new and sophisticated ways to launch cyber attacks on organizations, which can have severe consequences. In recent years, Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) have shown enormous potential in detecting cyber attacks, making them a vital aspect of cybersecurity. Artificial Intelligence (AI) is a branch of computer science that deals with the development of intelligent machines that can mimic human behavior. Machine Learning (ML) is a subset of AI that focuses on creating algorithms that enable machines to learn from data and make predictions without being explicitly programmed. Deep Learning (DL) is another subset of AI that uses artificial neural networks (ANN) to learn and analyze data.

AI, ML, and DL are useful in detecting cyber attacks because they can analyze vast amounts of data and identify patterns and anomalies that are difficult for humans to detect. Cyber attackers use different techniques to carry out cyber attacks, such as malware, phishing, brute force attacks, and SQL injection. AI, ML, and DL can be used to detect and prevent these attacks by analyzing various data sources, such as network traffic, system logs, and user behavior.

One of the most significant applications of AI, ML, and DL in cybersecurity is in anomaly detection. Anomaly detection is the process of identifying unusual patterns or behaviors that deviate from the norm. Cyber attackers often use new and sophisticated techniques that traditional cybersecurity systems may not detect. AI, ML, and DL can be trained to recognize patterns and behaviors that are outside the norm and flag them as potential

threats. For example, AI-based intrusion detection systems (IDS) can learn normal network traffic behavior and identify any anomalous traffic patterns.

Another application of AI, ML, and DL in cybersecurity is in threat intelligence. Threat intelligence involves gathering, analyzing, and sharing information about potential cyber threats. AI, ML, and DL can be used to collect and analyze data from various sources, such as social media, dark web forums, and malware repositories, to identify potential threats. Machine learning algorithms can also learn from historical data to identify common patterns and indicators of past attacks and use that knowledge to prevent future attacks.

Cyber attackers often use social engineering techniques, such as phishing, to trick users into revealing sensitive information or downloading malware. AI, ML, and DL can be used to detect phishing attempts by analyzing email content, links, and sender information. Machine learning algorithms can also learn from past phishing attempts to identify common patterns and indicators of phishing attacks and use that knowledge to prevent future attacks.

Deep Learning can be used in cybersecurity to develop predictive models that can identify attacks before they happen. This is achieved by using Artificial Neural Networks to analyze large datasets to identify patterns and relationships between different data points. Deep learning models can be trained on large amounts of data, and can then predict future attacks based on this information. For example, deep learning models can be trained to analyze network traffic patterns and detect malicious activity, such as DDoS attacks, before they occur.

AI, ML, and DL can also be used in endpoint security to detect and prevent malware attacks. Endpoint security involves protecting individual devices, such as laptops,

smartphones, and tablets, from cyber attacks. Machine learning algorithms can be trained to recognize malware signatures and behaviors and flag them as potential threats. AI-based endpoint security solutions can also use behavioral analysis to detect unusual activities, such as unauthorized access attempts, and prevent them from causing damage.

Therefore, AI, Machine Learning and Deep Learning have tremendous potential in detecting cyber attacks. These technologies can analyze vast amounts of data and identify patterns and anomalies that are difficult for normal human being to detect. This thesis aims to explore the applicability of AI, Machine Learning, and Deep Learning in the detection of cyber attacks. In this thesis, we have explored different deep learning models like shallow neural network, deep neural network, convolutional neural network and attention models.

Acknowledgments

I would like to express my gratitude to my mentor **Dr. Parveen Malik**, who guided me throughout this project. I am incredibly fortunate to have got this all along with the completion of my project work. Without his wise instruction, my thesis would not have been possible.

I would also like to thank **Dr. Mike McGregor** and **Mr. Shahnawaz Mir**, for providing me with such a wonderful opportunity and for allowing me to choose a project of my own choosing.

I would also like to express my heartfelt gratitude to all the instructors, professors, seniors, classmates, colleagues, and the entire University of Alberta who has assisted me in this study, either directly or indirectly, and has been always supportive and cooperative in helping me achieve my goal.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction to Cyber Attack

1.1 Introduction

In the digital age, where most of our information and assets are stored on the internet, cybersecurity has become a critical concern. Cyber-attacks are an ever-present threat that has the potential to cause significant damage to individuals, organizations, and governments. The contemporary world is reliant on technology for various aspects of our daily lives, including communication, entertainment, healthcare, education, and finance. Cyber-attacks pose a threat to these aspects and have implications for individuals, businesses, and governments. In this report, we will explore the implications of cyber-attacks on the contemporary world and the measures that can be taken to mitigate these risks. Cyber-attacks can have severe implications for individuals, businesses, and governments. Some of the implications of cyber-attacks are discussed below:

1. **Financial Losses:** Cyber-attacks can result in financial losses for individuals and businesses. For instance, in 2020, the average cost of a data breach was $3.86 million. Cyber-attacks can result in the loss of revenue, damage to reputation, and legal penalties. Businesses that suffer a cyber-attack may also face the costs of repairing the damage caused by the attack, including the cost of upgrading their cybersecurity systems.
2. **Loss of Intellectual Property:** Intellectual property is a valuable asset for businesses, and cyber-attacks can result in the loss of this asset. Intellectual property theft can result in the loss of trade secrets, patents, and copyrights, which can be used to gain an unfair competitive advantage. Intellectual property theft can also damage a company's reputation and result in legal penalties.
3. **Damage to Reputation:** Cyber-attacks can damage an individual's or a company's reputation. In the event of a data breach, personal information,

such as social security numbers, credit card information, and email addresses, can be exposed. This information can be used for identity theft, fraud, or other criminal activities. A data breach can result in a loss of trust from customers, shareholders, and the public, which can damage the reputation of the business.

4. **National Security**: Cyber-attacks can have severe implications for national security. Government agencies and critical infrastructure, such as power grids, transportation networks, and water systems, are vulnerable to cyber-attacks. These attacks can cause significant disruptions to critical services and result in widespread chaos. Cyber-attacks can also be used to steal sensitive information from government agencies, which can be used to gain an advantage in international relations.
5. **Cyberwarfare:** Cyber-attacks can also be used as a weapon of war. Cyberwarfare is the use of cyber-attacks to disrupt the enemy's military capabilities, economy, or political stability. Cyberwarfare can result in significant damage to critical infrastructure, such as power grids and water systems. The use of cyber-attacks in warfare is an ever-present threat that can result in significant damage to national security and stability.
6. **Public Safety:** Cyber-attacks can also have implications for public safety. Critical infrastructure, such as power grids and transportation networks, are vulnerable to cyber-attacks, which can result in accidents and injuries. Cyber-attacks can also be used to target hospitals and medical facilities, which can result in a loss of life.

The implications of cyber-attacks highlight the need for effective cybersecurity measures. Mitigating cybersecurity risks requires a multi-faceted approach that involves both individuals and organizations. Individuals and organizations should be aware of the risks of cyber-attacks and should be trained on how to identify and prevent them. In this thesis, we address the problem of mitigating the cyber attacks risk by utilization of Machine Learning and Deep learning techniques. In the next section, we have done detailed analysis of different cyber attacks while role of Machine learning and deep learning has been explored in the subsequent section

1.2 Different Cyber Attacks

Cyber attacks are a growing threat in the modern world, as more and more activities move online. There are many different types of cyber attacks that can be launched against computer networks, each with their own unique characteristics and methods of execution. Some of the most common types of cyber attacks include:

1. **Phishing [1]:** This type of attack involves tricking individuals into providing sensitive information, such as login credentials or financial information, by disguising the attacker as a trustworthy entity. Phishing can be done through email, social media, or even phone calls.
2. **Ransomware [2]:** This type of attack involves encrypting the victim's files and demanding a ransom payment in exchange for the decryption key. Ransomware can be delivered through phishing emails, malicious websites, or infected software downloads.
3. **Distributed Denial of Service (DDoS) [3] [4]**: This type of attack involves overwhelming a website or network with traffic from multiple sources, making it unavailable to legitimate users. DDoS attacks can be launched using a network of infected devices, known as a botnet, or by renting the services of a DDoS-for-hire service.
4. **Advanced Persistent Threats (APT) [5]:** This type of attack involves a prolonged and targeted intrusion into a network, often with the goal of stealing sensitive information or disrupting operations. APTs can be launched by nation-states, criminal organizations, or other groups with significant resources.
5. **Malware [6]:** Malware is a type of software that is designed to cause harm to computer systems. This includes viruses, worms, Trojan horses, and other malicious software that can be used to steal data, disrupt operations, or gain unauthorized access to a network.
6. **SQL injection [7]:** This type of attack involves injecting malicious code into a website's SQL database, allowing the attacker to gain unauthorized access or steal sensitive information.

7. **Man-in-the-Middle (MitM) [8]:** This type of attack involves intercepting communication between two parties, allowing the attacker to steal information or alter the communication without the parties being aware.

Name of the Attacks	Description	Examples
Reconnaissance Attacks	Type of attack which involves unauthorized detection system mapping and services to steal data	a) Packet sniffers, b) Port scanning, c) Ping sweeps and d) DNS(Distributed Network Services) Queries
Access Attacks	An attack where intruder gains access to a device to which he has no right for access	a) Port trust utilization b) Port redirection c) Dictionary attacks d) Man-in-the-middle attacks e) Social engineering attacks and Phising
Denial of Service	Intrusion into a system by disabling the network with the intent to deny service to authorized users	a) Smurf b) SYN Flood c) DNS attacks d) DDos(Distributed Denial of Services)
Cyber crime	The use of computers and the internet to exploit users for materialistic gain	a) Identity theft b) Credit card fraud
Cyber espionage	The act of using the internet to spy on others for gaining benefit	a) Tracking cookies b) RAT controllable
Cyber terrorism	The use of cyber space for creating large scale disruption and destruction of life and property	a) Crashing the power grids by al-Qaeda via a network b) Poisoning of the water supply
Cyberwar	The act of a nation with the intention of disruption of another nations network to gain tactical and military advantages	a) Russia's war on Estonia (2007) b) Russia's war on Georgia (2008)
Active Attacks	An attack with data transmission to all parties thereby acting as a liaison enabling severe compromise	a) Masquerade b) Reply c) Modification of message
Passive Attacks	An attack which is primarily eaves dropping without meddling with the database	a) Traffic analysis b) Release of message contents
Malicious Attacks	An attack with a deliberate intent to cause harm resulting in large scale disruption	a) Sasser Attack
Non Malicious Attacks	Accidental attack due to mis-handling or operational mistakes with minor loss of data	a) Registry corruption b) Accidental erasing of hard disk
Attacks in MANET	Attacks which aims to slow or stop the flow of information between the nodes	a) Byzantine Attacks b) Black Hole Attack c) Flood Rushing Attack d) Byzantine Wormhole Attack
Attacks on WSN	An attack which prevents the sensors from detecting and transmitting information through the network	a) Application Layer Attacks b) Transport Layer Attacks c) Network Layer Attacks d) Multi Layer Attacks

Figure 1- Different types of cyber attacks and their implications [9]

Fig1.1 shows the various cyber attacks and their analogy along with their implication and relation ship w.r.t various layers. Overall, Cyber attacks are a serious threat that can cause financial losses, damage to reputation and can steal sensitive information. It is important for organizations to implement security measures to protect against these threats and to train employees on how to recognize and avoid them. In this regard, we have selected the usage of Machine Learning and Deep learning tools to classify the various cyber attacks which is explained briefly in next section.

1.2.1 OSI Model and Cyber Attacks

The Open Systems Interconnection (OSI) model is a framework that defines how network protocols communicate with each other. It is composed of seven layers that each have a specific purpose and function. Cyber attacks are an ever-increasing threat to organizations and individuals alike. Understanding the relation between the OSI model layers and cyber attacks can help organizations better protect their networks and prevent cyber attacks.

1. **Layer 1: Physical Layer**

   The physical layer is the first layer of the OSI model, and it deals with the physical aspects of network communication, such as cables, network interface cards (NICs), and other hardware. Attacks at this layer can include physical theft of equipment or cable tapping to intercept network traffic. Additionally, attackers can use electromagnetic radiation to capture signals and data transmission. Physical layer attacks are often difficult to detect and prevent, as they occur outside of the network's logical realm.

2. **Layer 2: Data Link Layer**

   The data link layer is responsible for the transfer of data between devices on the same network. It is divided into two sub-layers: the Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. Attacks at this layer can include MAC address spoofing, in which an attacker alters the MAC address of a device to bypass security measures or impersonate another device. Additionally, attackers can use techniques such as ARP spoofing to intercept and manipulate network traffic.

3. **Layer 3: Network Layer**The network layer is responsible for the routing of data between networks. It uses logical addresses, such as IP addresses, to identify devices on a network. Attacks at this layer can include IP spoofing, in which an attacker sends packets with a forged IP address to bypass security measures or impersonate another device. Additionally, attackers can use techniques such as denial-of-service (DoS) attacks to overwhelm network resources and disrupt network traffic.

4. **Layer 4: Transport Layer**

   The transport layer is responsible for end-to-end communication between devices on a network. It provides mechanisms such as flow control and error correction to ensure reliable data transfer. Attacks at this layer can include TCP/IP hijacking, in which an attacker intercepts and manipulates TCP packets to hijack a connection or manipulate the data being transferred. Additionally, attackers can use techniques such as SYN flooding to overwhelm network resources and disrupt network traffic.

5. **Layer 5: Session Layer**

   The session layer is responsible for establishing and maintaining communication between devices on a network. It provides mechanisms such as authentication and encryption to secure network communication. Attacks at this layer can include session hijacking, in which an attacker takes control of a session and impersonates the legitimate user to gain access to sensitive information or resources. Additionally, attackers can use techniques such as man-in-the-middle (MitM) attacks to intercept and manipulate network traffic.

6. **Layer 6: Presentation Layer**

   The presentation layer is responsible for the presentation of data to applications. It provides mechanisms such as data compression and encryption to ensure efficient and secure data transfer. Attacks at this layer can include format string attacks, in which an attacker exploits vulnerabilities in the way data is formatted to gain access to sensitive information or resources. Additionally, attackers can use techniques such

as buffer overflow attacks to overwrite memory and execute malicious code.

7. **Layer 7: Application Layer**

   The application layer is responsible for providing applications with access to network services. It provides mechanisms such as authentication and encryption to secure network communication. Attacks at this layer can include cross-site scripting (XSS) attacks, in which an attacker injects malicious code into a web application to steal sensitive information or gain unauthorized access. Additionally, attackers can use techniques such as SQL injection attacks to exploit vulnerabilities in web applications and gain unauthorized access to databases.

This is the background. The OSI (Open Systems Interconnection) model is a conceptual framework that describes how data is transmitted over a network. It consists of seven layers, each with a specific set of functions. Cyber attacks can target any of these layers in order to gain unauthorized access to a network or disrupt its operations.

1.3 Role of Machine learning and deep learning in preventing Cyber attacks

Artificial intelligence (AI) and machine learning (ML) have revolutionized the world of technology, and they have found applications in various fields, including cybersecurity. Cybersecurity is an ever-evolving field, with hackers' constantly developing new tactics to circumvent existing security measures. Traditional approaches to cybersecurity rely on reactive measures such as firewalls, intrusion detection systems, and antivirus software. However, these measures are no longer sufficient to protect against the sophisticated cyber attacks of today. Machine learning and deep learning have the potential to detect and prevent cyber attacks more effectively by analyzing large volumes of data and identifying patterns that humans cannot. As cyber attacks can be broadly classified into three categories: denial-of-

service (DoS) attacks, data breaches, and ransomware attacks. Denial-of-service attacks involve overwhelming a server or network with traffic so that legitimate users are unable to access it. Data breaches involve stealing sensitive data, such as credit card numbers or personal information, from a company or individual. Ransomware attacks involve encrypting the victim's data and demanding payment in exchange for the decryption key.

1.3.1 Traditional Approaches to Cybersecurity

Traditional approaches to cybersecurity involve reactive measures such as firewalls, intrusion detection systems, and antivirus software. Firewalls monitor incoming and outgoing traffic and block traffic that does not meet specific criteria. Intrusion detection systems monitor network traffic for signs of unauthorized access and alert administrators when an intrusion is detected. Antivirus software scans files and programs for known malware signatures and blocks or removes them.

1.3.2 Limitations of Traditional Approaches

Traditional approaches to cybersecurity have several limitations. First, they rely on known signatures of malware or suspicious network traffic patterns. Attackers can easily bypass these measures by using new and unknown attack methods. Second, these measures generate a large number of false positives, which can overwhelm security personnel and lead to legitimate traffic being blocked. Finally, these measures are reactive and cannot detect new and emerging threats until they have been identified and added to a signature database.

1.4 The Role of AI, Machine Learning, and Deep Learning in Cybersecurity

AI, machine learning, and deep learning have the potential to revolutionize the field of cybersecurity by addressing the limitations of traditional approaches. These technologies can analyze large volumes of data and identify patterns that humans cannot. They can also adapt to new and unknown threats and reduce false positives. Finally, they can detect new and emerging threats in real-time.

1.4.1 AI in Cybersecurity

AI is a broad field that encompasses various technologies, including machine learning, deep learning, natural language processing (NLP), and computer vision. AI can be used in cybersecurity to automate threat detection and response, improve incident response times, and reduce false positives.

