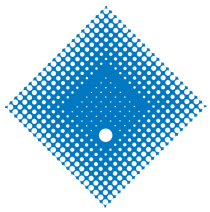

VICTORIA UNIVERSITY

MELBOURNE AUSTRALIA

Network Intrusion Detection using Deep Learning

Lakshit Sama

Master of Research

A thesis submitted for the degree of Masters of Research at Victoria University in July, 2022

Institute of Sustainable Industries & Liveable Cities

Abstract

As network size expands rapidly, network intrusions become more frequent, dynamic, and sophisticated. The topic of how to detect intrusions in such a vast network is crucial and challenging. Due to its intelligent potential, machine learning-based network intrusion detection has recently gained increased attention. Compared to rule-based solutions, machine learning-based solutions, especially those using deep learning, are better capable of identifying network attack variations. In contrast to other application fields, such as image recognition and natural language processing, however, deep learning for network intrusion detection is still in its infancy. It remains to be determined whether it is successful for actual application and if yes, several difficulties need to be studied.

This thesis focuses primarily on two challenges associated with deep learning for network intrusion detection. 1) Excessive human intervention in existing machine learning models, high false-positive rate and low accuracy in existing deep learning solutions; 2) Lack of adequate training data in the network intrusion detection sector; We propose a deep learning system (LightGBM, XGBoost, LSTM, and decision tree) to cope with inadequate data and verify it using three datasets. The standard datasets include the NSL-KDD dataset, the UNSW-NB15 dataset, and the CIC-IDS2017 dataset. Each model is selected based on qualities that are likely to increase the framework's detection.

The findings of the suggested framework indicate that all four consistently outperform the state-of-the-art machine learning-based solutions, demonstrating the efficacy of our thesis-developed design techniques.

Declaration by author

"I, Lakshit Sama, declare that the Master of Research thesis entitled Network Intrusion Detection using Deep Learning is no more than 50,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work".

"I have conducted my research in alignment with the Australian Code for the Responsible Conduct of Research and Victoria University's Higher Degree by Research Policy and Procedures".

Signature 

Date 11/08/2022

Publications included in this thesis

Some of the concepts, data, results, and figures presented in this thesis have been published or are under review for publication in journals or conference proceedings during my Master of Research candidature. I, Lakshit Sama, certify that I was the primary contributor and author of each of these works. The publisher has granted the author permission to reproduce the contents of these publications for academic purposes. A complete list of the aforementioned publications is provided below.

1. Sama, Lakshit, Hua Wang, and Paul Watters, "Enhancing system security by Intrusion Detection using Deep learning". Australian Database Conference 2022. pages: 169-176, doi: 10.1007/978-3-031-15512-3_14.

2. Sama, Lakshit, Hua Wang, and Aaisha Makkar. "Movie Recommendation System Using Deep Learning." 2021 9th International Conference on Orange Technology (ICOT), 2021, pages: 1-4, doi: 10.1109/ICOT54518.2021.9680609.

Acknowledgments

Many people have provided me with ongoing support and assistance throughout this dissertation and my candidacy. I'd like to thank everyone who helped make this paper possible. First and foremost, I want to express my heartfelt gratitude to Prof. Hua Wang, my principal supervisor, mentor, and friend, who provided me with continuous assistance, guidance, support, and encouragement throughout my candidature. The study would not have begun or been completed successfully without his advice and assistance.

Prof. Yanchun Zhang, my co-supervisor, deserves my heartfelt gratitude. He gave me valuable advice and time, which helped me improve the quality of my research. I'd like to thank Dr. Aaisha Makkar of the University of Derby. She shared her vast research experience with me and provided me with encouragement, direction, and invaluable advice to help me with my work. I'd like to thank my friends and colleagues in the VU researchers group. Thank you for your company and numerous discussions throughout my research candidature. I'd also like to thank my family for their help during this difficult time. Thank you to my partner for her encouragement and tolerance for my erratic lifestyle while I worked on my thesis.

Contents

Abstract ii

Contents **vi**

List of Figures **ix**

1 Introduction **1**

1.1 Network Security 1

1.2 Network Protocol and Intrusion 2

1.2.1 Network Intrusion Techniques 3

1.2.2 Impact of Network Intrusion 4

1.3 Case Study - Facebook Data Breach 5

1.4 Ransomware: A strategy of geopolitical destabilisation - The Case of Costa Rica . . 6

1.4.1 Introduction 6

1.4.2 Ransomware in attention and its adversaries? 7

1.4.3 Affect on the Nation 9

1.5 Preventive measures 10

1.6 Contributions 10

1.7 Thesis Organisation 11

2 Background **13**

2.1 Network Intrusion 13

2.1.1 Computer Virus 13

2.1.2 Malware 14

2.1.3 Computer Worm 14

2.1.4 Phishing 15

2.1.5 Botnet 15

2.1.6 DoS (Denial of Service) and DDoS Attacks 16

2.1.7 Man-in-the-middle 16

2.1.8 Ransomware 17

2.1.9 5G Based Attacks 18

2.1.10 SQL Injection Attacks 18

2.2 Recent Network Attacks . . . 19
2.3 Social Engineering . . . 19
2.4 Advanced Persistent Threats . . . 20
2.5 Protection from Network Attacks . . . 20
2.6 Datasets . . . 21
2.6.1 KDD Cup 1999 dataset . . . 21
2.6.2 NSL-KDD dataset . . . 23
2.6.3 DEFCON dataset . . . 25
2.6.4 CAIDA dataset . . . 26
2.6.5 CIC DoS dataset . . . 27
2.6.6 DARPA 1998 dataset . . . 28
2.6.7 CSE-CIC-IDS2018 dataset . . . 29
2.6.8 KYOTO dataset . . . 30
2.6.9 ISCX dataset . . . 31
2.6.10 UNSW-NB15 dataset . . . 33
2.6.11 TWENTE dataset . . . 34
2.6.12 CDX dataset . . . 35
2.6.13 CDX 2009 Dataset Filtering . . . 35
2.6.14 ADFA2013 dataset . . . 35

3 Role of Deep Learning in Network Intrusion 37
3.1 Deep Neural Networks . . . 40
3.2 Recurrent Neural Networks . . . 41
3.3 Convolutional Neural Networks . . . 42
3.4 Deep Auto-encoder . . . 43
3.5 Restricted Boltzmann Machine . . . 44
3.6 Deep Belief Network . . . 45
3.7 Miscellaneous . . . 46
3.7.1 Feedforward Neural Network . . . 46
3.7.2 Deep migration learning . . . 46
3.7.3 Replicator Neural Network . . . 47
3.7.4 Self-Taught Learning . . . 47
3.8 Other Applications of Deep Learning . . . 47
3.8.1 Movie Recommendation System Using Deep Learning . . . 47
3.8.2 Image and Video Recognition . . . 54
3.8.3 Fake videos and images . . . 56