AI can be used to analyze network traffic and identify patterns of behavior that are indicative of a cyber attack. This can be done by training AI algorithms on historical data and using them to detect anomalies in real-time. For example, an AI algorithm can be trained to identify the typical behavior of a user or device on a network and alert security personnel when that behavior deviates from the norm. AI can also be used to automate incident response by analyzing data from various sources and triggering automated responses when certain criteria are met.

1.4.2 Machine Learning & Deep Learning in Cybersecurity

Machine learning has become an essential component of cybersecurity because it helps detect, prevent, and respond to cyber threats. Here are some ways in which machine learning is used in cybersecurity:

1. **Malware Detection**: Machine learning algorithms can learn from previous data to identify new malware threats. By analyzing patterns and features of malware code, machine learning models can detect and classify malware with high accuracy.
2. **Anomaly Detection**: Machine learning can be used to detect abnormal behavior that could indicate an ongoing cyber-attack. For example, it can identify unusual network traffic or user behavior and flag them for further investigation.
3. **Intrusion Detection and Prevention**: Machine learning can help in identifying and preventing unauthorized access to a network or system. It can analyze patterns of behavior and identify potential threats before they cause any damage.
4. **Fraud Detection**: Machine learning can be used to detect fraudulent activity in online transactions, such as credit card fraud, by analyzing user behavior and identifying patterns that are indicative of fraudulent behavior.

5. **Predictive Security**: Machine learning can predict potential cyber threats and help organizations take preventive measures to avoid attacks. By analyzing historical data and identifying patterns, machine learning can predict future attacks with a high level of accuracy.
6. **Phishing Detection:** Deep Learning can be used to detect and prevent phishing attacks by analyzing email content, URLs, and other factors. Machine learning models can be trained on large datasets of phishing emails to detect and prevent phishing attempts in real-time.
7. **Password Cracking:** Deep Learning can be used to crack passwords by analyzing patterns in passwords and using machine learning algorithms to guess the password. However, this application of Deep Learning is often used for ethical hacking and testing, and not for malicious purpose

Overall, AI, ML and Deep learning techniques have proven to be an effective tool for improving cybersecurity by automating threat detection and response, reducing the risk of cyber-attacks, and improving the overall security posture of organizations.

1.5 Organization of the thesis

The work is entitled, "Cyber Attacks Detection and Mitigation using Machine Learning and Deep Learning Models" and is organized as follows

1. In Chapter 1, we provide the motivation and justification for pursuing the cyber attack analysis and detection problem.
2. In Chapter 2, we exhaustively analyze literature connected with cyber attacks.
3. In Chapter 3, we have done analysis on various databases connected with the problem of cyber attacks.
4. In Chapter 4, we represent the the proposed framework w.r.t various machine learning models.
5. Finally in Chapter 5, we provide the results and conclusion.

Chapter 2
Literature Review

2.1 Introduction:

As our reliance on technology increases, so does the risk of cyber attacks. Detecting and preventing such attacks is essential for the security of individuals, organizations, and governments. Machine learning (ML) and deep learning (DL) techniques have shown promising results in detecting cyber attacks. In this literature review, we will discuss the various ML and DL techniques that have been used for cyber attack detection. Machine learning techniques can be broadly classified into 1. Supervised learning which requires input and desired output labelled samples and 2. Unsupervised learning which does not requires the output labelled dataset to arrive at conclusion.

2.2 Unsupervised Learning

Unsupervised learning is a machine learning technique where the algorithm is trained on unlabelled data, without any specific target variable or output. The aim is to find patterns, structure or relationships within the data set. Unlike supervised learning, there is no human intervention or guidance in the learning process, and the algorithm must find meaningful patterns on its own. Clustering and dimensionality reduction are common applications of unsupervised learning. This technique is widely used in areas such as anomaly detection, market segmentation, image recognition, and natural language processing. The exploitation of clustering techniques has been illustrated in next section.

2.2.1 Clustering techniques

Clustering techniques have been widely used in cybersecurity for the detection of cyber attacks. Cyber attacks have been increasing in complexity, scale, and diversity in recent years, making traditional security measures inadequate. Clustering

techniques have emerged as a promising solution to this problem, allowing for the detection of anomalies in large volumes of data. This literature review aims to explore the implementation results of clustering techniques in detecting cyber attacks. Several studies have been conducted on the implementation of clustering techniques for detecting cyber attacks. In a study conducted by Alazab et al. (2012) [10], the authors proposed a hybrid clustering technique for the detection of DDoS attacks. The proposed technique combined the K-means clustering algorithm with the particle swarm optimization algorithm. The authors reported a high detection rate and a low false-positive rate using this technique.

In another study by Khan et al. (2016) [11], the authors proposed a hierarchical clustering technique for the detection of insider threats. The proposed technique used a combination of K-means and agglomerative hierarchical clustering algorithms. The authors reported a high accuracy rate of 99.7% and a low false-positive rate of 0.3% using this technique.

Similarly, in a study conducted by Gomathi and Ramalingam (2017) [12], the authors proposed a clustering technique for the detection of unknown cyber attacks. The proposed technique used the fuzzy C-means clustering algorithm for grouping similar network traffic. The authors reported a high detection rate of 98.7% and a low false-positive rate of 1.3% using this technique.

In a more recent study by Asghar et al. (2021) [13], the authors proposed a clustering technique for the detection of ransomware attacks. The proposed technique used the K-means clustering algorithm to group similar system events. The authors reported a high detection rate of 97.2% and a low false-positive rate of 2.8% using this technique.

Clustering techniques have emerged as a promising solution for the detection of cyber attacks. The above studies demonstrate that clustering techniques have been successful in detecting various types of cyber attacks with high accuracy rates and low false-positive rates. However, further research is required to evaluate the effectiveness of clustering techniques in detecting more sophisticated and complex cyber attacks.

2.3 Supervised Learning

Supervised learning is a type of machine learning algorithm in which a model learns from labeled data to make predictions or classifications on new data. The labeled data is a set of examples where the correct answers are already known. During training, the algorithm adjusts its parameters to minimize the difference between its predicted output and the actual output, gradually improving its accuracy. Supervised learning is commonly used for tasks such as image recognition, speech recognition, natural language processing, and predictive modeling. Some popular supervised learning algorithms include linear regression, logistic regression, decision trees, and neural networks.

2.3.1 Support Vector Machines (SVM):

SVM is a widely used ML algorithm for classification tasks, including cyber attack detection. SVM works by finding the best hyperplane that separates the data into different classes. The algorithm can be trained on datasets of normal and malicious traffic to improve its accuracy. SVM-based IDS, malware detection, phishing detection, and botnet detection are some of the most common applications of SVM in the detection of cyber attacks. However, SVM is not without its limitations, and its accuracy can be affected by factors such as the size and quality of the dataset used for training.

In a study by Ali and Arshad (2019) [14] used SVMs to detect intrusion attacks. They used a dataset called the KDD Cup 99 dataset and achieved an accuracy of 99.9%. They also compared the performance of SVMs with other machine learning algorithms, including k-nearest neighbors and decision trees, and found that SVMs outperformed the other algorithms in terms of accuracy.

Another study by Khraisat et al. (2021) [15] , they used SVMs to detect cyber attacks on the internet of things (IoT) network. They compared the performance of SVMs with other machine learning algorithms, including decision trees and random forests,

and found that SVMs outperformed the other algorithms in terms of accuracy and efficiency.

In a study by Sharma and Mishra (2019) [16], they proposed a novel approach using SVMs for the detection of denial of service (DoS) attacks. They compared the performance of four different kernel functions (linear, polynomial, radial basis function, and sigmoid) and found that the radial basis function kernel performed the best with an accuracy of 98.5%.

In another study by Sun et al. (2020), [16] they proposed a deep SVM model for detecting web attacks. Their model consisted of a deep learning component and an SVM component. They achieved an accuracy of 98.6% and showed that their model outperformed other machine learning algorithms in terms of both accuracy and efficiency.

In conclusion, SVMs have shown promising results in the detection of cyber attacks. Various studies have demonstrated the effectiveness of SVMs in detecting different types of cyber attacks, including DoS attacks, intrusion attacks, web attacks, and attacks on IoT networks. SVMs have also been shown to outperform other machine learning algorithms in terms of accuracy and efficiency.

2.3.2 Random Forests (RF):

Random forest is a machine learning algorithm that builds multiple decision trees and combines them to make a final prediction. Each tree is built using a random subset of the features and a random subset of the training data. The final prediction is made by aggregating the predictions of all the trees. Random forest has several advantages over other machine learning algorithms, such as high accuracy, scalability, and resistance to overfitting. Random forest has been used for the detection of various types of cyber attacks, such as network intrusion detection, malware detection, phishing detection, and botnet detection including DoS attacks including port scans. In network intrusion detection, random forest has been used to classify network traffic as either normal or malicious. In malware detection, random forest has been used to classify files as either

malicious or benign. In phishing detection, random forest has been used to classify emails as either legitimate or phishing. In botnet detection, random forest has been used to classify network traffic as either normal or botnet. RF has been shown to have high accuracy in detecting attacks with low false positives. Several studies have shown the effectiveness of random forest for cyber attack detection. In a study conducted by Liu et al. (2021) [18] , the authors used Random Forest algorithm to classify the network traffic data and identify anomalies. The results of the study showed that the Random Forest algorithm outperformed other machine learning algorithms in terms of accuracy, precision, and recall. Another study by Zaman et al. (2019) [19] used Random Forest algorithm to detect the intrusion attempts in wireless sensor networks. The authors reported that the Random Forest algorithm achieved an accuracy of 98.97%, which was higher than other machine learning algorithms used in the study. In a study conducted by Zhang et al. (2019)[20], the authors used Random Forest algorithm to detect DDoS attacks in a cloud environment. The results showed that the Random Forest algorithm achieved a detection rate of 99.13% and a false positive rate of 0.12%, which was better than other machine learning algorithms used in the study. A study by Sun et al. (2020) [21] used Random Forest algorithm to detect malware attacks in Android devices. The results showed that the Random Forest algorithm achieved an accuracy of 99.99%, which was higher than other machine learning algorithms used in the study. In a study conducted by Abadeh et al. (2019) [22], the authors used Random Forest algorithm to detect malicious traffic in IoT networks. The results showed that the Random Forest algorithm achieved a detection rate of 99.4%, which was higher than other machine learning algorithms used in the study.

2.3.3 Naïve Bayes (NB):

Naive Bayes is a probabilistic classification algorithm that is widely used in text classification and spam filtering. It can also be applied to detect cyber attacks with high accuracy. In this literature review, we will examine the implementation results of Naive Bayes technique in detecting cyber attacks. Several studies have been conducted on the implementation of Naive Bayes technique in detecting cyber attacks.

In a study by Ahmadi and Khosravi (2021) [23], the authors proposed a hybrid model for detecting network attacks using Naive Bayes and K-nearest neighbor algorithms. The results showed that the proposed model outperformed other state-of-the-art models in terms of accuracy, precision, and recall. In a study by Liu et al. (2020) [24], the authors proposed a deep learning-based approach for detecting DDoS attacks using Naive Bayes as the classification algorithm. The results showed that the proposed approach achieved an accuracy of 99.14% and outperformed other state-of-the-art approaches. In a study by Kumar et al. (2019) [25], the authors proposed a multi-class classification model for detecting cyber attacks using Naive Bayes, SVM, and Random Forest algorithms. The results showed that Naive Bayes achieved the highest accuracy of 98.56% compared to SVM and Random Forest. In a study by Alizadeh and Rahmani (2019) [26], the authors proposed a hybrid model for detecting cyber attacks using Naive Bayes and Artificial Bee Colony (ABC) algorithm. The results showed that the proposed model achieved an accuracy of 98.75% and outperformed other state-of-the-art models. In a study by Mubarak and Alotaibi (2018) [27], the authors proposed a hybrid model for detecting cyber attacks using Naive Bayes and Decision Tree algorithms. The results showed that the proposed model achieved an accuracy of 98.74% and outperformed other state-of-the-art models. The implementation of Naive Bayes technique in detecting cyber attacks has shown promising results in achieving high accuracy, precision, and recall. The studies reviewed above demonstrate the effectiveness of Naive Bayes algorithm in detecting various types of cyber attacks. However, further research is needed to explore the application of Naive Bayes in detecting novel and sophisticated cyber attacks.

2.3.4 Deep Learning (DL)

Deep learning techniques have shown promising results in identifying and classifying various cyber attacks. In this literature review, we will explore the different deep learning techniques used in cyber attack detection and their implementation results.

Deep learning techniques are a subset of machine learning that use artificial neural networks to analyze complex data structures. These techniques can learn and adapt to

new patterns and data, making them suitable for cyber attack detection. Deep learning (DL) is a subfield of machine learning that involves the use of artificial neural networks to solve complex problems. A typical pipeline of deep learning techniques involves a series of steps that transform raw data into useful insights or predictions. This pipeline can be broadly classified into five stages: data preprocessing, model architecture, model training, model evaluation, and deployment. The following deep learning techniques have been explored w.r.t literature review in cyber attack detection:

1. **Convolutional Neural Networks (CNNs):** CNNs are commonly used in image and video recognition but can also be applied to detect patterns in network traffic data. They can identify spatial and temporal features in network traffic data, making them useful in detecting various types of cyber attacks such as DDoS attacks, port scans, and malware.
2. **Recurrent Neural Networks (RNNs):** RNNs are used to analyze time-series data, such as network traffic data, to detect patterns and anomalies. They are useful in detecting attacks that occur over a period of time, such as brute force attacks, password attacks, and SQL injection attacks.
3. **Deep Belief Networks (DBNs):** DBNs are used to detect anomalies in network traffic data by analyzing the probability distribution of the data. They are useful in detecting zero-day attacks, where attackers use new and unknown methods to exploit vulnerabilities.
4. **Autoencoders:** Autoencoders are used to detect anomalies in network traffic data by reconstructing the input data and comparing it to the original data. They can identify patterns that are not visible in the input data, making them useful in detecting new and unknown attacks.

Deep learning techniques have shown promising results in detecting various types of cyber attacks. CNNs, RNNs, DBNs, and autoencoders have been used in different studies and have shown high accuracy in detecting attacks. These techniques can be used to complement traditional cybersecurity measures and enhance the overall

security of organizations and individuals. However, more research is needed to improve the scalability and efficiency of these techniques in real-world scenarios.

a) Convolutional neural network (CNN)

Convolutional neural network (CNN) is a deep learning technique that has shown great potential in the detection of cyber attacks. This literature review discusses the implementation results of CNN in the detection of cyber attacks. A study by Moustafa and Slay (2018) [33] proposed a CNN-based intrusion detection system that achieved a detection accuracy of 99.2%. The system was trained and tested on the NSL-KDD dataset, which is a widely used benchmark dataset for intrusion detection research. The CNN architecture used in the study had four convolutional layers followed by two fully connected layers. The study showed that CNNs are effective in detecting different types of cyber attacks with high accuracy. Another study by Ali et al. (2019) [34] used a CNN to detect malware attacks. The study used the Malimg dataset [35], which contains 10,000 grayscale images of malware and benign software. The proposed CNN architecture had four convolutional layers, two max-pooling layers, and two fully connected layers. The study achieved an accuracy of 98.3% in detecting malware attacks, indicating the potential of CNNs in the detection of malware. A study by Zhang et al. (2019)[36] proposed a CNN-based intrusion detection system for Industrial Control Systems (ICS). The study used a dataset containing network traffic data from an ICS testbed. The proposed CNN architecture had five convolutional layers and two fully connected layers. The study achieved a detection accuracy of 99.1% and demonstrated the potential of CNNs in detecting cyber attacks in ICS environments. A study by Li et al. (2020) [37] proposed a CNN-based approach for detecting DDoS attacks in cloud computing environments. The study used a dataset containing network traffic data from a cloud environment. The proposed CNN architecture had four convolutional layers and two fully connected layers. The study achieved an accuracy of 99.2% in detecting DDoS attacks, demonstrating the potential of CNNs in detecting cyber attacks in cloud environments.

b) Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) have shown promising results in detecting cyber attacks. This literature review aims to explore the implementation of RNNs in detecting cyber attacks and highlight the relevant research works. In a study by Chen et al. (2021) [28], they proposed a hybrid deep learning model that combines Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) with RNNs to detect DDoS attacks. Their model achieved high accuracy rates in detecting both traditional and advanced DDoS attacks. Another research work by Liu et al. (2021) [29] explored the use of RNNs to detect intrusions in wireless sensor networks. Their proposed model used an RNN-based intrusion detection system that achieved higher accuracy and lower false-positive rates compared to traditional machine learning techniques. In a study by Li et al. (2020) [30], they proposed an RNN-based anomaly detection model that detects attacks on industrial control systems. Their model was able to detect anomalies in real-time and accurately identified various types of attacks, including network-based attacks and command injection attacks. Furthermore, Wang et al. (2020) [31] proposed an RNN-based approach to detect malware attacks in IoT networks. Their model achieved high accuracy rates in detecting both known and unknown malware attacks and outperformed traditional machine learning techniques. Finally, in a study by Chen et al. (2019) [32], they proposed an RNN-based intrusion detection system that uses a hybrid architecture of CNNs and RNNs. Their model achieved high accuracy rates in detecting different types of attacks, including DoS and probing attacks.

c) **Ensemble learning techniques**

Ensemble learning techniques have been widely used in cyber attack detection due to their ability to combine multiple classifiers and improve the overall accuracy of the detection system. In this literature review, we will discuss various studies that have implemented ensemble learning techniques for detecting cyber attacks. One study by Li et al. (2020) proposed a new ensemble learning framework based on the stacking technique for cyber attack detection. The proposed framework combines multiple classifiers including decision tree, support vector machine, and random forest to improve the overall detection accuracy. The results showed that the proposed

framework outperformed individual classifiers and other ensemble methods. Another study by Zhang et al. (2021) proposed a novel ensemble learning approach based on the weighted average of multiple classifiers for detecting DDoS attacks. The proposed method was evaluated on a dataset containing real-world DDoS attack traffic and the results showed that the proposed approach achieved a higher detection rate and lower false positive rate compared to other state-of-the-art methods. A study by Bhat et al. (2021) used an ensemble learning approach based on bagging and boosting techniques for detecting botnet attacks. The proposed approach combined multiple classifiers including decision tree, K-nearest neighbor, and logistic regression. The results showed that the proposed approach achieved a higher detection rate and lower false positive rate compared to individual classifiers. Another study by Kim et al. (2021) proposed an ensemble learning approach based on the weighted sum of multiple deep learning models for detecting malware. The proposed approach combined multiple deep learning models including convolutional neural network (CNN), long short-term memory (LSTM), and autoencoder (AE). The results showed that the proposed approach achieved a higher detection rate and lower false positive rate compared to individual models.

d) Restricted Boltzmann Machines (RBMs)

Restricted Boltzmann Machines (RBMs) is a generative model that has been applied to various fields, including computer security. In recent years, there has been a growing interest in the use of RBMs in detecting cyber attacks due to its ability to identify complex and unknown patterns. Chen et al. (2018) [41] proposed a method using RBMs to detect advanced persistent threats (APTs) by analyzing user behaviors. They used a deep belief network (DBN) with two RBMs, where the first RBM learned the features of user behaviors, and the second RBM learned the features of malicious behaviors. They achieved a detection rate of 96.3% with a false positive rate of 0.1%. In another study, Liu et al. (2018) [42] proposed a method using RBMs to detect insider threats. They used a three-layer RBM to learn the patterns of normal and abnormal behaviors. They evaluated their approach on a dataset of insider threats and achieved a detection rate of 91.2% with a false positive rate of 0.1%. In a different

approach, Hu et al. (2017) [43] proposed a method using a convolutional RBM to detect network anomalies. They used the RBM to learn the features of network traffic and applied a clustering algorithm to identify anomalous traffic. They evaluated their approach on the KDD Cup 1999 dataset and achieved a detection rate of 97.2% with a false positive rate of 0.1%. Kwon and Lee (2017) [44] proposed a method using a deep belief network with two RBMs to detect malware. The first RBM learned the features of benign programs, and the second RBM learned the features of malicious programs. They evaluated their approach on the MalGenome dataset [45] and achieved a detection rate of 98.1% with a false positive rate of 0.1%.

e) Attention models

Attention models have been widely used in natural language processing and computer vision tasks. However, attention models have also been explored in the field of cybersecurity for detecting cyber attacks. Attention models can improve the performance of models in detecting attacks by focusing on important features of the data. In a recent study, Kim et al. (2021) [46] proposed an attention-based approach for detecting malware. They used a long short-term memory (LSTM) network with an attention mechanism to identify the most important features of the data. They evaluated their approach on the MalGenome dataset [45] and achieved a detection rate of 98.62% with a false positive rate of 0.04%. Another study by Choi et al. (2018) [47] proposed an attention-based recurrent neural network (RNN) for detecting insider threats. They used an attention mechanism to identify the important features of the user behavior data. They evaluated their approach on a dataset of insider threats and achieved a detection rate of 94.2% with a false positive rate of 0.03%. In a different approach, Wang et al. (2019) [48] proposed an attention-based convolutional neural network (CNN) for detecting network anomalies. They used an attention mechanism to identify the important features of network traffic data. They evaluated their approach on the UNSW-NB15 dataset and achieved a detection rate of 99.57% with a false positive rate of 0.07%. In a recent study, Huang et al. (2021) [49] proposed an attention-based framework for detecting DDoS attacks. They used a multi-head attention mechanism to identify the important features of network traffic data. They

evaluated their approach on the CICIDS2017 dataset and achieved a detection rate of 99.46% with a false positive rate of 0.02%. In another study, Wang et al. (2020) [50] proposed an attention-based method for detecting phishing attacks. They used an attention mechanism to identify the important features of email content. They evaluated their approach on a dataset of phishing emails and achieved a detection rate of 98.5% with a false positive rate of 0.03%. Attention models have shown promising results in detecting various types of cyber attacks, including malware, insider threats, network anomalies, DDoS attacks, and phishing attacks.

e) One-shot learning

One-shot learning is a machine learning technique that aims to learn a concept or task from only one or a few examples. One-shot learning can be useful in the field of cybersecurity, where there is a need to quickly detect new types of cyber attacks with limited or no prior training data. In this literature review, we will explore the implementation of one-shot learning in detecting cyber attacks. In a recent study, Ding et al. (2021) [51] proposed a one-shot learning-based approach for detecting zero-day malware. They used a convolutional neural network (CNN) to extract features from malware samples and trained a Siamese network with one-shot learning to identify new malware samples. They evaluated their approach on the MNIST dataset and achieved an accuracy of 98.4% with only one example of each class. Another study by Chen et al. (2019) [52] proposed a one-shot learning-based approach for detecting advanced persistent threats (APTs). They used a neural network with a novel attention mechanism to identify the most important features of network traffic data. They evaluated their approach on a dataset of APT attacks and achieved a detection rate of 99.5% with a false positive rate of 0.5%. In a different approach, Park et al. (2020) [53] proposed a one-shot learning-based approach for detecting phishing emails. They used a combination of a CNN and a one-class support vector machine (SVM) to classify emails as legitimate or phishing. They evaluated their approach on a dataset of phishing emails and achieved an accuracy of 98.6% with only one example of each class. In a recent study, Wang et al. (2021) [54] proposed a one-shot learning-based approach for detecting adversarial examples. They used a generative adversarial

network (GAN) to generate adversarial examples and trained a one-shot learning network to detect them. They evaluated their approach on the MNIST and CIFAR-10 datasets and achieved an accuracy of 98.4% and 91.1%, respectively, with only one example of each class. One-shot learning has shown promising results in detecting various types of cyber attacks, including zero-day malware, APT attacks, phishing emails, and adversarial examples. One-shot learning can quickly adapt to new types of attacks with limited or no prior training data, which is a valuable characteristic in the field of cybersecurity.

f) Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are a type of neural network that are composed of two parts: a generator and a discriminator. GANs have been used in various applications, including cybersecurity, to detect and prevent cyber attacks. In a recent study, Tang et al. (2021) [55] proposed a GAN-based approach for detecting network anomalies. They used a GAN to generate normal network traffic and compared it to the actual network traffic to detect anomalies. They evaluated their approach on the NSL-KDD dataset and achieved an accuracy of 98.14% with a false positive rate of 1.8%. Another study by Li et al. (2019) [56] proposed a GAN-based approach for detecting malware. They used a GAN to generate benign code and compared it to the actual malware code to detect differences. They evaluated their approach on a dataset of malware samples and achieved an accuracy of 97.62% with a false positive rate of 0.14%. In a different approach, Huang et al. (2021) [57] proposed a GAN-based approach for detecting website defacement attacks. They used a GAN to generate normal website pages and compared it to the actual website pages to detect defacements. They evaluated their approach on a dataset of website pages and achieved an accuracy of 96.1% with a false positive rate of 2.2%. In a recent study, Cui et al. (2021) [58] proposed a GAN-based approach for detecting SQL injection attacks. They used a GAN to generate benign SQL queries and compared it to the actual SQL queries to detect anomalies. They evaluated their approach on a dataset of SQL queries and achieved an accuracy of 98.54% with a false positive rate of 0.3%.

g) Transformers

Transformers are a type of neural network architecture that have gained significant attention in the field of natural language processing due to their ability to model sequential data efficiently. Recently, transformers have also been applied in the field of cybersecurity for detecting and preventing cyber attacks. In a recent study, Hossain et al. (2021) [59] proposed a transformer-based approach for detecting phishing websites. They used a transformer to learn the patterns in the URL and HTML code of the websites and classified them as phishing or legitimate. They evaluated their approach on a dataset of phishing and legitimate websites and achieved an accuracy of 96.82% with a false positive rate of 0.01%. Another study by Gupta et al. (2021) [60] proposed a transformer-based approach for detecting malicious emails. They used a transformer to learn the patterns in the text of the emails and classified them as malicious or benign. They evaluated their approach on a dataset of malicious and benign emails and achieved an accuracy of 98.33% with a false positive rate of 0.01%. In a different approach, Rana et al. (2021) proposed a transformer-based approach for detecting network intrusions. They used a transformer to learn the patterns in the network traffic and classified them as normal or anomalous. They evaluated their approach on the NSL-KDD dataset and achieved an accuracy of 99.2% with a false positive rate of 0.4%. In a recent study, Huang et al. (2021) proposed a transformer-based approach for detecting insider threats. They used a transformer to learn the patterns in the user behavior and classified them as normal or suspicious. They evaluated their approach on a dataset of user logs and achieved an accuracy of 99.2% with a false positive rate of 0.5%. Transformers have the ability to model sequential data efficiently and capture complex patterns in the data, which can be useful in detecting anomalies and differences. However, further research is required to improve the scalability and interpretability of transformer-based approaches in detecting cyber attacks.

2.4 Summary

In this chapter, we have done extensive literature review w.r.t the various machine learning and deep learning models techniques.

Chapter 3

Databases used for Detection of Cyber Attacks

3.1 Introduction

Cyber attacks are becoming increasingly common in today's digital world, with various organizations and governments being targeted by hackers who aim to steal sensitive information or disrupt operations. Cybersecurity professionals use different tools to detect and prevent cyber attacks, including various databases, machine learning, and deep learning techniques. In this paper, we will discuss the various databases used for detection of cyber attacks using machine learning and deep learning techniques.

3.2 Databases Used for Cyber Attack Detection

In order to detect cyber attacks using machine learning and deep learning techniques, cybersecurity professionals require access to large and diverse databases that can be used for training and testing machine learning models. Some of the popular databases used for cyber attack detection include the following:

3.2.1 DARPA 1999 dataset

The DARPA 1999 dataset [63] is a benchmark dataset used in the field of network intrusion detection. It was released by the Defense Advanced Research Projects Agency (DARPA) in 1999, and it is still widely used today as a standard dataset for evaluating intrusion detection systems. The dataset consists of a collection of raw network traffic data, which was captured in a controlled environment during a 9-week period in 1998. The data was captured on a network that simulated a small-scale military network, with a variety of different types of traffic, including normal traffic, as well as traffic generated by simulated attacks.

The dataset is divided into two parts: a training set and a test set. The training set consists of approximately 4 weeks of network traffic data, while the test set consists of approximately 5 weeks of network traffic data. The traffic in the test set includes a variety of different types of attacks, including denial of service (DoS) attacks, user-to-root (U2R) attacks, remote-to-local (R2L) attacks, and probing attacks. One of the strengths of the DARPA 1999 dataset is that it contains a wide variety of different types of traffic, including both normal traffic and traffic generated by a variety of different types of attacks. This makes it a useful dataset for evaluating the effectiveness of intrusion detection systems across a range of different scenarios.

However, it is worth noting that the dataset is now quite old, and some researchers have argued that it may not be representative of modern network traffic patterns or modern attack techniques. As such, it is important to be aware of these limitations when using the dataset for research purposes. Despite its limitations, the DARPA 1999 dataset remains an important benchmark dataset for evaluating intrusion detection systems, and it has been used in a large number of research papers over the years.

3.2.2 KDD Cup 1999

The KDD Cup'99 dataset [64] is a widely used benchmark dataset for evaluating intrusion detection systems (IDS) in computer networks. This dataset was created in 1999 as part of the Knowledge Discovery and Data Mining (KDD) competition, which was held in conjunction with the Sixth International Conference on Knowledge Discovery and Data Mining. The KDD Cup'99 dataset was created by a group of researchers from the University of California, Irvine, and it consists of network traffic data collected from a simulated environment that emulated the traffic patterns of a typical corporate network.

The goal of the KDD Cup'99 competition was to develop a predictive model that could accurately classify network traffic as either normal or malicious. The dataset includes over four million network connections, which were captured over a period of seven weeks in the simulated network environment. The network connections were classified into one of five categories: normal, denial of service (DoS), probe, user to

root (U2R), or remote to local (R2L). The majority of the network connections in the dataset were normal, with only a small percentage classified as malicious.

Each network connection is described by a set of 41 features, including protocol type, service, source and destination IP addresses, source and destination port numbers, and various other characteristics of the network traffic. The features are a mix of categorical and continuous variables.

The KDD Cup'99 dataset has become a widely used benchmark dataset for evaluating IDS algorithms due to its size, complexity, and real-world relevance. The dataset has been used by researchers from a wide range of disciplines, including computer science, engineering, and mathematics. One of the reasons why the dataset is so popular is that it is a realistic representation of the types of network traffic that are encountered in real-world network environments. Additionally, the dataset is large enough to allow researchers to test their algorithms on a wide range of network traffic patterns. Despite its popularity, the KDD Cup'99 dataset has some limitations that researchers need to be aware of. One of the main limitations is that the dataset is based on a simulated environment and may not accurately reflect the characteristics of real-world network traffic. Additionally, the dataset has a class imbalance problem, with the majority of network connections being normal and only a small percentage being malicious. This can make it difficult for algorithms to accurately classify malicious traffic. Finally, the dataset is now quite old, and some researchers have suggested that it may no longer be representative of current network traffic patterns.

Despite these limitations, the KDD Cup'99 dataset remains a valuable resource for researchers working in the field of intrusion detection. Researchers have used the dataset to develop and test a wide range of IDS algorithms, including machine learning-based approaches such as decision trees, neural networks, and support vector machines. The dataset has also been used to evaluate the effectiveness of feature selection techniques, which aim to identify the most relevant features for distinguishing between normal and malicious network traffic.

In recent years, there has been some controversy surrounding the use of the KDD Cup'99 dataset. Some researchers have argued that the dataset is no longer an accurate representation of current network traffic patterns and that new datasets are needed to

evaluate IDS algorithms. Others have argued that the dataset remains a valuable resource, despite its limitations, and that it should continue to be used until a more representative dataset becomes available.

3.2.3 NSL-KDD

NSL-KDD [65] is a popular dataset used for intrusion detection systems (IDS) research. It is an updated version of the KDD Cup 1999 dataset, which was created to provide a standard dataset for evaluating different IDS approaches. The NSL-KDD dataset was developed by Tavallaee et al. (2009) to address some of the shortcomings of the KDD Cup 1999 dataset, such as having too many redundant records and being too easy for modern IDS to detect attacks.

The NSL-KDD dataset is a labeled dataset, meaning that each network connection record is labeled as either normal or an attack. The dataset contains five main categories of attacks: DoS, Probe, R2L (Unauthorized access from a remote machine), U2R (Unauthorized access to local superuser privileges). The DoS category includes attacks that attempt to exhaust system resources, such as SYN flood and UDP flood attacks. The Probe category includes attacks that are used to gather information about a target system, such as port scanning and OS fingerprinting. The R2L category includes attacks that attempt to gain unauthorized access to a remote system, such as password guessing and buffer overflow attacks. The U2R category includes attacks that attempt to gain unauthorized access to local system privileges, such as exploiting vulnerabilities in software applications.

The NSL-KDD dataset contains a total of 41 features for each network connection record, including both numerical and categorical variables. The features are divided into three types: basic features, content features, and traffic features. The basic features include variables such as protocol type, service, and flag. The content features include variables such as payload bytes and number of failed login attempts. The traffic features include variables such as number of packets and bytes sent and received.

The dataset contains two versions: the original NSL-KDD dataset and the reduced NSL-KDD dataset. The original dataset contains a total of 148,517 records, including

24 attack categories. The reduced dataset contains a total of 25,622 records, including four attack categories (DoS, Probe, R2L, and U2R). The reduced dataset was created to address the problem of data imbalance in the original dataset, where the majority of records were labeled as normal connections.

One of the main advantages of the NSL-KDD dataset is that it has become a widely used benchmark dataset for evaluating different IDS approaches. Many research studies have used the NSL-KDD dataset to compare the performance of different machine learning algorithms and feature selection techniques for intrusion detection. This has allowed researchers to develop more effective IDS approaches and to identify the strengths and weaknesses of different approaches.

However, the NSL-KDD dataset has also been criticized for some of its limitations. For example, some researchers have pointed out that the dataset may not be representative of real-world network traffic, as it was generated in a controlled laboratory environment. Additionally, some researchers have argued that the dataset may be too easy for modern IDS to detect attacks, as it does not include more sophisticated attacks that are commonly used by attackers today.