4 Proposed work 59
4.1 Introduction . . . 59
4.2 Contributions . . . 61

4.3 Related Work 61
4.4 Proposed Scheme 64
4.5 Experimental Setup 66
4.6 Results and Discussion 66
4.6.1 Dataset-1: NSL-KDD dataset 66
4.6.2 Dataset-2: UNSW-NB15 Dataset 66
4.6.3 Dataset-3: CIC-IDS2017 66
4.7 Impact of deep learning on the proposed scheme 67
4.8 Impact of different datasets on the proposed scheme 69
4.9 Summary 70

5 Conclusion **71**
5.1 Summary of the thesis 71
5.2 Future work 72

Bibliography **75**

List of Figures

1.1 Network Architecture 2
1.2 Statistics led to Attacks 3
1.3 IDS and IPS process 4
1.4 Process of Network Intrusion 5
1.5 Data Breach statistics in Industries 6

2.1 Different types of attacks and working of IDS 14
2.2 Concept of Bots 16
2.3 Man-in-the-middle attack 17
2.4 Ransomware attack 17
2.5 SQL attack technique 18
2.6 Phases of APT 19
2.7 Public datasets available 21
2.8 Comparison of different Datasets 22

3.1 Machine Learning techniques for Intrusion Detection 38
3.2 Extraction of best movies 52
3.3 Extraction of similar movies 52
3.4 Extraction of other movies of same content 53
3.5 Prediction of movies by the movie recommendation system 53
3.6 Extraction of best movies to watch 54

4.1 Conceptual view 61
4.2 Summary of existing techniques 62
4.3 Steps followed in the proposed work 64
4.4 Samples of dataset-1 66
4.5 Training and Testing data of dataset-2 67
4.6 Setting of hyper parameters for each deep learning model 67
4.7 Attack matrix by different models on NSL-KDD dataset 68
4.8 Comparison of proposed scheme 68
4.9 Accuracy of four models with Dataset-1 69

4.10 Accuracy of four models with Dataset-2 69
4.11 Accuracy of four models with Dataset-3 69

Chapter 1

Introduction

1.1 Network Security

In today's world, businesses rely heavily on computer networks to efficiently and effectively transmit information throughout the corporation [1, 2]. Corporate computer networks are getting increasingly massive and widespread. A large-scale organisation has thousands of workstations and many servers on the network. Assuming that each employee has a dedicated workstation, these workstations are unlikely to be centrally managed or protected from the outside world. They may use a variety of operating systems, hardware, software, and protocols, and their users may have varying levels of cyber awareness. Imagine if thousands of workstations in your company's network are connected to the Internet directly. This type of unsecured network becomes a target for an attack since it contains important information and has flaws [3–5].

Networking Infrastructure: A network is defined as two or more computing devices that are linked together to share resources efficiently. Internet working is the process of linking two or more networks together. As a result, the Internet is nothing more than a collection of interconnected networks [6]. An organisation has several possibilities for setting up its internal network. To connect all workstations, it can employ a wired or wireless network. In today's world, most businesses use a combination of wired and wireless networks. Both wired and wireless devices in a wired network are connected to one another via wires. Wired networks are typically based on the Ethernet protocol, with devices connected to switches using Unshielded Twisted Pair (UTP) connections. These switches are also connected to the network router, which allows them to access the Internet. The device is connected to an access point via radio broadcasts in a wireless network [7, 8]. The access points are also connected to the switch/router via wires for external network access.

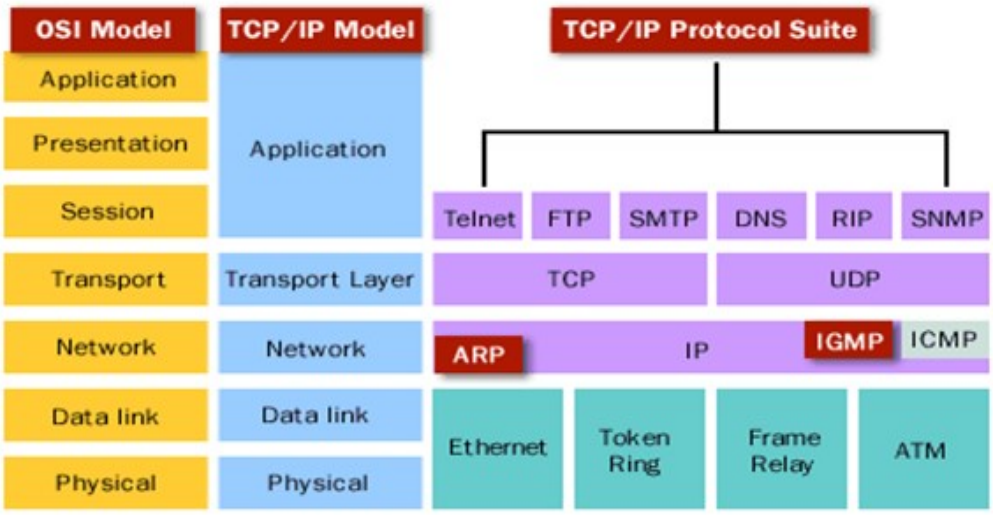

Figure 1.1: Network Architecture

1.2 Network Protocol and Intrusion

A set of rules that control communications between devices linked to a network is known as a network protocol. These include connection-making methods as well as data-packaging formatting guidelines for messages transmitted and received. Several computer network protocols have been created, each with its own set of requirements. TCP/IP, as shown in Figure 1.1, and its associated higher and lower-level protocols are the most popular and commonly used protocols. Transmission Control Protocol (TCP) and Internet Protocol (IP) are two separate computer network protocols that are commonly used together [9]. Because of their widespread use and popularity, they are included in all networked device operating systems. In the OSI model (Figure 1.1), IP corresponds to the Network layer (Layer 3) and TCP to the Transport layer (Layer 4). TCP/IP is a network communication protocol that uses TCP transport to send data across IP networks. At the application layer, TCP/IP protocols are often used alongside additional protocols such as HTTP, FTP, and SSH, and at the data link/physical layer, Ethernet. In 1980, the TCP/IP protocol suite was established as an inter networking solution with minimal regard for security. It was created for communication in a trusted network with a small number of users. This protocol, however, became the de-facto standard for unprotected Internet communication over time [10].

Any unauthorised penetration into a computer network is referred to as a network intrusion [11–13]. Over the last two decades, people's reliance on technology has skyrocketed, resulting in a new wave of computer-related crimes, as discussed in Figure 1.2. In most cases, a network is breached for one of three reasons:

1. **Hacktivism** — Hacktivism is a term that combines the words hacking and activism. Intruders who wish to hack so to prove a social cause or a political agenda are the ones who do it [14–16].

2. **Stolen Money or Data** - This intrusion is carried out in order to steal cash or information from a 3rd party. Usually, the goal is to take advantage of the other party for financial profit [17].

3. **Spy** - It is the use of state-sponsored network infiltration to spy on opponents and sometimes allies [18].

Individuals, corporate organisations, and governments can all be targets of network intrusion assaults [19, 20]. To avoid Network Intrusion, these organisations' cybersecurity teams must first grasp how it is carried out. To deal with the problems caused by network intrusions, a Network Intrusion Detection System must be installed.