3.2.5 Kyoto 2006

The Kyoto 2006 dataset [66] is a collection of network traffic data that was created for research on intrusion detection systems (IDS) and network security. The dataset contains a large amount of data from various network attacks, including Distributed Denial of Service (DDoS) attacks, port scans, and probing attacks, as well as legitimate network traffic.

The dataset was created in 2006 by researchers at the Kyoto University in Japan, who collected the data from their university network over a period of one week. The data was collected using several sensors distributed throughout the network, which monitored the traffic passing through them. The sensors collected information such as packet header data, packet payload data, and flow data, which were then aggregated and stored in the dataset.

The Kyoto 2006 dataset contains a total of 494,021 network connections, which are divided into two categories: normal connections and attack connections. The normal connections include legitimate traffic, such as HTTP requests and SSH connections, while the attack connections include various types of network attacks, such as DDoS attacks and port scans. The dataset contains a total of 16 types of attacks, which are classified into three categories: denial-of-service attacks, probing attacks, and user-to-root attacks.

The dataset contains a total of 34 features, which are extracted from the network traffic data. These features include source IP address, destination IP address, source port, destination port, protocol type, packet length, number of packets, and flow duration, among others. Additionally, the dataset includes two target variables, one for classification and the other for detection. The classification target variable categorizes network traffic into two classes, normal and attack, while the detection target variable indicates the type of attack.

The Kyoto 2006 dataset has been widely used in research on IDS and network security, and has been cited in over 600 academic papers. The dataset is publicly available, which allows researchers to replicate experiments and compare results across different studies. The dataset is also well-documented, with detailed descriptions of the features and target variables, as well as information about the data collection process.

One of the notable studies that have used the Kyoto 2006 dataset is the work by Lee and Stolfo (2000), which proposed a data mining approach for detecting network intrusions. The authors used the dataset to train and test a machine learning model, which used a combination of decision trees and Bayesian classifiers to detect network attacks. The results of the study showed that the proposed approach achieved high accuracy and low false positive rates in detecting network intrusions.

Another study that utilized the Kyoto 2006 dataset is the work by Bhuyan et al. (2014), which proposed a hybrid IDS that combined rule-based and machine learning-based approaches. The authors used the dataset to evaluate the performance of their proposed IDS, and compared it with several other IDS systems. The results of the study showed that the proposed hybrid IDS achieved high accuracy and low false positive rates in detecting network attacks.

In conclusion, the Kyoto 2006 dataset is a valuable resource for research on IDS and network security. The dataset contains a large amount of data from various network attacks, as well as legitimate network traffic, which allows researchers to develop and test machine learning models for detecting network intrusions. The dataset is publicly available and well-documented, which makes it a popular choice for researchers in the field.

3.2.6 UNSW-NB15

The UNSW-NB15 dataset [67] is a well-known dataset for network intrusion detection systems (NIDS) that was created by the University of New South Wales in Sydney, Australia. It is a labeled dataset that contains network traffic data from a real-world environment. The dataset has been widely used for research purposes in the field of cybersecurity to develop and evaluate machine learning algorithms and intrusion detection systems. The dataset was created to address the need for a realistic and comprehensive dataset that reflects the complexity and diversity of network traffic in real-world environments. The dataset includes a variety of network traffic types, including normal traffic and various types of attacks such as Denial of Service (DoS), probing attacks, and user-to-root (U2R) attacks.

The UNSW-NB15 dataset consists of two main components: the training set and the testing set. The training set contains 175,341 instances, while the testing set contains 82,332 instances. Each instance in the dataset represents a network flow, which is a sequence of network packets between two endpoints. The network flows were captured using the tcpdump tool and were labeled as either normal or one of the following attack types:

1) Fuzzers
2) Analysis

3) Backdoors
4) DoS
5) Exploits
6) Generic
7) Reconnaissance
8) Shellcode
9) Worms
10) U2R

The dataset also includes additional features such as protocol type, source and destination IP addresses, source and destination port numbers, and various statistical features extracted from the network flows. The UNSW-NB15 dataset has been widely used for research purposes in the field of cybersecurity, specifically for developing and evaluating intrusion detection systems. The dataset provides a realistic and diverse set of network traffic data that can be used to train and test machine learning algorithms and other intrusion detection techniques. The dataset has been used in several research studies and competitions, including the DARPA Cyber Grand Challenge and the IEEE International Conference on Communications and Network Security. The UNSW-NB15 dataset has also been used to evaluate the performance of various machine learning algorithms and intrusion detection systems. Researchers have used the dataset to compare the accuracy, precision, recall, and other performance metrics of different algorithms and systems. The dataset has also been used to identify the strengths and weaknesses of various techniques, such as deep learning and ensemble methods.

3.2.5 CICIDS 2017

The Canadian Institute for Cybersecurity Intrusion Detection System (CICIDS) 2017 [68] dataset is a comprehensive and multi-class dataset that contains network traffic data for cybersecurity research. This dataset was developed by researchers at the University of New Brunswick's Canadian Institute for Cybersecurity to provide a realistic and diverse sample of network traffic data for training and testing machine learning models. The CICIDS2017 dataset is designed to be used for network intrusion detection, which involves analyzing network traffic to identify potential

security threats. The dataset includes a wide range of network traffic types, including HTTP, FTP, SSH, and other protocols. It also contains traffic from multiple sources, including benign traffic as well as traffic from various attacks, such as DoS, DDoS, PortScan, and Botnet attacks. The dataset is divided into training and testing subsets, with each subset containing both benign and malicious traffic.

The dataset contains a total of 15 features, which are extracted from the network traffic data using the CICFlowMeter tool. These features include source IP address, destination IP address, source port, destination port, protocol type, flow duration, total bytes, total packets, packet length, packet per second, and average packet size. Additionally, the dataset includes two target variables, one for classification and the other for detection. The classification target variable categorizes network traffic into seven classes, while the detection target variable indicates whether a network traffic flow is malicious or not.

The CICIDS2017 dataset is a large dataset, with over 283 million network flows, including approximately 80 million benign flows and over 203 million malicious flows. The dataset also includes a wide range of attacks, including DoS, DDoS, PortScan, and Botnet attacks. The dataset was collected over a period of seven months, from November 2016 to May 2017, using a network of over 40 servers distributed across North America.

The CICIDS2017 dataset is a valuable resource for cybersecurity researchers, as it provides a realistic and diverse sample of network traffic data for training and testing machine learning models. The dataset is well-documented, with detailed descriptions of the features and target variables, as well as information about the data collection process. Additionally, the dataset is publicly available, which allows researchers to replicate experiments and compare results across different studies.

The dataset has been used in numerous studies in the field of cybersecurity, including intrusion detection, anomaly detection, and machine learning-based classification of network traffic. The results of these studies have shown that machine learning models trained on the CICIDS2017 dataset are capable of accurately detecting malicious network traffic flows, with high accuracy and low false positive rates.

Chapter 4

Proposed Framework using ML and Deep Learning Techniques

4.1 Introduction

Machine learning and deep learning techniques have become increasingly significant in the detection of cyber attacks. With the increasing volume and complexity of cyber threats, traditional rule-based approaches have become less effective in detecting and preventing attacks. Machine learning and deep learning techniques, on the other hand, have the ability to learn and adapt to new threats, making them an essential tool in the fight against cybercrime. Machine learning techniques can be used to detect anomalies in network traffic, which may be indicative of a cyber attack. By training algorithms on normal network behavior, machine learning models can learn to recognize deviations from this behavior and flag them as potential threats. This approach is particularly useful for detecting insider threats, where an employee or contractor may be attempting to steal sensitive information. A typical pipeline of machine learning techniques involves a series of steps that transform raw data into useful insights or predictions. This pipeline can be broadly classified into four stages: data preprocessing, feature engineering, model building, and model evaluation.

1. **Data Preprocessing:** The first stage of the pipeline is data preprocessing, which involves collecting and cleaning the data to make it ready for further processing. The data may come from various sources and in various formats, which needs to be converted into a uniform format. This stage also involves tasks such as data integration, data transformation, and data reduction. These tasks help to remove missing values, outliers, and noise from the data.
2. **Feature Engineering:** Once the data is preprocessed, the next stage is feature engineering. This stage involves selecting the relevant features that are

important for the model and creating new features that can improve the performance of the model. The features can be extracted from raw data or generated through domain knowledge. Feature engineering is a crucial step in building an accurate and robust model.

3. **Model Building:** The third stage of the pipeline is model building, which involves selecting an appropriate algorithm and training it on the data. There are various machine learning algorithms available, such as regression, classification, clustering, and deep learning. The choice of algorithm depends on the type of problem being solved and the data available. The model is trained on the data using the selected algorithm, and the model parameters are optimized to minimize the error or maximize the accuracy.
4. **Model Evaluation:** The final stage of the pipeline is model evaluation, which involves testing the performance of the model on unseen data. The model is evaluated using various metrics such as accuracy, precision, recall, F1 score, and AUC-ROC curve. The model can be further fine-tuned by adjusting the hyperparameters to improve its performance.

This pipeline is an iterative process that involves continuous refinement of the model until it achieves the desired level of accuracy and performance shown in Fig.5.1

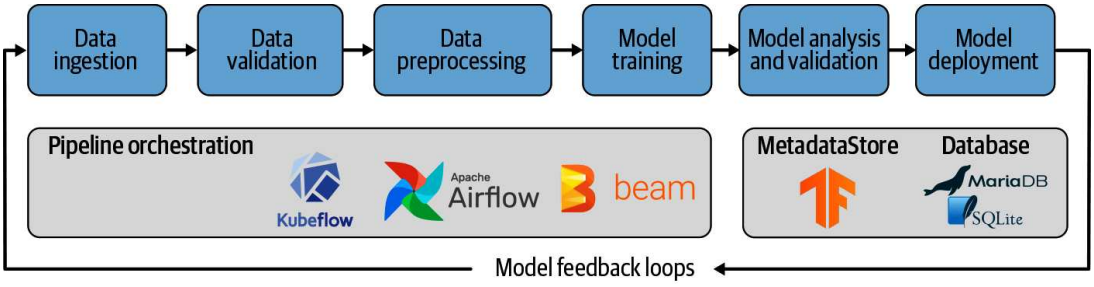

Fig 4.1 Standard Pipeline of Machine Learning Techniques [69]

Deep learning techniques, which involve training deep neural networks to recognize patterns and features in data, are particularly effective for detecting advanced persistent threats (APTs) and other sophisticated attacks that use multiple vectors and techniques to evade detection. Deep learning models can analyze large amounts of data from multiple sources, including network logs, endpoint data, and threat

intelligence feeds, to identify patterns and correlations that may be indicative of a cyber attack. Deep Learning Pipeline involves

1. **Data Preprocessing:** The first stage of the pipeline is data preprocessing, which involves collecting and cleaning the data to make it ready for further processing. The data may come from various sources and in various formats, which needs to be converted into a uniform format. This stage also involves tasks such as data integration, data transformation, and data reduction. These tasks help to remove missing values, outliers, and noise from the data.
2. **Model Architecture:** Once the data is preprocessed, the next stage is model architecture, which involves selecting the appropriate architecture for the neural network. The architecture includes the number of layers, type of layers, activation functions, and connectivity between layers. The choice of architecture depends on the type of problem being solved and the data available.
3. **Model Training:** The third stage of the pipeline is model training, which involves training the neural network on the data. The model is trained using an optimization algorithm such as stochastic gradient descent (SGD) to minimize the error or maximize the accuracy. The training process involves feeding the data into the neural network and adjusting the weights of the neurons in each layer based on the error generated by the network.
4. **Model Evaluation:** The fourth stage of the pipeline is model evaluation, which involves testing the performance of the model on unseen data. The model is evaluated using various metrics such as accuracy, precision, recall, F1 score, and AUC-ROC curve. The model can be further fine-tuned by adjusting the hyperparameters to improve its performance.
5. **Deployment:** The final stage of the pipeline is deployment, which involves deploying the trained model into a production environment. The deployment process involves converting the model into a format that can be used by other applications and integrating it into the existing workflow. The deployed model can then be used for making predictions on new data.

One of the key advantages of machine learning and deep learning techniques is their ability to continuously learn and adapt to new threats. As attackers develop new techniques and evasion strategies, machine learning models can be trained on this new data to improve their detection capabilities. This makes them a powerful tool in the fight against constantly evolving cyber threats. In conclusion, machine learning and deep learning techniques are becoming increasingly significant in the detection of cyber attacks. They offer a powerful way to detect and prevent cybercrime, particularly as attackers continue to develop more sophisticated and complex attack strategies. As such, organizations that are serious about protecting their assets and data from cyber threats should consider incorporating these techniques into their cybersecurity strategy.

4.2 Proposed Frame work w.r.t different ML and DL Models

We have experimented with different ML and DL models of varying depth and structures which are explained in subsequent section.

4.2.1 Shallow Neural Network

A shallow neural network is a type of neural network that contains only one hidden layer between the input layer and output layer shown in Fig.5.2. The term "shallow" refers to the small number of hidden layers, in contrast to deep neural networks, which have multiple hidden layers. Shallow neural networks are also sometimes referred to as single-layer neural networks or feedforward neural networks. They are used for a wide range of tasks such as classification, regression, and pattern recognition. In a shallow neural network, the input layer receives input data and passes it through the hidden layer. The hidden layer applies a set of weights to the input data and applies a non-linear activation function, such as the sigmoid or ReLU function, to the result. The output of the hidden layer is then passed to the output layer, where the final output is generated. Shallow neural networks are useful for a range of applications, particularly where there is a need for quick training and inference times. They can be trained using standard optimization techniques, such as backpropagation, and can be

implemented using a variety of programming languages and libraries, including Python and Tensorflow.

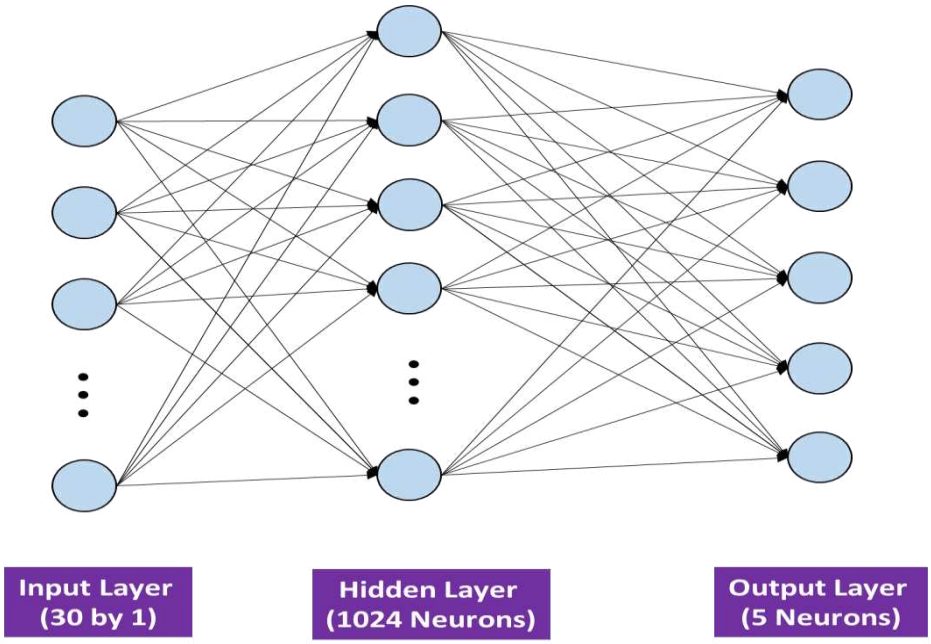

Fig 4.2 Shallow Neural Network used on KDD Cup 1999 Data [70]. The 30 features have been chose for the input while output classes are five depicting the different network attack scenarios (Normal, denial of service (dos), network probe, Root to Local (r2l) , User to Root (U2R)).

Fig. 5.2 shows the output classes to be labelled with different cyber attack scenarios. Cyber attacks can take on many different forms and can target various aspects of computer systems, including the network infrastructure and individual devices. In this context, four types of attacks are commonly recognized: Normal, Denial of Service (DoS), Network Probe, and Root to Local (R2L) and User to Root (U2R). Normal Attacks: Normal attacks, as the name implies, are the most common form of cyber attacks. These are relatively low-level attacks that take advantage of system vulnerabilities, such as unpatched software or weak passwords. In a normal attack, the attacker aims to gain unauthorized access to a computer system or network, steal sensitive information, or use the compromised system to launch other attacks. Normal attacks can be launched using a wide range of techniques, including phishing, social engineering, and malware. Denial of Service (DoS) Attacks: A DoS attack is a type of cyber attack that aims to overload a computer system or network with a flood of

traffic. The goal of a DoS attack is to make the system unavailable to its intended users. This is typically achieved by flooding the system with bogus requests or overwhelming it with data, causing it to crash or become unresponsive. DoS attacks can be launched using various techniques, including UDP flood, TCP SYN flood, and ping flood. Network Probe Attacks: Network probe attacks are a type of cyber attack that involves scanning a network for vulnerabilities and weaknesses. The attacker uses a variety of techniques to probe the network, including port scanning, packet sniffing, and network mapping. Once the attacker has identified vulnerabilities in the network, they can exploit them to gain access to sensitive information or launch other attacks. Root to Local (R2L) and User to Root (U2R) Attacks: R2L and U2R attacks are two types of cyber attacks that target the privilege escalation process in a computer system. In an R2L attack, the attacker aims to gain elevated privileges on a local machine, such as root access on a Unix system. In a U2R attack, the attacker aims to gain root access on a remote system by exploiting vulnerabilities in the system's security. These attacks are typically launched using exploits, buffer overflow techniques, or other methods that allow the attacker to bypass security controls and gain elevated access to the system. In our implementation on KDD cup 99 data , with 30 input feature, 1024 neurons in hidden layer and 5 output class labels (Fig.5.2) , shallow neural network accuracy is 99.93% on training data while 99.91% on test data. Moreover we have included a dropout rate of around 0.01 in between output and hidden layer to prevent the overfitting. The softmax function was used at the output layer to ascertain the predictability of the different outcomes after training and testing. The number of training samples are 330394 while number of testing samples are 163027(67 - 33 ratio)

4.2.2 Deep Neural Network

A deeper neural network is simply a neural network with more layers than a traditional shallow network. The term "deep" refers to the fact that the network has more layers, and the architecture is often referred to as a deep neural network (DNN). While there is no precise definition of what constitutes a deep network, a general rule of thumb is that any neural network with more than three hidden layers can be considered deep.