Intrusion Detection Technologies and Intrusion Prevention Systems are two types of systems that can help prevent network attacks. Intrusion Detection Systems (IDS) are passive systems that identify harmful behaviour on a network, whereas Intrusion Prevention Systems (IPS) not only detect but also actively prevent unwanted conduct [21] as shown in Figure 1.3.

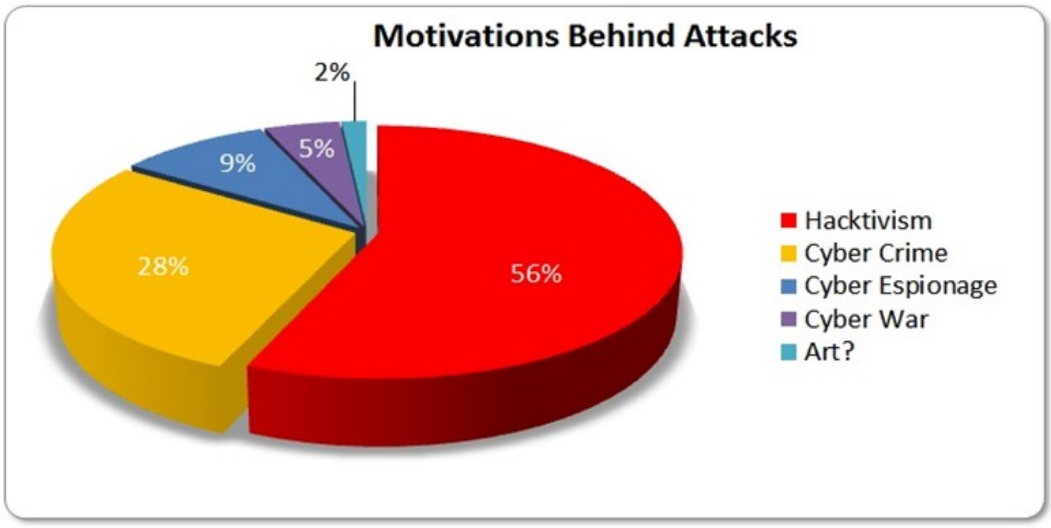

Figure 1.2: Statistics led to Attacks

1.2.1 Network Intrusion Techniques

Due to the vastness of the Internet, pinpointing a specific method of Network Intrusion is extremely challenging [22, 23]. However, following are some of the most typical ways that Network Intrusion (Figure 1.4), has occurred [24]:

1. Multi-Routing- When intruders invade using several sources to prevent detection, it is referred to as multi-routing. Asymmetric routing is another name for multi-routing.

2. Buffer Overflow Attacks- This type of attack occurs when portions of the computer's memory code are altered so that they can later be exploited in the intrusion.

3. Traffic Flooding- Intruders flood the victim's machine with the traffic they can't handle in order to cause chaos and confusion. When the systems have too much traffic to screen, they can simply slip through the cracks.

4. Trojan Horse Malware- It offers attackers with a network backdoor, allowing them unrestricted access to the network.

5. Worms – They are the most prevalent and effective type of virus. Worms are spread by email or instant chat over a network.

1.2.2 Impact of Network Intrusion

Businesses that suffer data breaches face serious and growing implications. It is mostly due to the increasing regulatory burden associated with notifying individuals whose personal information has been exposed. The standards for notification and fines for organisations affected by a data breach vary by jurisdiction, both inside the United States and Canada, as well as worldwide [25].

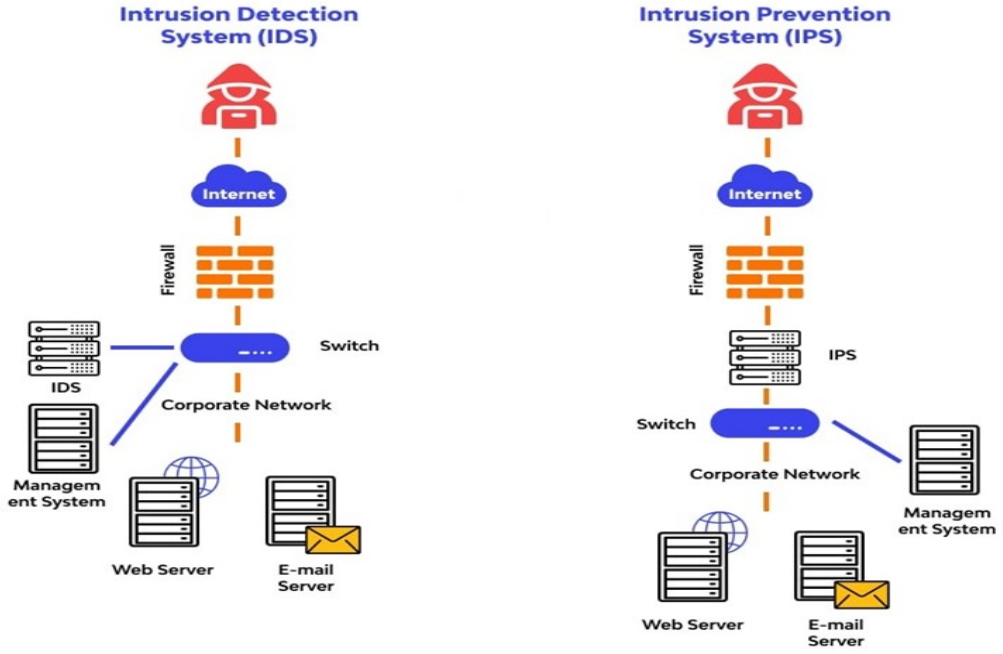

Figure 1.3: IDS and IPS process

Companies that have a consumer data breach must determine where their customers live and which regulatory entity has jurisdiction. Regulations specify the types of data that must be notified following a breach, as well as who must be contacted, how the notification must be carried out, and whether specific authorities must be alerted or not [26–28]. Personal, financial, and health data breaches are usually subject to notification obligations, however the exact definitions vary by state. Companies that do business globally may have customers in a range of jurisdictions and must adhere to a variety of regulations. The costs of such a process, combined with legal penalties, potential compensation for damages, and any resulting lawsuits, can be enough to put some businesses out of business [29].

Data breaches involving different sorts of data can have a significant negative impact on a company's brand and financial status. A data leak could jeopardise a company's planned sale, in addition to contractual duties, as happened recently with Verizon's purchase of Yahoo. Your firm may not survive

if your competitors learn about your business methods and are able to market products identical to yours at a lower price [9].