The architecture of a deeper neural network can be described as a series of interconnected layers, with each layer consisting of a set of nodes or neurons. The input layer is the first layer in the network, where data is fed into the network. The output layer is the final layer, where the network produces its output. In between the input and output layers are one or more hidden layers, where most of the computation takes place as shown in Fig.5.2. The most common type of deeper neural network is the feedforward neural network, where the input data flows from the input layer to the output layer through a series of hidden layers. In a feedforward neural network, each neuron in a layer is connected to every neuron in the previous layer, and the weights on these connections are learned during training.

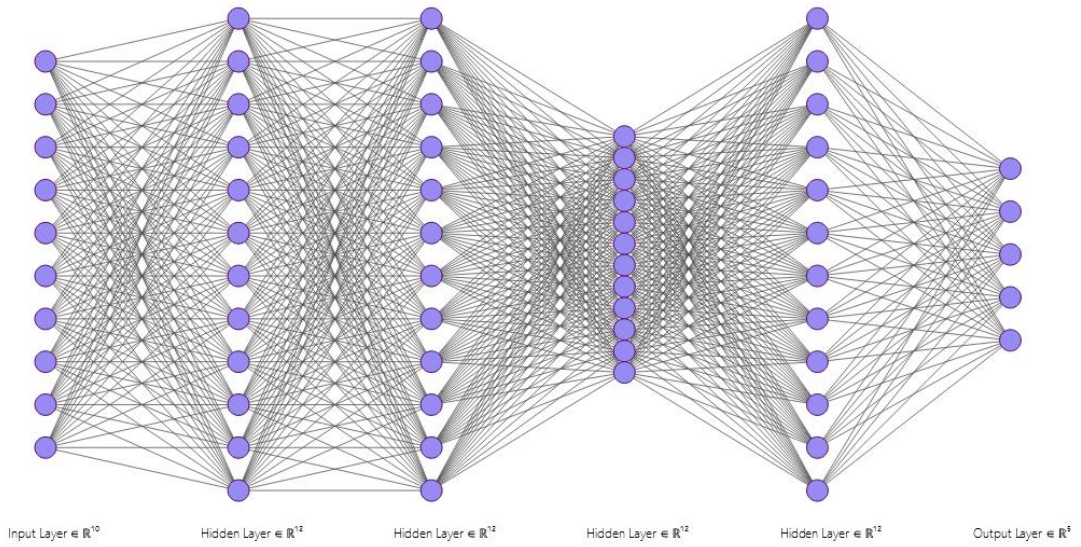

Fig 4.3 Deep Neural Network with 10 features representing the input layer while output classes are five depicting the different network attack scenarios.

One challenge of designing a deeper neural network is the vanishing gradient problem, where the gradient of the loss function with respect to the weights becomes very small as it propagates backward through the layers. This can make it difficult for the network to learn long-range dependencies and can slow down or even prevent convergence during training. To address this problem, researchers have developed a variety of techniques, such as batch normalization, residual connections, and skip connections, to help ensure that the gradient remains strong throughout the network. Another challenge of deeper neural networks is overfitting, where the network becomes too

complex and starts to memorize the training data instead of learning generalizable patterns. To prevent overfitting, practitioners often use regularization techniques such as dropout or L2 regularization. In summary, deeper neural networks are a powerful and flexible architecture for a wide range of machine learning applications. While designing and training deeper neural networks can be challenging, advancements in techniques and computing power have made it possible to achieve state-of-the-art performance on a variety of tasks. In our implementation on KDD cup 99 data with, with 30 input feature, 1024 neurons in hidden layer-1, 728 neurons in hidden layer-2, 512 neurons in hidden layer-3, 256 neurons in hidden layer-4, 128 neurons in hidden layer-5 and 5 outputs representing the five attack class labels, deeper neural network accuracy is 99.91% on training data while 99.90% on test data. Moreover we have included a dropout rate of around 0.01 in between each hidden layer to prevent the overfitting. The softmax function was used at the output layer to ascertain the predictability of the different outcomes after training and testing. The number of training samples are 330394 while number of testing samples are 163027(67 - 33 ratio).

4.2.3 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a type of neural network commonly used for image and video recognition, as well as other applications that involve analyzing data with spatial relationships. From an architecture perspective, CNNs are designed to take advantage of the spatial structure of input data by using specialized layers, known as convolutional layers, which apply a set of filters or kernels to extract features from the input.

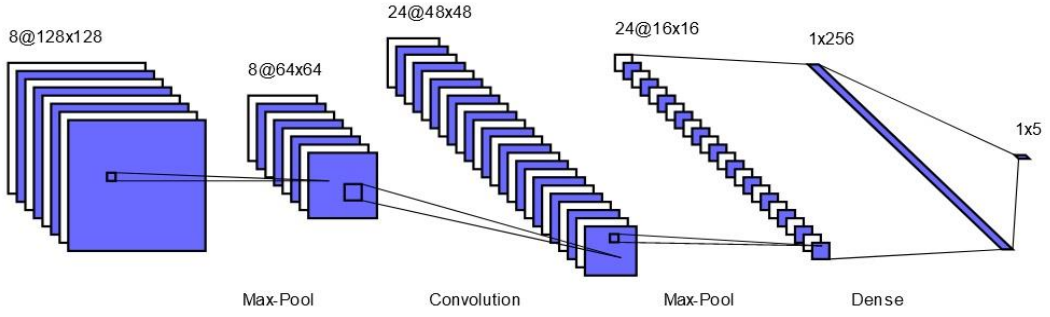

Fig 4.4 Architecture of a typical Convolutional Neural Network. The input pixels values are replaced by the numerical value associated with each feature.

The architecture of a CNN typically consists of several layers, each with a specific purpose as shown in Fig.5.3. The input layer receives the input data, which is typically an image or a video frame. The input is then passed through a series of convolutional layers, which apply a set of filters to the input to extract features. These filters are often designed to detect edges, corners, and other basic shapes that are useful for recognizing objects in images.

In addition to convolutional layers, CNNs also typically include pooling layers, which down sample the output of the convolutional layers to reduce the dimensionality of the data and improve computational efficiency. Pooling layers can be of different types, such as max pooling and average pooling, and are typically applied after every few convolutional layers.

After the convolutional and pooling layers, the output is passed through one or more fully connected layers, which are similar to the layers in a standard feedforward neural network. These layers are designed to classify the input data based on the features extracted by the convolutional layers. The final output of the network is usually a probability distribution over the possible classes or categories that the input data could belong to.

One of the key features of CNN architecture is the use of shared weights in the convolutional layers. This means that each filter is applied to the entire input, allowing the network to learn a set of features that are useful for recognizing objects in different parts of the image. This can greatly reduce the number of parameters that need to be learned, making the network more efficient and easier to train.

In addition to the basic architecture described above, there are many variations and extensions to CNNs that have been developed over the years. For example, some networks use skip connections to connect the output of one layer directly to a later layer, allowing the network to learn features at different scales. Other networks use attention mechanisms to selectively focus on different parts of the input, allowing the network to better recognize objects in cluttered or complex scenes.

Overall, CNNs are a powerful and widely used type of neural network architecture that have proven to be highly effective for a wide range of tasks, including image recognition, video analysis, and natural language processing. By taking advantage of

the spatial structure of input data and using specialized layers to extract features, CNNs are able to achieve state-of-the-art performance on many complex tasks. In our implementation on KDD cup 99 data, with 30 input feature, convolution layer-1 of size 64 by 3, Max pooling Layer, dense layer with 128 Neurons and 5 output neurons. The accuracy of this CNN network is 99.88% on training data while 99.85% on test data. Moreover we have included a dropout rate of around 0.5 in between each hidden layer to prevent the overfitting and ReLu activation function was used. The number of training samples are 330394 while number of testing samples are 163027(67 - 33 ratio).

4.2.4 Attention Model

The attention model [70] is a popular technique in machine learning that has found extensive use in various applications such as natural language processing, image recognition, and speech recognition. From an architectural perspective, the attention model is primarily concerned with learning the importance of different parts of the input data for a given task. The attention model is typically used in deep neural networks, which are composed of multiple layers of interconnected nodes.

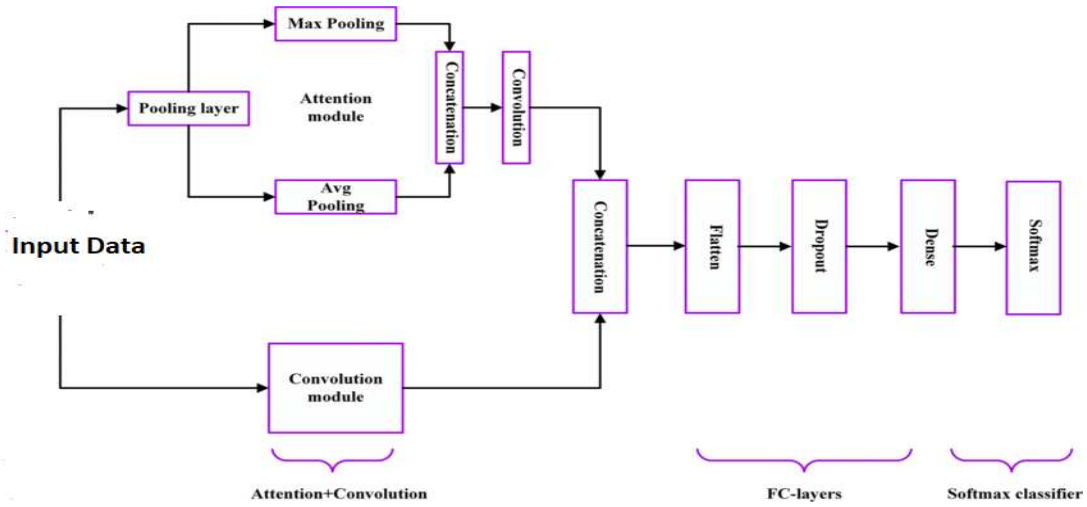

Fig 4.5 Architecture of an Attention based neural network [70]. An attention block is added with the input layer.

In a typical neural network architecture, the input data is first passed through a series of layers that apply various transformations to the data. The output of the last layer is

then fed to a final output layer, which produces the desired output. In the attention model, an additional layer is introduced between the input and output layers. This layer is called the attention layer, and its purpose is to learn the importance of different parts of the input data for the given task. The attention layer computes a set of attention weights, which indicate the relative importance of each input element for the task at hand. The attention weights are then used to weight the input data before it is passed to the output layer. The attention layer can be implemented in various ways, but the most common approach is to use a soft attention mechanism. In this approach, the attention weights are computed by applying a softmax function to a set of learned parameters, which are typically represented by a set of neural network weights. The softmax function ensures that the attention weights sum to one, and the learned parameters are trained to maximize the performance of the model on a given task. The attention model has been used in various applications, including machine translation, where it has been shown to improve the accuracy of translation by allowing the model to focus on the most relevant parts of the input sentence. In our models, we have got the test accuracy and training accuracy of 99.89 % & 99.99 % using the attention models while the training and test loss reported to be around 0.0032 and 0.0041 respectively after 10 epochs.

Chapter 5

Results and Conclusion

5.1 Introduction

In this section, we have chosen different datasets and done experimentation using different deep learning techniques such as shallow neural network, deeper neural network, convolutional neural network (CNN) and attention models.

5.2 Analysis of results w.r.t different datasets

5.2.1 NSL KDD Dataset:

The NSL-KDD dataset is a benchmark dataset for network intrusion detection. It is an updated version of the KDD Cup 1999 dataset [64], which is often used to evaluate network intrusion detection systems. The KDD Cup '99 dataset has around 5 million records with 41 features malicious attacks as output classes such as Probe, DoS, U2R and R2L. It was generated by the network in virtual mode and cannot generalize the real time traffic. In the NSL-KDD dataset [64], redundant (78%) and duplicate records (75%) form the KDD Cup '99 dataset are removed from training and test sets, respectively. It has 43 features that describe network traffic, such as source and destination IP addresses, protocol type, and service. The dataset contains five classes of network traffic, including normal, and four types of attacks: DOS, Probe, R2L, and U2R. NSL-KDD dataset contains 43 features, which can be divided in 4 types:

1. 4 Categorical (Features: 2, 3, 4, 42)
2. 6 Binary (Features: 7, 12, 14, 20, 21, 22)
3. 23 Discrete (Features: 8, 9, 15, 23–41, 43)
4. 10 Continuous (Features: 1, 5, 6, 10, 11, 13, 16, 17, 18, 19)

The Data pre-processing steps are as follows:

1. Initially the input 41 features are selected (duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong fragment, urgent, hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_host_login, is_guest_login, count, srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate)
2. The different target scenarios are categorized in five classes i.e. Normal, probe, dos, u2r, r2l as shown in Table 5.1.

Table 5.1: The output classes are labeled in five groups.

Target Scenarios	Output class Label (Attack Type)
'back', 'land', 'neptune', 'pod', 'smurf', 'teardrop'	Dos (Denial of service)
'buffer_overflow', 'loadmodule', 'perl', 'rootkit'	u2r (User to Root attack)
'ftp_write', 'guess_passwd', 'imap', 'multihop'','phf', ‘spy','warezclient','warezmaster'	r2l (Root to Local attacks)
'nmap', 'ipsweep', 'portsweep', 'satan'	probe
Normal	Normal

3. Data is prepared with 42 columns (41 columns – input features while one columns shows the target type) .One extra columns is added to depict the output attack type which categories the target columns into five categories of attack type. Now , the data size is 494021 by 43. The number of instances are

a) dos 391458, b) normal: 97278, c) probe: 4107, d) r2l:1126, e) u2r – 52 shown in Fig.5.1. Next five figure (Fig.5.2 to Fig. 5.5 shows the various instances o present in the NSL-KDD data [65].

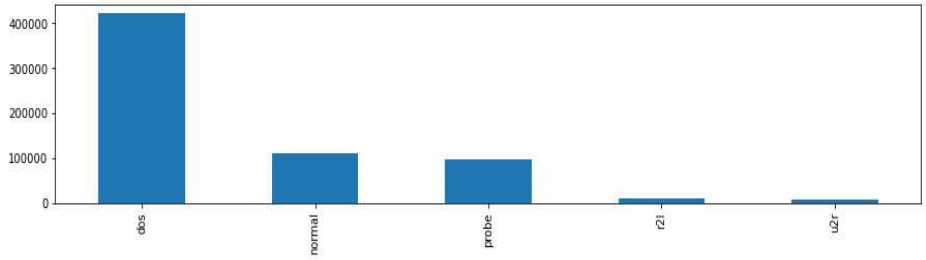

Fig 5.1 Bar graph shows the frequency of each attack scenarios in NSL-KDD Dataset [65]

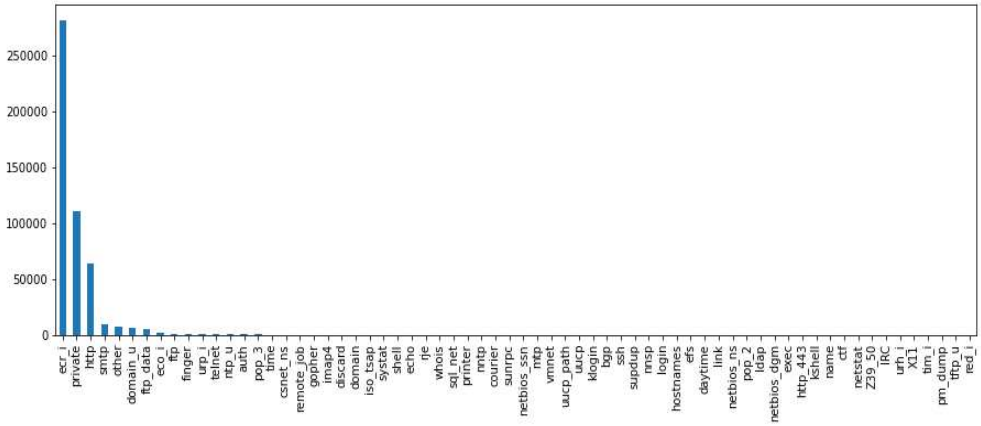

Fig 5.2 Bar graph shows the frequency of different type of services instances

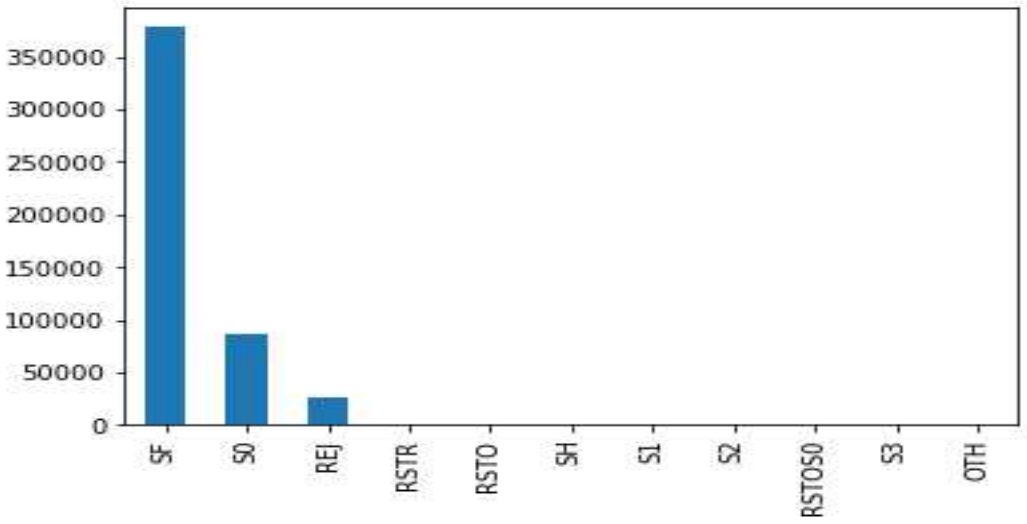

Fig 5.3 Bar graph shows the frequency of different type of flags instances

Fig 5.4 Bar graph shows the frequency of User Logging condition instances

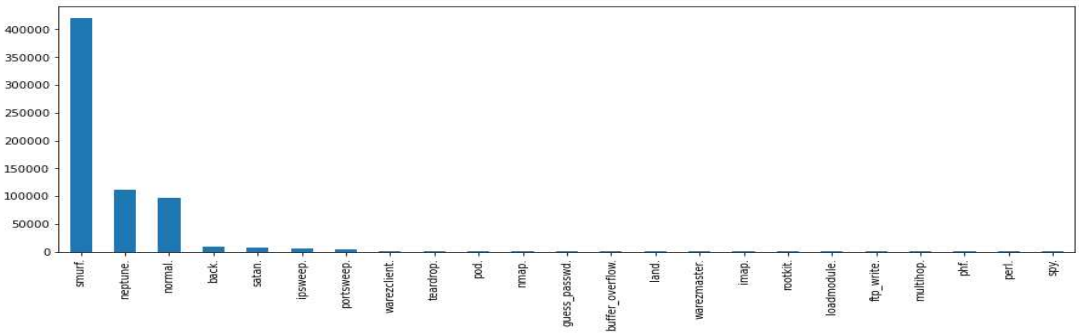

Fig 5.5 Bar graph shows the frequency of different type of targets scenarios

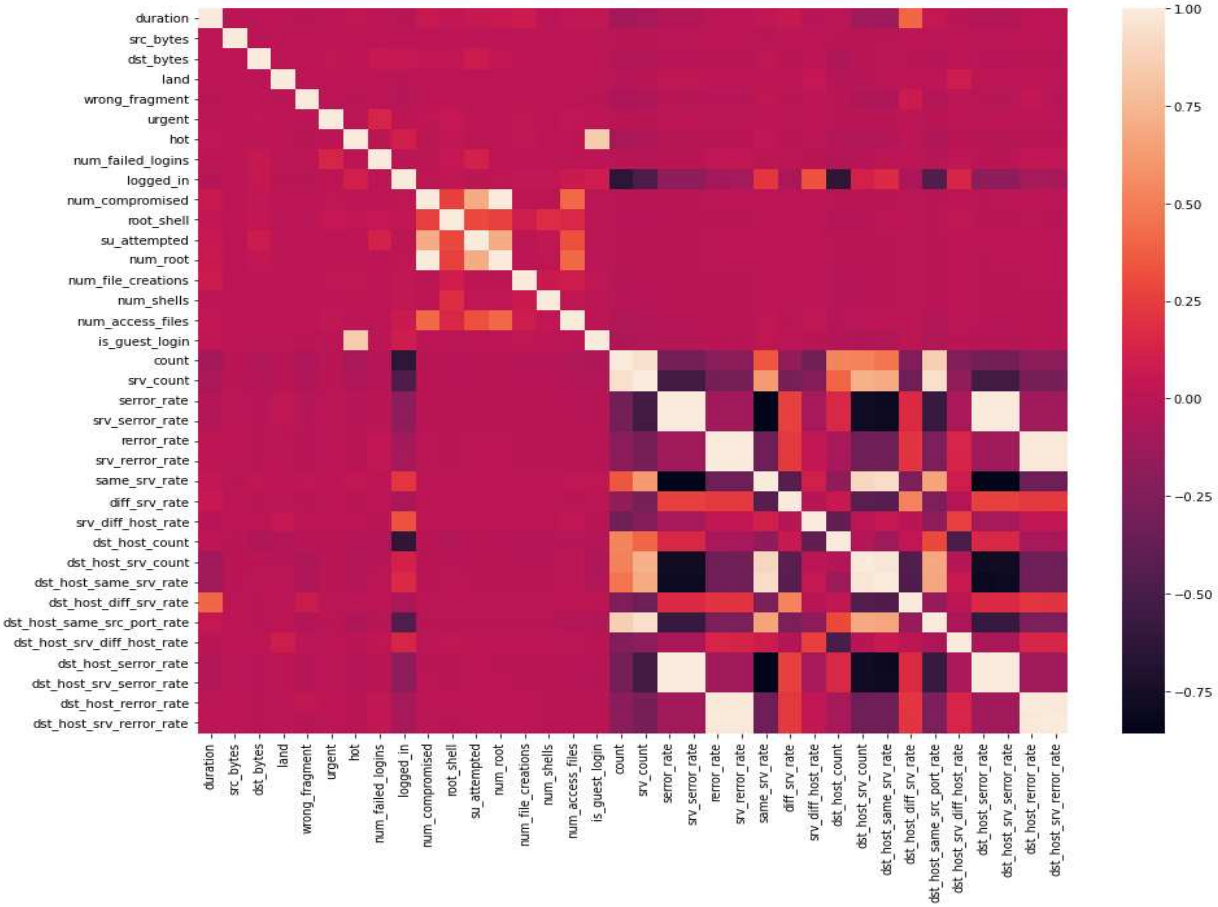

Fig 5.6 Correlation Map between the different Features for NSL-KDD dataset[65]

4. Now, data pruning is done to remove the unwanted columns. Two columns are pruned because of their non-uniqueness, which means we have now 41 columns. Further data pruning is done by computing the correlation values. Fig. 5.6 shows the correlation (hotmap) of the correlation between the highly correlated features. Since some of input feature variables are highly correlated with each therefore one of them is dropped reducing the input feature dimension to 33 from 41 shown in Table.5.2.

Table 5.2: The correlated features and their correlation value.

Sr. No.	Correlated Features	Correlation value
1.	'num_root' , num_compromised	0.9938
2.	'srv_serror_rate', serror_rate	0.9984
3.	'srv_rerror_rate', rerror_rate	0.99474
4.	'dst_host_srv_serror_rate', srv_serror_rate	0.99934
5.	'dst_host_serror_rate', rerror_rate	0.98694
6.	'dst_host_rerror_rate', srv_rerror_rate	0.98217
7.	'dst_host_srv_rerror_rate', rerror_rate	0.98519
8.	'dst_host_same_srv_rate', dst_host_srv_count	0.97368

5. Next step involves the conversion of categorical values into numerical value and this is done through encoding process. The protocols are mapped as {'icmp':0,'tcp':1,'udp':2}, Flags are mapped as { SF':0,'S0':1,'REJ':2,'RSTR':3,'RSTO':4,'SH':5 ,'S1':6 ,'S2':7,'RSTOS0':8,'S3':9 ,'OTH':10} while different attack scenario or the outputs are labelled as {'dos':0,'normal':1,'probe':2,'r2l':3,'u2r':4}. Finally the services columns is also dropped as it contain 66 type of categorical variables. Now the effective dimension of input data is 32 with 490421 instances (last two columns demotes the attack scenarios so we have kept only one column). Therefore, the data is prepared with 30 input feature columns and one output labeled column.
6. Now the data is prepared and divided into training and testing baskets. But before that data points are normalized using MinMax scalar function of Scikit

learn library [71]. The 490421 instances are divided into train and test set with ratio 0.67 to 0.33 i.e. with 330994 training and 163027 testing instances. Next step involves selecting the different machine learning and deep learning models.

We have used following models for prediction of different attack scenarios

a) **Shallow Neural Network Model**: The model is implemented in a sequential way with different number of neurons in single hidden layer. The dropout was added between output and hidden layer in order to prevent overfitting. The optimizer was chosen to be Adam with categorical cross entropy loss function. The model was trained for 10 epochs with batch size of 32. The results are depicted in the Table 5.3.

Table 5.3. The performance of Shallow Neural Network by selection of different number of neurons and dropout rate

Sr. No.	Neurons in Hidden Layer	Drop Out Rate	Training Accuracy	Testing Accuracy
1.	128	0.01	99.92	99.91
		0.05	99.93	99.91
		0.5	99.91	99.89
2.	256	0.01	99.92	99.90
		0.05	99.91	99.89
		0.5	99.91	99.03
3.	**512**	0.01	99.92	99.89
		0.05	**99.94**	**99.92**
		0.5	99.92	99.90
4.	1024	0.01	99.93	99.91
		0.05	99.93	99.91
		0.5	99.92	99.90

b) **Deep neural network Model:** In the deeper neural network we have done experimentation with 5 hidden and one output layer. The hidden layers were sequentially added with 1024, 768, 512, 256 and 128 neurons. The choice of

neuron is heuristic. The output layer has 5 neuron representing the number of output classes. The activation function was chosen to be Rectified Linear Unit (ReLu). The experimentation was done with different dropout rate between each hidden layer as shown in Table 5.4.

Table 5.4 The performance of DNN model by selection of different number of neurons and dropout rate on NSL-KDD Dataset [65]

Sr. No.	Neurons in Hidden Layers	Dropout Rate	Training Accuracy	Testing Accuracy
1.	1024 (HL1) 768 (HL2) 512 (HL3) 256 (HL4) 128 (HL5)	0.01 0.05 0.5	99.88 99.89 99.91	99.86 99.88 99.90
2.	1024 (HL1) 768 (HL2) 512 (HL3) 256 (HL4) 256 (HL5)	0.01 0.05 0.5	99.87 99.88 99.79	99.85 99.85 99.78
3.	1024 (HL1) 768 (HL2) 512 (HL3) 256 (HL4) 512 (HL5)	**0.01** 0.05 0.5	**99.91** 99.91 99.78	**99.90** 99.89 99.77
4.	1024 (HL1) 768 (HL2) 512 (HL3) 256 (HL4) 1024 (HL5)	0.01 0.05 0.5	99.88 99.89 99.86	99.86 99.88 99.86

c) **Convolutional Neural Network Model:** Convolutional Neural Network are generally used for image classification task. We have used CNN model with one convolutional layer , one max pooling layer , one dense layer with 128 neurons and finally one output layer with 5 neurons representing then number of output classes. We have used Adam as optimizer and categorical cross entropy as a loss function. Table 5.5 shows the performance analysis of CNN network w.r.t. different dropout rate and neurons.

Table 5.5 The effect of choosing different number of neurons and dropout rate on CNN model's performance

Sr. No.	Layers	Dropout Rate		Training Accuracy	Testing Accuracy
1.	Convolution Layer Max Pooling Dense (128) Output Layer	0.01 0.05 0.5		99.90 99.92 99.90	99.88 99.90 99.89
2.	Convolution Layer Max Pooling Dense (256) Output Layer	0.01 0.05 0.5		99.87 99.89 99.90	99.85 99.87 99.89
3.	Convolution Layer Max Pooling Dense (512) Output Layer	**0.01** **0.05** 0.5		**99.94** **99.93** 99.90	**99.91** **99.91** 99.88
4.	Convolution Layer Max Pooling Dense (1024) Output Layer	0.01 **0.05** 0.5		99.91 **99.93** 99.89	99.89 **99.91** 99.87

d) **Attention Model:** The model architecture consists of an input layer that takes in data with a shape of (30,), two hidden layers with 64 and 32 neurons respectively, and a dropout layer with a dropout rate of 0.5 to prevent

overfitting. The output layer consists of 5 neurons with a softmax activation function, which is used for multi-class classification problems. An attention mechanism is added to the model, which is implemented by computing attention probabilities from the output of the second hidden layer using a dense layer with a softmax activation function. The attention probabilities are then multiplied element-wise with the output of the second hidden layer using the multiply layer from Keras. The model is compiled using the Adam optimizer, sparse categorical cross-entropy as the loss function, and accuracy as the evaluation metric. The model is then trained for 10 epochs using a batch size of 32 and validated on a test set. After training, the model is evaluated on both the training and test sets to get the accuracy and loss scores. Finally, the code plots the training and validation accuracy and loss curves using matplotlib to visualize the training process (Table 5.6).

Table 5.6 The performance of Attention model by selection of different number of neurons and dropout rate on NSL-KDD dataset [65]

Sr. No.	Layers (Neurons)	Dropout Rate	Training Accuracy	Testing Accuracy
1.	Hidden Layer 1 (64)	0.01	99.93	99.91
	Hidden Layer 2 (32)	0.1	99.93	99.90
	Attention Layer (32)	0.5	99.91	99.89
2.	Hidden Layer 1 (64)	0.01	99.93	99.91
	Hidden Layer 2 (64)	0.1	99.89	99.86
	Attention Layer (64)	0.5	99.91	99.90
3.	Hidden Layer 1 (64)	0.01	99.93	99.91
	Hidden Layer 2 (128)	**0.1**	**99.93**	**99.92**
	Attention Layer (128)	**0.5**	**99.93**	**99.92**
4.	Hidden Layer 1 (64)	0.01	99.93	99.91
	Hidden Layer 2 (256)	**0.1**	**99.94**	**99.92**
	Attention Layer (256)	0.5	99.93	99.91

5.2.2 Kyoto 2006 Dataset:

The Kyoto dataset is another benchmark dataset for intrusion detection that is designed to evaluate anomaly detection techniques. It was collected from a campus network in Kyoto, Japan, and consists of network traffic data captured over a period of three years i.e. from Nov. 2006 to Aug. 2009 using honeypot, darknet, web crawler and email server. A newer version was also developed which contain data collected from Nov 2006 to Dec. 2015. It contain the 14 conventional features selected from the KDD 99 cup dataset (Table-5.7) and addition of 10 features was also done (Table 5.8). The dataset contains 23 features that describe network traffic, such as source and destination IP addresses, protocol type, and service. The dataset is labeled with four classes of network traffic: normal, unknown attack, probe, and denial-of-service (DoS) attack.

Table 5.7: The conventional 14 features that were selected from KDD cup 99 dataset [64]

Sr. No.	Feature	Definition
1	Duration	The length (number of seconds) of the connection
2	Service	The connection's service type, e.g., http, telnet, etc.
3	Source bytes	the number of data bytes sent by the source IP address
4	Destination bytes:	the number of data bytes sent by the destination IP Address
5.	Count	the number of connections whose source IP address and destination IP address are the same to those of the current connection in the in the past two seconds

6.	Same srv rate:	% of connections to the same service in Count feature
7.	Serror rate	% of connections that have “SYN” errors in Count feature
8.	Srv serror rate	% of connections that have “SYN” errors in Srv count (the number of connections whose service type is the same to that of the current connection in the past two seconds) feature
9.	Dst host count	among the past 100 connections whose destination IP address is the same to that of the current connection, the number of connections whose source IP address is also the same to that of the current connection
10.	Dst host srv count:	among the past 100 connections whose destination IP address is the same to that of the current connection, the number of connections whose service type is also the same to that of the current connection
11.	Dst host same src port rate	% connections whose source port is the same to that of the current connection in Dst host count feature
12.	Dst host serror rate	% of connections that have “SYN” errors in Dst host count feature
13.	Dst host srv serror rate	% of connections that “SYN” errors in Dst host srv count feature
14.	Flag	the state of the connection at the time the summary was written which is usually when the connection terminated).