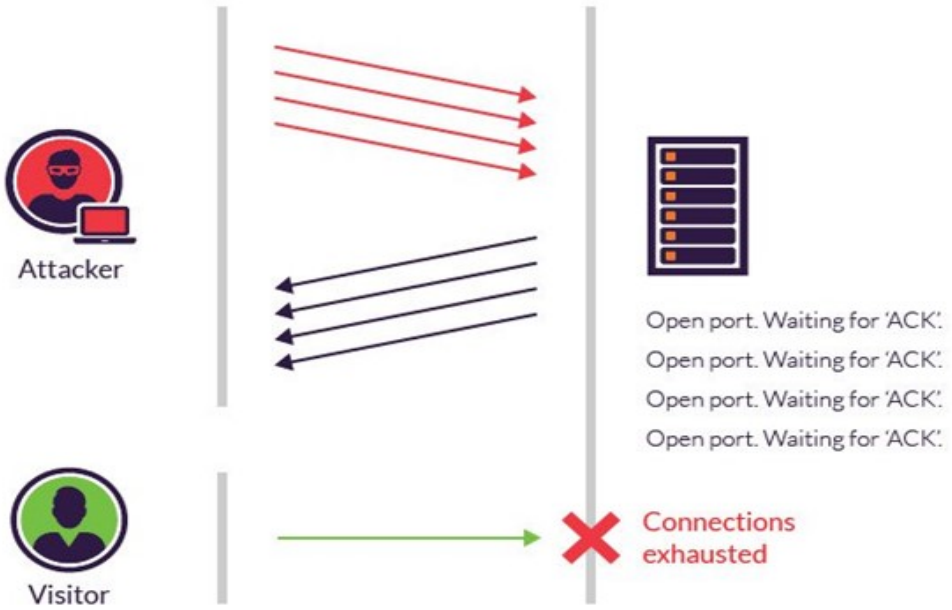

Figure 1.4: Process of Network Intrusion

1.3 Case Study - Facebook Data Breach

Personal data should not be accessed without authorisations [30–32]. Hackers gained access to the personal information of over 533 million Facebook users because of a data breach. The user's name, date of birth, current city, and wall posts were all included. A white hat security group discovered the vulnerability in 2021, and it has been active since 2019. The 2021 Facebook data breach is still fresh in many people's minds. Symantec, a cybersecurity outfit, was the one who brought it to light. Millions of people's personal information was exposed in the database, including mobile numbers, Facebook IDs, names, birthdays, and even some email addresses [33].

This breach occurred when fraudsters exploited a flaw in Facebook's contact importer to scrape data from the company's servers. As a result, they could acquire access to millions of people's personal information. While it's unknown what the crooks intended to do with all of this information, its speculated that it might be used for large-scale social engineering attacks in the future. Although Facebook recognised this as an external attack, the core cause of this and similar breaches is a typical occurrence: misconfiguration problems. The speed with which these breaches can escalate is what makes them so hazardous. Like in Figure 1.5, the data breaches in 2021 among various companies are being discussed [34].

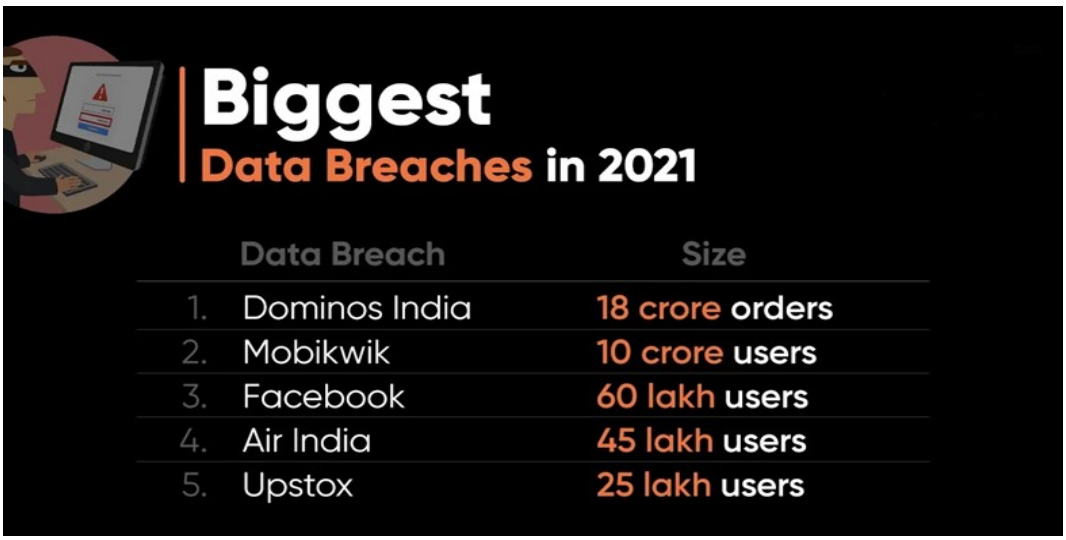

Figure 1.5: Data Breach statistics in Industries

1.4 Ransomware: A strategy of geopolitical destabilisation - The Case of Costa Rica

There has been an upsurge in the number of global hackers in recent years, aiming to better comprehend the world's realities and the implications that these truths have for governments. This is vital at this time since global tensions and instability are rising. Recent ransomware assaults on Costa Rican government institutions show a nexus between crime and aggressive military cyber activities. This is shown by current occurrences, which indicate that this convergence is taking place. Ransomware is malicious software that may be used to aid kidnapping and extortion by utilising data as a kind of ransom. In this context, "ransomware" refers to malicious software encrypting data and demanding payment in exchange for release. This study aims to investigate how flawed code and cyber-offensive capabilities are released to generate instability and influence public opinion. New techniques that require less participation in conventional kinetics have been designed to gain a swift victory from the inside.

1.4.1 Introduction

Recent changes in geopolitics have made global tensions worse, forcing governments all over the world to rethink how they run their countries and plan their agendas. The growing number of wars around the world was a key factor in the making of this new standard. The political and diplomatic conflict has gotten worse because of Russia's "special military operations" in Ukraine. This fight started because of what Russia calls "special military operations" that happened not long ago. During "regular" military operations in the Middle East, there used to be a time of calm. Now, because of this new development, it has become a separate phenomenon. This is because things have changed.

New realities in international relations and commerce are not only upending old international dynamics but also constitute a new frontier in the formation of strategic alliances. These alliances

seek to construct not merely a formidable and efficient bloc to restrict rivals' advances, but also a new paradigm of security and defence that goes beyond typical diplomatic exercises. This new security and defence paradigm includes the ability to use military force in addition to conventional diplomatic exercises [35].

In this context, cyber operations are critical because they have the same effect, destabilisation, and destruction potential as conventional kinetic operations. As a result, they have a major edge over combats that rely heavily on the employment of weapons. Cyber operations may have the same effect, disruption, and damage as traditional kinetic operations with conventional weapons. Viewing cyberspace as a possible place for military operations and the application of conflict resolution strategies may aid in the comprehension of contemporary conflicts. Even when these clashes occur in unanticipated locations, they are fueled by disinformation and manipulation of public opinion. Following recent incidents involving the hacking of Costa Rican government organisations, new cyberspace-based geopolitical conflicts are currently being detected. International criminal organisations seeking to profit from these events are employing ransomware (or kidnapping and extortion with data, abbreviated RSW) as a weapon and strategy that exploits the weakening of institutions and the cognitive vulnerabilities of nations to instigate chaos, thereby altering the dynamics of national societies through negative actions. Since ransomware is created to be both a weapon and a strategy, it may be used as both a weapon and a tactic (or abduction and extortion).