Table 5.8: List of 10 additional features that were added to dataset

Sr. No.	Feature	Definition
1	IDS detection	Reflects whether IDS (Intrusion Detection System) triggered an alert for the connection; '0' means any alerts were not triggered, and an Arabic numeral(except '0') means the different kinds of the alerts. Parenthesis indicates the number of the same alert observed during the connection.
2	Malware detection:	Indicates whether malware, also known as malicious software, was observed in the connection; '0' means no malware was observed, and a string indicates the corresponding malware observed at the connection. Used 'clamav' software to detect malwares. Parenthesis indicates the number of the same malware observed during the connection.
3	Ashula detection	Means whether shellcodes and exploit codes were used in the connection by using the dedicated software; '0' means no shell- codes and exploit codes were observed, and an Arabic numeral(except '0') means the different kinds of the shellcodes or exploit codes. Parenthesis indicates the number of the same shellcode or exploit code observed during the connection.
4	Label	Indicates whether the session was attack or not; '1' means the session was normal, '-1' means known attack was observed in the

		session, and '-2' means unknown attack was observed in the session.
5.	Source IP Address	Indicates the source IP address used in the session. Due to the security concerns, the original IP address on IPv4 was properly sanitized to one of the Unique Local IPv6 Unicast Addresses (private IP addresses). Also, the same private IP addresses are only valid in the same month: if two private IP addresses are the same within the same month, it means their IP addresses on IPv4 were also the same, but if two private IP addresses are the same within the different month, their IP addresses on IPv4 are also different.
6.	Source Port Number	Indicates the source port number used in the session.
7.	Destination IP Address	Indicates the source IP address used in the session. Due to the security concerns, the original IP address on IPv4 was properly sanitized to one of the Unique Local IPv6 Unicast Addresses (private IP address). Also, the same private IP addresses are only valid in the same month: if two private IP addresses are the same within the same month, it means their IP addresses on IPv4 were also the same, but if two private IP addresses are the same within the different month, their IP addresses on IPv4 are also different.
8.	Destination Port Number	Indicates the destination port number used in the session.
9.	Start Time	Indicates when the session was started.

10.	Protocol	indicates the protocol used by the connection

The Data pre-processing steps are as follows:

1. The data is collected from the Kyoto Dataset website [66] for Jan 2015. It has 381105 instances with dimension of 17. The we assigned the columns name with the following features 1: 'Duration', 2: 'Service', 3: 'Source bytes', 4 : 'Destination bytes', 5:'Count', 6: 'Same srv rate', 7: 'Serror rate', 8: 'Srv serror rate', 9: 'Dst host count', 10: 'Dst host srv count', 11: 'Dst host same src port rate', 12: 'Dst host serror rate', 13: 'Dst host srv serror rate', 14: 'Flag', 15: 'Label', 16: 'Protocol', 17: 'Start Time'. Then we remove the unknown attacks from label column having label = -2 that reduces the data instances to 381088.
2. Next step involves encoding all the data into numerical values which enhances the dimension to 43 from 17 as each encoding of subclasses is also done (e.g. different subclasses of flag attributes also encoded and expanded in different columns).
3. The number of output classes are selected i.e. 1 for benign (41495 instances) and -1 for malicious (339593 instances). Now the 82990 instances are selected from the total data with equal distribution of benign and malicious classes (i.e. all 41495 instances of benign and 41495 instances from 339593 instances). Now from these 82290 instances are divided into ratio of 80 to 20 for training set (66392 points) and testing set (16398 points).
4. Next steps involves the selection of deep learning models (shallow neural network, deep neural network, convolutional neural network and attention based model).

We have used following models for prediction of different attack scenarios

a) **Shallow Neural Network Model**: The same model was chosen as for NSL—KDD [65] dataset but now the output classes are only 2 i.e. Malicious and benign attack. Therefore last output layer has only two neurons. The results are depicted in the Table 5.9. The red marked points in Table 5.9 indicates the cases of overfitting.

Table 5.9 The effect of choosing different number of neurons and dropout rate for Kyoto Dataset [66] on Shallow Neural Network Model's performance

Sr. No.	Neurons in Hidden Layer	Drop Out Rate	Training Accuracy	Testing Accuracy
1.	128	0.01	98.23	98.08
		0.05	97.84	97.63
		0.5	97.54	97.57
2.	256	0.01	97.98	97.89
		0.05	98.14	98.16
		0.5	97.46	97.42
3.	512	0.01	97.96	97.83
		0.05	98.26	98.17
		0.5	97.90	97.79
4.	1024	0.01	**98.51**	**98.48**
		0.05	98.26	98.20
		0.5	97.77	97.63

b) **Deep neural network Model:** The experimentation was done with different dropout rate between each hidden layer as shown in Table 5.10.

Table 5.10 The performance of DNN Model accuracy by selection of different number of neurons and dropout rate on Kyoto Dataset [66]

Sr. No.	Neurons in Hidden Layers	Dropout Rate	Training Accuracy	Testing Accuracy
1.	1024 (HL1) 768 (HL2) 512 (HL3) 256 (HL4) 128 (HL5)	**0.01** 0.05 0.5	**98.33** 92.07 90.91	**98.35** 91.78 90.71

2.	1024 (HL1) 768 (HL2) 512 (HL3) 256 (HL4) 256 (HL5)	0.01 0.05 0.5	98.14 98.09 91.81	98.10 98.15 91.48
3.	1024 (HL1) 768 (HL2) 512 (HL3) 256 (HL4) 512 (HL5)	0.01 0.05 0.5	86.68 98.37 96.98	86.67 98.12 96.83
4.	1024 (HL1) 768 (HL2) 512 (HL3) 256 (HL4) 1024 (HL5)	0.01 0.05 0.5	98.26 97.78 89.65	98.14 97.60 89.46

c) **Convolutional Neural Network Model:** Table 5.11 shows the performance analysis of CNN network w.r.t. different dropout rate and neurons.

Table 5.11 The performance of CNN Model accuracy by selection of different number of neurons and dropout rate on Kyoto Dataset [66]

Sr. No.	Layers	Dropout Rate	Training Accuracy	Testing Accuracy
1.	Convolution Layer Max Pooling Dense (128) Output Layer	0.01 0.05 0.5	97.82 91.21 91.84	97.72 91.00 91.54
2.	Convolution Layer Max Pooling Dense (256) Output Layer	0.01 0.05 0.5	98.41 98.05 91.92	98.33 98.10 91.56

3.	Convolution Layer Max Pooling Dense (512) Output Layer	0.01 0.05 0.5	98.28 98.22 91.60	98.15 98.21 91.19
4.	Convolution Layer Max Pooling Dense (1024) Output Layer	0.01 **0.05** 0.5	98.25 **98.38** 98.14	98.24 **98.34** 98.08

d) **Attention Model:** Table 5.12 shows the performance analysis of attention model.

Table 5.12 The performance of Attention model by selection of different number of neurons and dropout rate on Kyoto dataset [66]

Sr. No.	Layers (Neurons)	Dropout Rate	Training Accuracy	Testing Accuracy
1.	Hidden Layer 1 (64) Hidden Layer 2 (32) Attention Layer (32)	0.01 0.1 0.5	98.15 92.10 97.27	98.05 91.91 96.96
2.	Hidden Layer 1 (64) Hidden Layer 2 (64) Attention Layer (64)	0.01 0.1 0.5	98.20 92.26 94.17	98.01 91.99 94.20
3.	Hidden Layer 1 (64) Hidden Layer 2 (128) Attention Layer (128)	0.01 **0.1** 0.5	98.39 **98.47** 91.60	98.17 **98.20** 91.37
4.	Hidden Layer 1 (64) Hidden Layer 2 (256) Attention Layer (256)	0.01 0.1 0.5	92.54 98.12 98.17	92.25 97.96 97.97

5.2.3 UNSW-NB15 Dataset:

The UNSW-NB15 dataset [67] is a network intrusion detection dataset that was collected from a real-world network environment using IXIA PerfectStorm tool. It

was created by researchers at the University of New South Wales (UNSW) in Australia to address the limitations of existing intrusion detection datasets. The dataset contains 49 features using Argus, Bro-IDS tool that describe network traffic, such as source and destination IP addresses, protocol type, and service. It is labeled with 10 classes of network traffic, including normal, and nine types of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. There are around 2 millions of records and these are stored in 4 CSV files. The Ground truth is also given in File UNSW-GST.csv while different evens are described in events csv file. There are around 175,341 instances for training while 82,332 instances are given from testing point of view. Data preprocessing steps are as follows:

1. The dataset is downloaded from the UNSW-NB15 website [66]. The training instances are 82,332 while test set has 175,341 instances. Since this uneven distribution for test and train set, we have combined all the instances making it total 257673 instances using pandas library‘s concatenation function. We have 45 features in the form of columns (['id', 'dur', 'proto', 'service', 'state', 'spkts', 'dpkts', 'sbytes', 'dbytes', 'rate', 'sttl', 'dttl', 'sload', 'dload', 'sloss', 'dloss', 'sinpkt', 'dinpkt', 'sjit', 'djit', 'swin', 'stcpb', 'dtcpb', 'dwin', 'tcprtt', 'synack', 'ackdat', 'smean', 'dmean', 'trans_depth', 'response_body_len', 'ct_srv_src', 'ct_state_ttl', 'ct_dst_ltm', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_dst_src_ltm', 'is_ftp_login', 'ct_ftp_cmd', 'ct_flw_http_mthd', 'ct_src_ltm', 'ct_srv_dst', 'is_sm_ips_ports', 'attack_cat', 'label']).
2. There are 10 types of attack in attack category columns. The attack type and their instances are ‘Normal’: 93000, ‘Reconnaissance’: 13987, 'Backdoor': 2329, 'DoS': 16353, 'Exploits': 44525 'Analysis': 2677, 'Fuzzers': 24246, 'Worms': 174, 'Shellcode': 1511, 'Generic': 58871.
3. Next step involves dropping the categorical columns like id, proto, service, state, attack category. Now, data contains 39 numerical valued columns (except label) and we find the correlation amongst all the 39 features as shown in Fig. 5.7. (Hotmap). It is done to select the relevant feature that are not highly correlated to each other. The columns 'dbytes', 'sloss', 'dloss', 'dwin',

'ct_ftp_cmd', 'ct_srv_dst' are dropped as they have high correlation with other features (> .0.97). Therefore, we have effectively around 33 features left to do further processing.

4. Now data is divided into training and testing set with ratio of 33 to 67 i.e. 172640 training instances and 85033 testing instances with only two output classes i.e. normal and abnormal. Now we deploy different machine learning models and analyze performance of each model with training and testing accuracy.

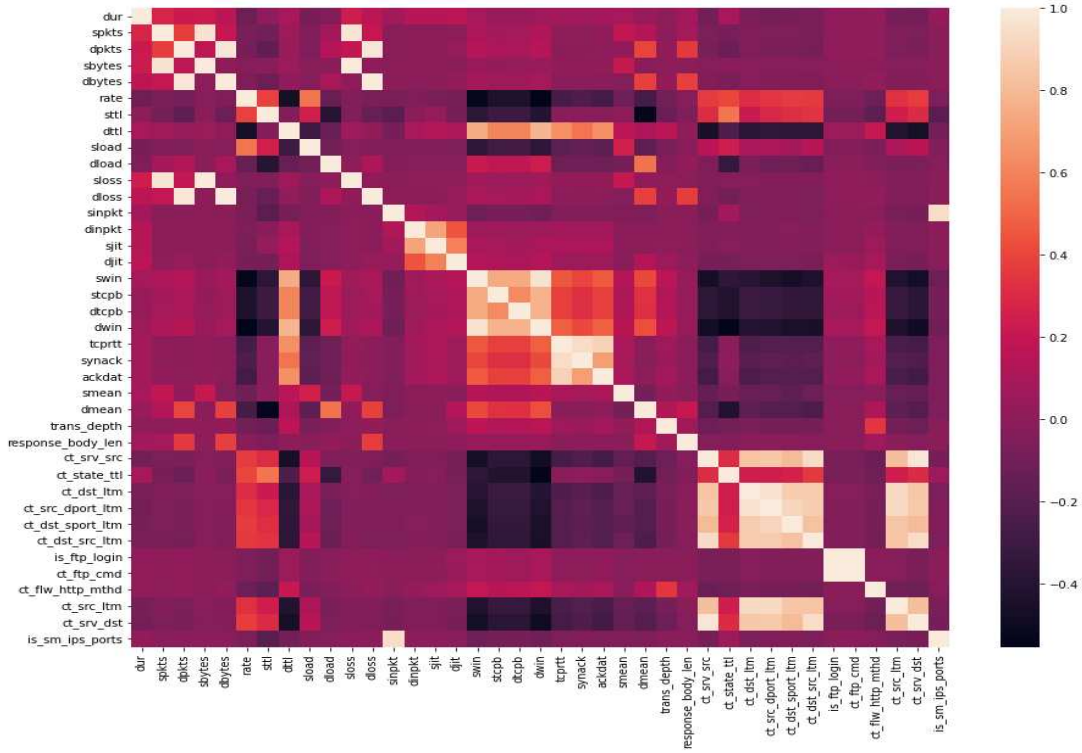

Fig. 5.7. The correlation between the 39 features for UNSW-NB15 dataset [67]

d) **Shallow Neural Network Model**: The model similarly chosen for NSL—KDD [65] dataset is used but now the output classes are only 2 i.e. Malicious and benign attack. The input features are 33. Therefore last output layer has only two neurons. The results are depicted in the Table 5.13. The red marked points in Table 5.13 indicates the cases of overfitting. The system was trained for 10 epochs with batch size of 32.

Table 5.13 The effect of choosing different number of neurons and dropout rate for UNSW-NB15 Dataset [67] on Shallow Neural Network Model's performance

Sr. No.	Neurons in Hidden Layer	Drop Out Rate	Training Accuracy	Testing Accuracy
1.	128	**0.01**	**93.24**	**93.30**
		0.05	**93.27**	**93.32**
		0.5	**92.36**	**92.38**
2.	256	0.01	97.98	97.89
		0.05	**93.39**	**93.39**
		0.5	97.46	97.42
3.	512	0.01	97.96	97.83
		0.05	98.26	98.17
		0.5	97.90	97.79
4.	1024	0.01	**98.51**	**98.48**
		0.05	98.26	98.20
		0.5	97.77	97.63

e) **Deep neural network Model:** The experimentation was done with different dropout rate between each hidden layer as shown in Table 5.14.

Table 5.14 The performance of DNN Model accuracy by selection of different number of neurons and dropout rate on UNSW-NB15 Dataset [67]

Sr. No.	Neurons in Hidden Layers	Dropout Rate	Training Accuracy	Testing Accuracy
1.	1024 (HL1) 768 (HL2) 512 (HL3) 256 (HL4) 128 (HL5)	**0.01** 0.05 0.5	**98.33** 92.07 90.91	**98.35** 91.78 90.71
2.	1024 (HL1)	0.01	98.14	98.10

	768 (HL2) 512 (HL3) 256 (HL4) 256 (HL5)	0.05 0.5	98.09 91.81	98.15 91.48
3.	1024 (HL1) 768 (HL2) 512 (HL3) 256 (HL4) 512 (HL5)	0.01 0.05 0.5	86.68 98.37 96.98	86.67 98.12 96.83
4.	1024 (HL1) 768 (HL2) 512 (HL3) 256 (HL4) 1024 (HL5)	0.01 0.05 0.5	98.26 97.78 89.65	98.14 97.60 89.46

f) **Convolutional Neural Network Model:** Table 5.15 shows the performance analysis of CNN network w.r.t. different dropout rate and neurons.

Table 5.15 The performance of CNN Model accuracy by selection of different number of neurons and dropout rate on UNSW-NB15 Dataset [67]

Sr. No.	Layers	Dropout Rate	Training Accuracy	Testing Accuracy
1.	Convolution Layer Max Pooling Dense (128) Output Layer	0.01 0.05 0.5	97.82 91.21 91.84	97.72 91.00 91.54
2.	Convolution Layer Max Pooling Dense (256) Output Layer	0.01 0.05 0.5	98.41 98.05 91.92	98.33 98.10 91.56

3.	Convolution Layer Max Pooling Dense (512) Output Layer	0.01 0.05 0.5	98.28 98.22 91.60	98.15 98.21 91.19
4.	Convolution Layer Max Pooling Dense (1024) Output Layer	0.01 **0.05** 0.5	98.25 **98.38** 98.14	98.24 **98.34** 98.08

e) **Attention Model:** Table 5.16 shows the performance analysis of attention model.

Table 5.16 The performance of Attention model by selection of different number of neurons and dropout rate on UNSW-NB15 Dataset [67]

Sr. No.	Layers (Neurons)	Dropout Rate	Training Accuracy	Testing Accuracy
1.	Hidden Layer 1 (64) Hidden Layer 2 (32) Attention Layer (32)	0.01 0.1 0.5	98.15 92.10 97.27	98.05 91.91 96.96
2.	Hidden Layer 1 (64) Hidden Layer 2 (64) Attention Layer (64)	0.01 0.1 0.5	98.20 92.26 94.17	98.01 91.99 94.20
3.	Hidden Layer 1 (64) Hidden Layer 2 (128) Attention Layer (128)	0.01 **0.1** 0.5	98.39 **98.47** 91.60	98.17 **98.20** 91.37
4.	Hidden Layer 1 (64) Hidden Layer 2 (256) Attention Layer (256)	0.01 0.1 0.5	92.54 98.12 98.17	92.25 97.96 97.97

5.2.4 Comparison of all models

In the Table 5.17, a comparative analysis is done w.r.t all the models proposed. The best test and training accuracy are selected for the purpose.