1.4.2 Ransomware in attention and its adversaries?

The tactic of data theft followed by extortion by third parties is not new in the field of malware attacks. It is an expression of the specialisation and sophistication of malware that takes advantage of a specific action taken by a user (such as opening an email, clicking on a link, displaying an image, etc.) to silently activate a malicious program, which deploys capabilities to identify files, take possession of them, encrypt them, and then take control of the machine, resulting in the machine being compromised (which is not a guarantee of full recovery of the files). If payment is not made, the 3/11 opponent may take one of two basic actions: delete or publicly reveal the material, with each option having potential implications When a country or organisation recognises that it is in an RSW situation, it has few legal options to escape the worst punishments. This is true since there aren't many good legal options. There are more options, such as purchasing cyber insurance, which, based on the kind of protection provided and the limitations imposed, may help your organisation or nation weather the storm. Protection against online threats. Working with an organisation that specialises in preventing cyberattacks is another possibility. It is quite improbable that the person who took your data will let you get it back, even if you bargain with them. Even if they could do so, which is very doubtful, this is still true. Alternately, if you have access to diplomatic lines, you may contact the relevant authorities and seek outside aid while using a variety of techniques to find the attacker and disable the encryption system. If there were adequate diplomatic routes, this might be a possibility. You need to be ready for everything that could happen.

Analysing RSW's manifestation necessitates considering not just the assailant's motivations but also the entity, nation, or person the assailant wanted to harm. Depending on the circumstances of the debate, the company in question, or the environment, the following factors may be considered:

- A company's health and safety procedures and management practices inspire confidence.
- Utilisation of a large number of control and safety systems by a complex (commercial or government) organisation.
- The catastrophe recovery and business continuity strategies are evaluated, and the results give useful data.
- As you traverse the RSW, you will be able to simulate a variety of environments and engage in several activities.
- A study of how people use public and corporate internet resources, as well as their online navigation habits.
- The capability of an organisation to establish and sustain an information security culture (including personal cyber-hygiene).
- To detect prospective threats, dormant and emerging risks are evaluated regarding present operations and plans, as well as the difficulties the nation is now facing.
- The term "risk aversion" may be defined in both the corporate environment and the personal life of an individual [36].

Deficiencies or results that fall short of expectations in any of the aforementioned elements will be attributed to a lack of management capacity in both the organisation and the country, or the person in charge, or their property, which ultimately translates to a potentially negligent action that can be demonstrated through auditing or independent verification [37, 38]. This may be shown by auditing or independent verification.

Among other things, the following may indicate the aggressor's perspective:

- We need information and the ability to widen our minds to do this.
- Individuals' various hopes and ambitions drive their behaviour in a variety of ways.
- It is possible to employ conventional or specialised equipment depending on the task.
- Transactions using bitcoin and other forms of monetisation, as well as previously observed patterns of behaviour, all of which are publicly available information, both locally and globally [39].

It is feasible to determine the origin of the attacker using any of the information in the list. There are various approaches to accomplish this objective [40–43]. This objective may be accomplished in several ways. If everything goes as planned, these two will be able to piece together a picture of an attacker's unlawful conduct, leading to his or her arrest. They will be able to piece together the aggressor's acts and determine how they fit together to produce a picture of the aggressor's unlawful behaviour if they collaborate. They will be able to piece together the aggressor's activities and understand how they connect to form a picture of the aggressor's unlawful behaviour. To remain safe in a constantly changing environment, it is essential to collect as much information as possible to create the most accurate maps possible. This will allow you to remain competitive. As a result, we will be able to fully use all of our resources [44].

1.4.3 Affect on the Nation

The Central American nation of Costa Rica is renowned for its peaceful and nonviolent way of life. This is stated in the nation's constitution. Countries, where the people are treated with respect and conciliation, provide the greatest motivation for sovereign governments to establish strategies for the general welfare and long-term health of the environment. This is because these nations are more likely to develop positive ties with their inhabitants. This is because individuals in these nations are more inclined to support and cooperate with their administrations. The United States is a superb example of a nation that exemplifies this notion. When greeting one another, Costa Ricans use the phrase "Pura Vida," which translates to "pure life" in English. This phrase may also be found in other settings. This expression is used in two distinct ways. By responding in this manner, people demonstrate the significance of Costa Rica's environmental regulations and their desire to embrace renewable energy. This tranquilly was recently shattered by cyberattacks on numerous public organisations after the inauguration of President Rodrigo Cháves' new government

- The Ministry of Finance.
- Ministry of Science, Innovation, Technology, and Telecommunications
- Ministry of Labor and Social Security
- The Social Development and Family Benefits Fund.
- NMI stands for National Meteorological Institute.
- The Social Security System of Costa Rica.
- Headquarters of the Alajuela Interuniversity

As a result, on April 17, 2022, a wicked and nasty criminal organisation known as Conti began performing a series of devious and unpleasant activities, and it just announced on the "Darkweb" that it had accessed more than 700 gigabytes of data.

1.5 Preventive measures

Any organisation's cybersecurity team aims to ensure that the infiltration is detected and prevented at the earliest possible stage. The following are some methods for detecting and preventing intrusion:

Network based intrusion prevention systems (NIPS)

The Network Intrusion Prevention System, or NIPS, is a system that watches for odd activity on a network in order to protect it from malware or cyber attacks. NIPS is an inline detection and prevention device that analyses network traffic and, if it detects anything suspicious, takes action based on a set of criteria. Because it performs hundreds of commands at once, NIPS unlike a microprocessor, is rapid and application-based [45].

Wireless based intrusion prevention systems (WIPS)

WIPS (Wireless Intrusion Prevention System) is a wireless network security system that scans the radio spectrum in a radio's wireless environment for unauthorised access. This technology is very valuable because it can automatically detect and shut down unauthorised entries. Modern WIPS are useful not just for detecting and preventing intrusions, but also for complying with requirements such as the General Data Protection Regulation (GDPR) [46].

Network Behaviour Analysis (NBA)

This technology ensures the security of a network by monitoring traffic and identifying any unauthorised access. In order to do a full offline analysis, NBA monitors the network and generates data packets. The NBA is a tool that is used to relieve network administrators of their duties [47].

Host based intrusion prevention systems (HIPS)

In most cases, HIPS defends the host computer from harmful attacks. From the network layer to the application layer, HIPS is active. A HIPS analyses system calls, application logs, and file-system updates to discover intrusions using a database of monitored system objects (binaries, password files, capability databases, and access control lists) [32, 48–50].

1.6 Contributions

This research work is focused:

1. To identify the existing DL algorithms, which can work best for analysing the network features.

2. To identify the role of Deep Learning in Network Intrusion.
3. To investigate Deep Learning models developed for Network Intrusion Detection.
4. To achieve the best possible network intrusion detection by making use of Deep Learning algorithms.
5. To minimise the human intervention in network intrusion detection.
6. To maximise the accuracy rate of detection.
7. To design a DL classifier that can automatically detect intrusion attacks.
8. To design an efficient mechanism for computing the effect of network intrusion constraints.

1.7 Thesis Organisation

The rest of the thesis is organised as:
Chapter 2: The ten different network intrusion techniques are explored, such as Botnet, malware. The preventive measures along with the research vision is being discussed (This chapter emphasizes on Research Contribution 1).
Chapter 3: The role of deep learning in network intrusion is discussed. The various existing deep learning models for network intrusion detection are well discussed (This chapter emphasizes on Research Contributions 2 and 3).
Chapter 4: The network intrusion is successfully detected with deep learning models in the proposed framework as discussed in this chapter (This chapter emphasizes on Research Contributions 4-8).
Chapter 5: Finally, this chapter concludes the proposed thesis and future work.

Chapter 2

Background: Research Vision

2.1 Network Intrusion

An unauthorised entrance into your network or an address in your respective domain is referred to as a network intrusion. An intrusion can be passive (in which access is achieved quietly and undetected) or active (in which access is gained covertly, without detection and changes to network resources are affected). Intrusions might occur from outside or from within your network structure (an employee, customer, or business partner). Some invasions are just aimed to alert you that an intruder has entered your site and is defacing it with various messages or obscene graphics. Others are more malicious, attempting to harvest sensitive data on a one-time basis or as part of a long-term parasitic connection that will continue to syphon data until it is identified. Some intruders will try to implant code that will crack passwords, capture keystrokes, or imitate your site while redirecting unwitting users to their own. Others will infiltrate the network, stealthily syphoning out data on a regular basis or altering public-facing Web pages with varied messages [51].

An attacker can acquire physical access to your system (by physically accessing a restricted computer and its hard drive and/or BIOS), externally (by assaulting your Web servers or breaching your firewall), or internally (by physically accessing a restricted machine and its hard disc and/or BIOS) (your own users, customers, or partners).

There are lots of examples about how network risks and assaults may wreak havoc on your network's security and applications [52–54]. Furthermore, as individuals become more reliant on digital communication technology, the number of frequent types of network attacks is increasing [55, 56]. Top ten networking dangers and assaults are discussed here:

2.1.1 Computer Virus

One of the most common network security dangers is computer viruses, that can result in major data loss. They're a sort of malware that contain of one-of-a-kind pieces of code that can wreak havoc on computers and spread from one to the next. Did you know that computer viruses have infected at least 30% of all computers on the planet? Malware infections are frequent, and a Trojan horse

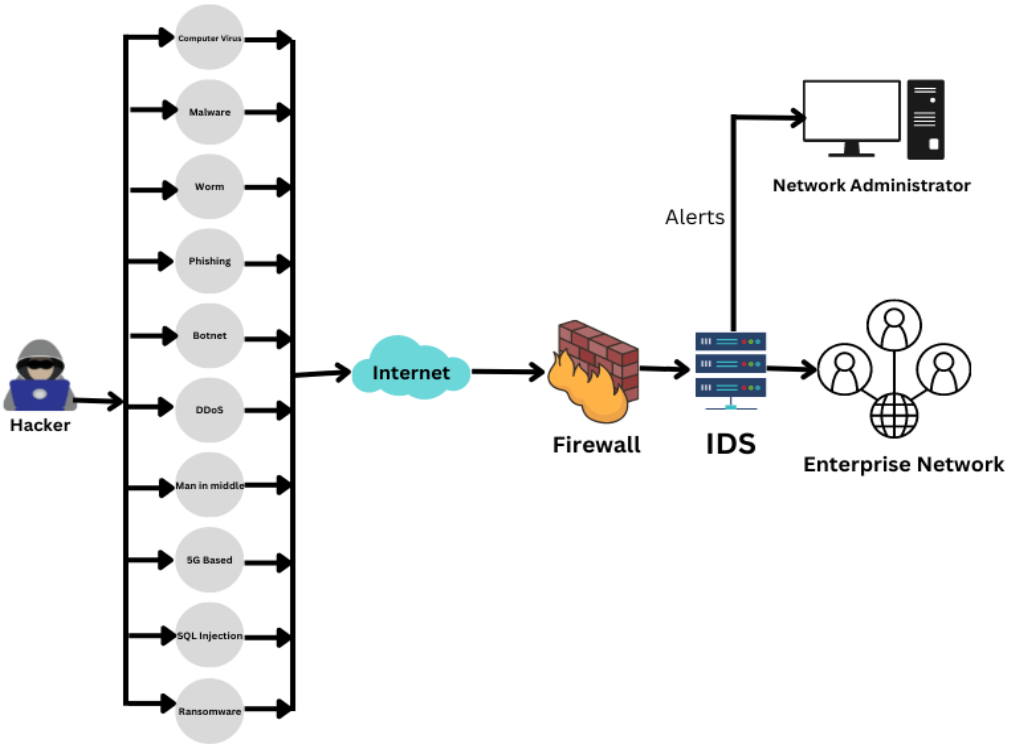

Figure 2.1: Different types of attacks and working of IDS

masquerading as a virus can do significant damage to a computer network. These viruses can corrupt your files, infect other devices on your list, and steal your personal information if you click on a bad link in an email or download links from infected websites [57].

2.1.2 Malware

One of the most dangerous cyber crimes that can inflict substantial damage is a malware attack. Hackers use harmful software, sometimes known as malware, to gain unauthorised access to a target system and disrupt or corrupt its files and data. It can also harm the network's internal and external endpoint devices. Malware can infect networks and devices, with the goal of causing harm to the devices, networks, and/or people. The user or endpoint may be harmed in numerous ways depending on the type of malware and its intent. Malware can have a mild and benign effect in some circumstances, but it can also be disastrous in others [58]. Whatever mechanism is used, all types of malware are meant to exploit devices at the expense of the user and in favour of the hacker who created and/or delivered the malware.

2.1.3 Computer Worm

The phrase "computer worm" first appeared in John Brunner's novel "The Shockwave Rider" in 1975. The protagonist of the story constructs a data-gathering worm in this novel. Worms were created in the early days of computer science to exploit a system's flaws. Instead of causing significant damage to the infected machines, they just multiplied in the background. However, the function of computer

worms has evolved through time. Today, criminals frequently employ them to obtain complete control of their victims' machines [59].

Computer worms can infect computers connected to a network. A computer worm is a type of malware that replicates and spreads across the internet. Rather than infecting computer files, a computer worm frequently infects another machine on the network. A worm accomplishes this by proliferating. The worm's clone inherits this ability, allowing it to infect additional systems in the same way. This section also explains the difference between computer worms and viruses. Computer worms are self-replicating programs that run in the background, whereas viruses require a host file to infect. Machine worms are a harmful sort of malware that replicates itself and spreads from one infected computer to another. They achieve their goals by taking advantage of network flaws. Furthermore, it has the ability to affect your system without the assistance of other users [59].

2.1.4 Phishing

Phishing is a type of social engineering assault that is frequently used to obtain sensitive information from users, such as login credentials and credit card details. It happens when a hacker poses as a trustworthy entity and convinces a victim to open an email, instant message, or text message. The recipient is subsequently duped into clicking a malicious link, which can result in malware installation, system freeze as part of a ransomware assault, or the disclosure of sensitive information. An attack has the potential to be devastating. Unauthorized purchases, money theft, and identity theft are examples of this for individuals [60].