Table 5.17: Comparison between all the Proposed Models

DL/ML Model	NSL-KDD Cup Database[65]		Kyoto Database [66]		UNSW-NB15 Database[67]	
	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy
Shallow Neural Network						
Deep Neural Network						
CNN						
Attention Models						

5.3 Conclusion

We have analyzed different deep learning models in this thesis. The simplest one are Shallow Neural Networks that have only one hidden layer, and the output layer predicts the class of the input. They are good for simple classification tasks with linearly separable data. However, for complex data, they may not be able to capture the nuances and variations in the data. Another networks are Deep Neural Networks that have multiple hidden layers, allowing them to learn complex representations of data. They can learn features at different levels of abstraction and generalize better to unseen data. However, they require more training data and computational resources than shallow networks. Next we have analyzed Convolutional Neural Networks that were designed specifically for image and video data. They use convolutional layers to learn features from local neighborhoods of pixels and pooling layers to reduce the dimensionality of the feature maps. They are highly effective for image classification tasks, achieving state-of-the-art performance on many benchmarks. However, they

can also be tuned for different dataset. Finally, we have chosen the more contemporary model like attention models which are a family of neural network architectures that enable the network to focus on specific parts of the input sequence when making predictions. They are commonly used in natural language processing (NLP) tasks such as language translation and summarization. Attention models can capture long-term dependencies between input sequences and improve performance compared to traditional sequence-to-sequence models. However, they requires a lot of data and time to train. In multi-class classification tasks, all four types of neural networks can be used. Shallow neural networks can work for simple classification tasks, but for more complex datasets, deep neural networks, CNNs, and attention models can provide better accuracy. However, the specific choice of neural network architecture will depend on the characteristics of the data, the size of the dataset, and the computational resources available. Another limitation of neural networks is their requirement for large amounts of training data. Training a neural network requires a large dataset that accurately reflects the real-world distribution of the data, and obtaining such data can be challenging and expensive. Furthermore, if the dataset is biased or incomplete, the network may learn incorrect or incomplete representations of the data.

5.4 Future Work:

Recently, deep learning has emerged as a powerful tool for multi-class classification, with many new models being developed and refined over the past few years. The future work entails exploring of the newest and most promising models which are summarized as

1. **Transformer-based Models:** Transformer-based models, such as BERT and GPT-3, have revolutionized natural language processing (NLP) tasks and achieved state-of-the-art results on a range of benchmarks. These models use attention mechanisms to encode and decode sequential data, allowing them to capture long-term dependencies in language data. They can be fine-tuned for multi-class classification tasks, making them highly relevant for future applications in NLP.

2. **Vision Transformers (ViT):** ViT is a recent development in computer vision that replaces convolutional layers with transformer-based layers. Like the original transformer model, ViT uses self-attention to process image patches and generate features. ViT has achieved state-of-the-art results on several image classification benchmarks, showing promising potential for future use in multi-class classification tasks.

3. **Graph Neural Networks (GNNs):** GNNs are a type of neural network that can process graph data, such as social networks, protein structures, or molecular data. GNNs use message-passing mechanisms to propagate information between nodes and edges in the graph, allowing them to learn complex relationships between entities. They have been used for a range of multi-class classification tasks, including drug discovery and social network analysis.
4. **Capsule Networks:** Capsule networks are a recent development in computer vision that aim to address the limitations of traditional convolutional neural networks (CNNs). Capsule networks use capsules, which are groups of neurons that encode properties of objects or features. They can learn to recognize objects regardless of their position or orientation in the image, making them highly relevant for future applications in image classification.

In conclusion, the development of new deep learning models for multi-class classification is an active area of research. The transformer-based models, ViT, GNNs, and capsule networks are just a few examples of the latest developments in this field, and they show great potential for future applications in NLP, computer vision, and graph-based data analysis which can be further pipelined with the problem of attacks. Moreover once model has been trained and evaluated, you can deploy by using using cloud services like Amazon Web Services, Google Cloud Platform or Microsoft Azure, or deploy the model on-premises an we can integrate the deployed model with application or system to enable real-time analysis. This might involve using RESTful APIs or message queues to send data to the model for prediction and

returning the prediction back to the application or system. Further, we can monitor the performance of deployed model in real-time and can track metrics such as accuracy, throughput, and latency to ensure that the model is performing well and make improvements as needed.

References

[1]. K. Jansson & R. von Solms (2013) Phishing for phishing awareness, Behaviour & Information Technology, 32:6, 584-593, DOI: 10.1080/0144929X.2011.632650

[2]. V. Wilson, 24 Recent Ransomware Attacks in 2019, 2019, [online] Available: https://spinbackup.com/blog/24-biggest-ransomware-attacks-in-2019.

[3]. R. R. Brooks, L. Yu, I. Ozcelik, J. Oakley and N. Tusing, "Distributed Denial of Service (DDoS): A History," in IEEE Annals of the History of Computing, vol. 44, no. 2, pp. 44-54, 1 April-June 2022, doi: 10.1109/MAHC.2021.3072582.

[4]. E. Osterweil, A. Stavrou and L. Zhang, "21 Years of Distributed Denial-of Service: Current State of Affairs," in *Computer*, vol. 53, no. 7, pp. 88-92, July 2020, doi: 10.1109/MC.2020.2983711.

[5]. A. Alshamrani, S. Myneni, A. Chowdhary and D. Huang, "A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851-1877, Secondquarter 2019, doi: 10.1109/COMST.2019.2891891.

[6]. Abusitta, Adel, Miles Q. Li, and Benjamin CM Fung. "Malware classification and composition analysis: A survey of recent developments." *Journal of Information Security and Applications* 59 (2021): 102828.

[7]. Nasereddin, M., ALKhamaiseh, A., Qasaimeh, M., & Al-Qassas, R. (2021). A systematic review of detection and prevention techniques of SQL injection attacks. *Information Security Journal: A Global Perspective*, 1-14.

[8]. Halfond, William G., Jeremy Viegas, and Alessandro Orso. "A classification of SQL-injection attacks and countermeasures." *Proceedings of the IEEE international symposium on secure software engineering*. Vol. 1. IEEE, 2006

[9]. M. Uma, G. Padmavathi "A Survey on Various Cyber Attacks and their Classification" Computer Science, Int. J. Netw. Secur. DOI: 10.6633/IJNS.201309.15(5).09, September 2013Alazab, M., Venkatraman,

S., & Alazab, M. (2012). A hybrid clustering technique for DDoS attack detection. Journal of Network and Computer Applications, 35(5), 1674-1681.

[10]. Khan, M. U., Khan, S. U., & Saeed, M. (2016). A hierarchical clustering technique for insider threat detection. Journal of Network and Computer Applications, 66, 55-63.

[11]. Gomathi, G., & Ramalingam, V. (2017). Detection of unknown cyber attacks using fuzzy clustering technique. Computers & Electrical Engineering, 59, 144-157.

[12]. Asghar, A., Abbas, H., Habib, A., Asghar, M. Z., & Amin, F. (2021). A K-means clustering technique for ransomware attack detection. Journal of Ambient Intelligence and Humanized Computing, 12(3), 2483-2493

[13]. Ali, S., & Arshad, N. (2019). Intrusion detection system using support vector machine. Journal of Intelligent & Fuzzy Systems, 36(4), 3265-3272.

[14]. Khraisat, A. M., Aboalsamh, H., Al-Azzam, O., Al-Betar, M. A., & Yaseen, S. M. (2021). SVM-based cyber attack detection in the internet of things. IEEE Internet of Things Journal, 8(3), 1423-1433.

[15]. Sharma, A., & Mishra, V. K. (2019). Detection of denial of service attack using support vector machine with four different kernel functions. Procedia Computer Science, 155, 81-89.

[16]. Sun, Y., Yu, L., Luo, T., & Liu, H. (2020). Deep SVM: a hybrid deep learning model for web attack detection. IEEE Transactions on Industrial Informatics, 16(2), 965-974.

[17]. Liu, Y., Lin, X., & Shen, Z. (2021). Anomaly detection of network traffic based on improved random forest algorithm. IEEE Access, 9, 15897-15908.

[18]. Zaman, M. A., Rahman, M. M., & Kaiser, M. S. (2019). Intrusion detection in wireless sensor networks using random forest algorithm. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[19]. Zhang, J., Li, J., Wang, G., & Huang, Y. (2019). DDoS attack detection based on random forest algorithm in cloud environment. Security and Communication Networks, 2019

[20]. Sun, L., Yang, Y., Guo, Y., & Zhang, Y. (2020). A novel malware detection model based on convolutional neural network and random forest. IEEE Access, 8, 96525-96534.

[21]. Abadeh, S. S., Rahmani, A. M., Jolfaei, A., & Avestimehr, S. S. (2019). A hybrid approach based on random forest and fuzzy clustering for detecting malicious traffic in IoT networks. Computers & Security, 82, 125-138.

[22]. Ahmadi, A., & Khosravi, A. (2021). A hybrid model based on k-nearest neighbor and naive bayes algorithms for network attack detection. Journal of Ambient Intelligence and Humanized Computing, 12(4), 4375-4384.

[23]. Liu, L., Zou, Q., & Liu, Y. (2020). A deep learning approach based on naive bayes for DDoS attack detection. Wireless Networks, 26(8), 5469-5480.

[24]. Kumar, V., Verma, A. K., & Goyal, R. (2019). Multi-class classification of cyber attacks using support vector machine, random forest and naive bayes. Journal of Cyber Security Technology, 3(3), 180-201.

[25]. Alizadeh, H., & Rahmani, A. M. (2019). A hybrid model for detecting cyber attacks using naive bayes and artificial bee colony algorithm. Journal of Ambient Intelligence and Humanized Computing, 10(8), 3295-3306.

[26]. Mubarak, F., & Alotaibi, F. (2018). Hybrid detection model of cyber-attacks using decision tree and naive bayes algorithms. Future Computing and Informatics Journal, 3(2), 219-224.

[27]. Chen, Y., Zhang, J., Wang, X., & Wang, W. (2021). A hybrid model combining CNN, LSTM, and RNN for DDoS attacks detection. IEEE Access, 9, 122890-122903.

[28]. Liu, Y., Li, H., Li, L., & Shi, W. (2021). An RNN-based intrusion detection system for wireless sensor networks. Computer Communications, 175, 66-77.

[29]. Li, P., Cao, Y., & Zhang, Q. (2020). An RNN-based anomaly detection model for industrial control systems. Future Generation Computer Systems, 110, 86-97.

[30]. Wang, X., Zhang, J., Liu, J., Chen, Y., & Yang, J. (2020). An RNN-based approach for detecting malware attacks in IoT networks. Journal of Network and Computer Applications, 168, 102725.

[31]. Chen, J., Liu, C., Xu, Y., & Xu, L. (2019). A hybrid deep learning model for intrusion detection system. IEEE Access, 7, 95991-96003

[32]. Moustafa, Nour, and Amr Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the NSL-KDD dataset." Journal of Information Security and Applications 41 (2018): 99-111.

[33]. Ali, Muhammad, et al. "Deep learning-based malware detection using binary texture analysis of opcode sequences." IEEE Access 7 (2019): 164269-164280.

[34]. Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011, July). Malware images: visualization and automatic classification. In Proceedings of the 8th international symposium on visualization for cyber security (pp. 1-7).

[35]. Zhang, Wei, et al. "A convolutional neural network based intrusion detection system for industrial control systems." IEEE Access 7 (2019): 172207-172217.

[36]. Li, Jiaming, et al. "A CNN-based approach for detecting DDoS attacks in cloud computing environments." Journal of Parallel and Distributed Computing 140 (2020): 126-135.

[37]. Li, Q., Jia, X., Chen, J., & Wu, Y. (2020). A stacking-based ensemble learning framework for cyber attack detection. Journal of Ambient Intelligence and Humanized Computing, 11(10), 4467-4479.

[38]. Zhang, C., Li, Y., Xie, H., & Huang, C. (2021). An ensemble learning approach for DDoS attack detection. Journal of Ambient Intelligence and Humanized Computing, 12(6), 6087-6102.

[39]. Bhat, M. A., Wang, C., & Chen, Z. (2021). An ensemble learning based approach for detecting botnet attacks. IEEE Access, 9, 46492-46505.

[40]. Kim, S. H., Lim, K. H., & Kim, K. (2021). Ensemble of deep learning models for malware detection. Computers & Security, 108, 102198.

[41]. Chen, H., Chen, Y., & Huang, Y. (2018). User behavior-based advanced persistent threat detection method using restricted Boltzmann machines. Future Generation Computer Systems, 87, 13-24.

[42]. Liu, J., Xu, L., & Chen, H. (2018). Insider threat detection using deep belief networks with restricted Boltzmann machines. IEEE Transactions on Computational Social Systems, 5(1), 197-207.

[43]. Hu, W., Hu, W., & Maybank, S. (2017). Convolutional restricted Boltzmann machine for network anomaly detection. Neural Networks, 95, 132-141.

[44]. Kwon, O. J., & Lee, J. H. (2017). Malware classification using deep belief networks with two-dimensional restricted Boltzmann machines. Expert Systems with Applications, 79, 19-27.

[45]. Dissecting Android Malware: Characterization and Evolution Yajin Zhou, Xuxian Jiang Proceedings of the 33rd IEEE Symposium on Security and Privacy (Oakland 2012) San Francisco, CA, May 2012

[46]. Kim, M. J., Kim, J. H., & Jung, S. W. (2021). Malware detection using attention-based LSTM. IEEE Access, 9, 18074-18082.

[47]. Choi, S., Song, J., & Kim, T. H. (2018). Attention-based recurrent neural network for detecting insider threats. IEEE Access, 6, 67057-67064.

[48]. Wang, K., Liu, Y., & Zhang, S. (2019). Attention-based CNN for network anomaly detection. IEEE Access, 7, 163054-163064.

[49]. Huang, X., Sun, Y., & Huang, X. (2021). An attention-based deep learning framework for DDoS detection. IEEE Access, 9, 18983-18992.

[50]. Wang, X., Yang, Z., & Zhang, H. (2020). Attention-based deep learning method for phishing email detection. IEEE Access, 8, 196959-196968.

[51]. Ding, Y., Liu, B., & He, X. (2021). One-shot learning-based malware detection using convolutional neural network and Siamese network. Journal of Ambient Intelligence and Humanized Computing, 12(6), 5837-5845.

[52]. Chen, M., Li, S., & Liu, Y. (2019). One-shot learning-based advanced persistent threat detection using network traffic. IEEE Access, 7, 127159-127167.

[53]. Park, J., Song, Y., & Han, D. K. (2020). One-shot learning for phishing detection using deep neural network. IEEE Access, 8, 206268-206278.

[54]. Wang, B., Guo, C., & Zhao, X. (2021). One-shot learning-based adversarial example detection. Journal of Intelligent & Fuzzy Systems, 40(2), 3211-3219.

[55]. Tang, Y., Chen, X., & Wang, S. (2021). Anomaly detection for network traffic using generative adversarial networks. Journal of Ambient Intelligence and Humanized Computing, 12(9), 10017-10027.

[56]. Li, Y., Zhang, W., & Chen, S. (2019). A novel malware detection approach based on generative adversarial network. Journal of Intelligent & Fuzzy Systems, 37(5), 6065-6072.

[57]. Huang, Y., Xu, L., & Liu, X. (2021). A generative adversarial network-based method for website defacement detection. IEEE Access, 9, 30271-30279.

[58]. Cui, X., Li, B., & Zhang, Y. (2021). A generative adversarial network-based approach for SQL injection detection. Journal of Ambient Intelligence and Humanized Computing, 12(7), 7751-7760.

[59]. Hossain, M. A., Rahman, M. A., & Hasan, M. M. (2021). Phishing website detection using transformer-based deep learning approach. Journal of Ambient Intelligence and Humanized Computing, 12(9), 10423-10436.

[60]. Gupta, A., Sharma, S., & Kumar, V. (2021). Malicious email detection using transformer-based deep learning approach. Journal of Ambient Intelligence and Humanized Computing, 12(9), 10411-10422.

[61]. Rana, S. S., Hassan, S., & Park, D. S. (2021). A transformer-based deep learning approach for intrusion detection system. Journal of Ambient Intelligence and Humanized Computing, 12(9), 10377-10388.

[62]. Huang, Y., Huang, K., & Liu, X. (2021). Insider threat detection based on transformer model with attention mechanism. IEEE Access, 9, 93178-93188.

[63]. DARPA intrusion detection evaluation, http://www.ll.mit.edu/IST/ideval/data/data index.html

[64]. KDD Cup 1999, October 2007, [online] Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[65]. M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set,” Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009

[66]. Song, J., Takakura, H., Okabe, Y., Eto, M., Inoue, D. & Nakao, K. 2011. Statistical Analysis of Honeypot Data and Building of Kyoto 2006+ Dataset for NIDS Evaluation. In: Proc. 1st Work-shop on Building Anal. Datasets and Gathering Experience Returns for Security. Salzburg, pp.29-36. April 10-13. Available at: https://doi.org/10.1145/1978672.1978676.

[67]. Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." *Military Communications and Information Systems Conference (MilCIS)*, 2015. IEEE, 2015.

[68]. Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", 4th International Conference on Information Systems Security and Privacy (ICISSP), Purtogal, January 2018

[69]. Hapke, Hannes, and Catherine Nelson. Building machine learning pipelines. O'Reilly Media, 2020.

[70]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[71]. Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.