Furthermore, phishing is frequently used as part of a bigger attack, such as an advanced persistent threat (APT) event, to build a foothold in business or governmental networks. Employees are compromised in this scenario in order to circumvent security perimeters, distribute malware inside a closed environment, or get privileged access to protected data. An organisation that falls victim to such an attack usually suffers significant financial losses as well as a loss of market share, reputation, and consumer trust. Depending on the extent, a phishing attempt could turn into a security disaster from which a company may struggle to recover [61].

2.1.5 Botnet

A botnet [short for bot network] is a hacker-controlled network of hijacked computers and gadgets infected with bot software. The bot network may be rented out to other hackers and used to disseminate spam and perform Distributed Denial of Service [DDoS] assaults. Botnets can also exist without a command and control (C&C) server by transferring commands from one bot to another via peer-to-peer [P2P] architecture and other management channels. Botnets made up of linked devices have become more prevalent as the internet of things (IoT) develops and becomes more widely used [62].

IRC clients were initially used by botnet operators to deliver instructions and carry out DDoS operations. Botnets (as shown in Figure 2.2) have been spotted mining bitcoins, intercepting data in

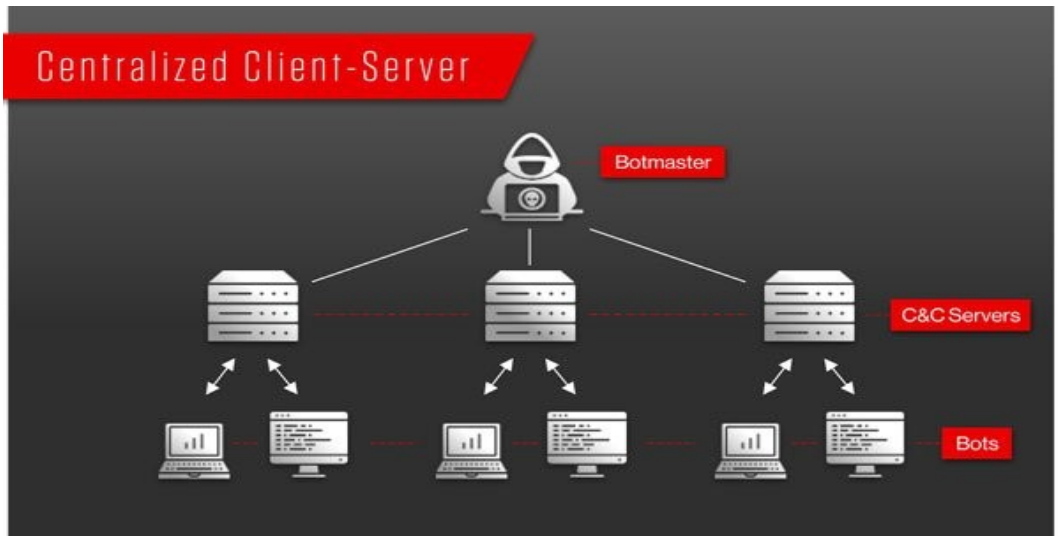

Figure 2.2: Concept of Bots

transit, sending logs containing sensitive user information to the botnet master, and using the user's PC resources in recent months.

2.1.6 DoS (Denial of Service) and DDoS Attacks

At some point or another, we have all probably encountered website crashes. A rush in website traffic, whether due to a product introduction, a new promotional plan, or a sale, might cause the server to crash. Cyberattacks in the form of DoS and DDoS attacks, on the other hand, can cause website breakdowns. As a result of the malicious traffic overload, the system fails, and users are unable to access the website. The goal of these types of network security assaults is to bring the victim network's IT infrastructure to a halt [63]. DoS attacks differ from DDoS attacks in that hackers initiate DoS attacks over a single host network. DDoS attacks are more complex, and attackers can hack targeted systems with several computers. DDoS attacks are difficult to detect since they are launched from multiple hacked systems [64].

2.1.7 Man-in-the-middle

A Man in the Middle (MitM) attack happens when an unauthorised entity intercepts a communication between two systems or people. The interceptor tries to eavesdrop on the conversation or impersonate one of the permitted parties in such a way that the intrusion is not noticed. A MitM attack's goal is usually to intercept transmissions of personal data that could be valuable, sensitive, or profitable if used fraudulently (e.g. logins, account details, credit card information, etc.) [65–67].

MitM (as shown in Figure 2.3) attacks are typically targeted at financial and e-commerce businesses, as well as other websites that require authenticated logins. Stolen information can then be used for identity fraud, unauthorised financial transfers, or sold to a third party [68].

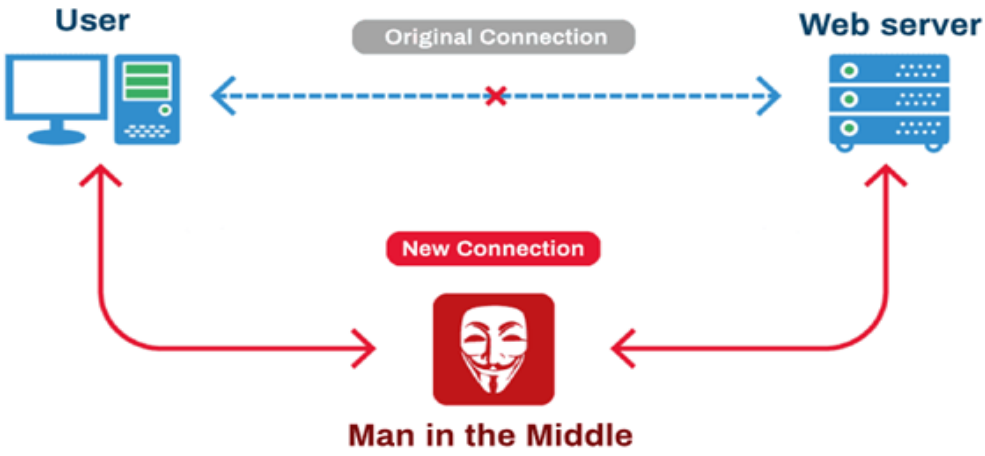

Figure 2.3: Man-in-the-middle attack

Figure 2.4: Ransomware attack

2.1.8 Ransomware

In 2022, ransomware assaults are on the rise. We've recently seen a slew of such threats, all of which have resulted in serious consequences. Ransomware is a type of malicious software that encrypts all files on a victim's computer, network, or server. Other ransomware operations use weak passwords and other vulnerabilities to obtain access to a network and encrypt files until a ransom is paid in exchange for the decryption key. Ransomware (Figure 2.4) is a sort of malicious software (malware) that threatens to expose or limit access to data or a computer system, generally by encrypting it, unless the victim pays the attacker a ransom price. The ransom demand is frequently accompanied by a deadline. If the victim does not pay the ransom in a timely manner, the data will be lost forever, or the ransom will be increased [69].

These days, ransomware assaults are all too common. It has affected major corporations in both

North America and Europe. Cyber criminals will target any individual or firm, and victims will come from a variety of industries. Several government authorities, including the FBI, and the No More Ransom Project, advise against paying the ransom to avoid promoting the ransomware cycle. Furthermore, half of those who pay the ransom are at risk of future ransomware assaults, especially if the malware is not removed from the system.

2.1.9 5G Based Attacks

Attacks based on 5G technology are a more advanced type of network security threats. While 5G networks allow for faster data transfers, they also increase the risk of cyberattacks. Hackers are leveraging 5G devices to launch swarm-based network security assaults against numerous systems, mobile devices, and IoT (Internet of Things) networks. In addition, the attacker has the ability to make changes in real time [70].

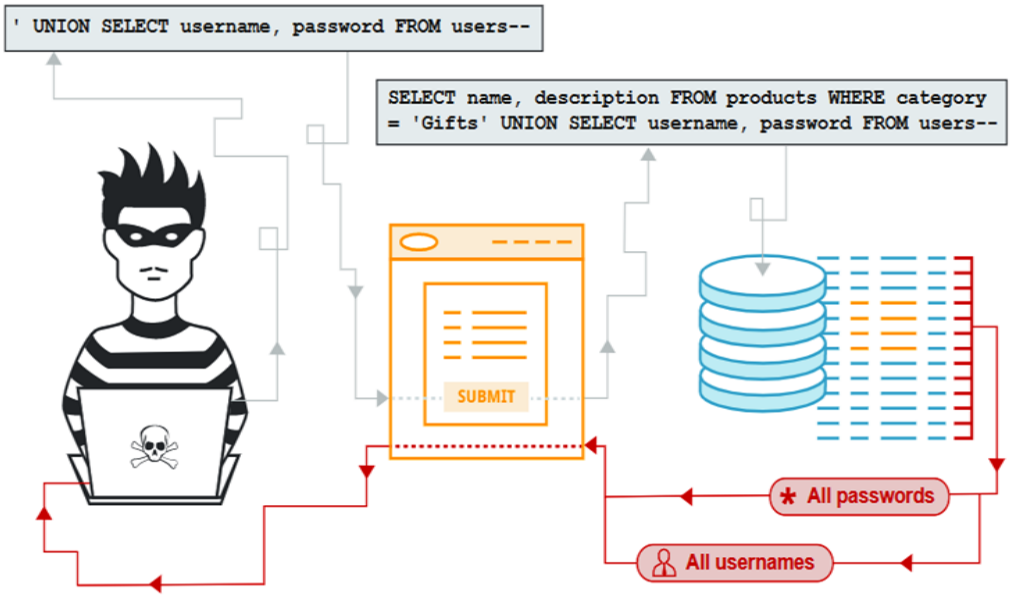

Figure 2.5: SQL attack technique

2.1.10 SQL Injection Attacks

SQL Injection (as shown in Figure 2.5) is one of the most common attack methods used by hackers to steal data. This type of network assault is common in poorly built programs and websites. Hackers can easily target them by changing the scripts because they contain vulnerable user-input fields (such as search and login pages, product and support request forms, comments sections, and so on). SQL injection is a severe threat and one of the most prevalent hacking techniques. It can also easily infect or exploit any website that uses a SQL-based database. A company's competent personnel must possess the essential skill set [71].

2.2 Recent Network Attacks

Network attacks are still a concern for businesses as they move to distant operations and rely more on confidential network interactions. Recent network assaults have demonstrated that malicious actors can strike at any time. As a result, cyber security and alertness should be a top priority in all industries.

2.3 Social Engineering

According to ISACA's State of Cybersecurity 2020 Report, social engineering is the most common network attack method, with 15% of penetrated parties citing it as their infiltration vehicle [72]. Social engineering entails sophisticated deception and trickery techniques, such as phishing, that take advantage of users' trust and emotions to get access to their personal information. The skill of manipulating individuals so that they divulge personal information is known as social engineering. The types of information these criminals seek can vary, but when you're targeted, they're usually trying to trick you into giving them your passwords or bank information, or into gaining access to your computer so they can secretly install malicious software that gives them access to your passwords and bank information as well as control over your computer [73].

Social engineering is used by criminals as it is usually easier to exploit your natural tendency to trust than it is to figure out how to hack your program. For example, convincing someone to give you their password is far easier than attempting to hack their password.

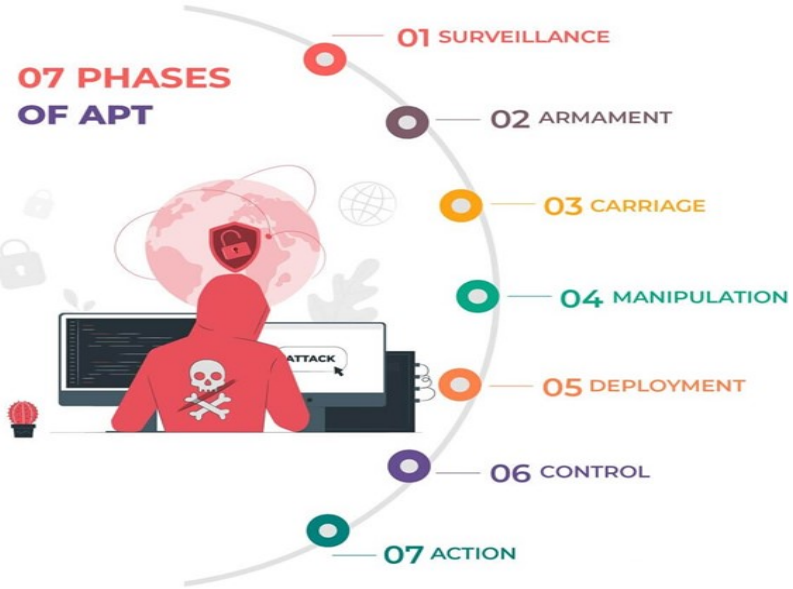

Figure 2.6: Phases of APT

2.4 Advanced Persistent Threats

An advanced persistent threat (APT) is a generic term for an attack operation in which an intruder, or a group of intruders, establish a long-term unlawful presence on a network in order to harvest extremely sensitive data [74].

The targets of these attacks, which are meticulously selected and researched, are usually huge corporations or government networks. The ramifications of such invasions are numerous, and include:

- Theft of intellectual property (e.g., trade secrets or patents)
- Compromise of sensitive information (e.g., employee and user private data)
- Sabotaging of critical organisational infrastructures (e.g., database deletion)
- Site takeovers in their entirety

An APT (Figure 2.6) attack takes more resources to carry out than a normal web application attack. The culprits are usually groups of well-funded cyber criminals with a lot of experience. APT attacks are sometimes supported by the government and employed as cyber warfare weapons [75].

Traditional web application dangers differ from APT attacks in the following ways:

- They're a lot more complicated.
- They aren't hit-and-run attacks; once a network has been breached, the culprit stays to collect as much data as possible.
- They're thrown indiscriminately against a vast pool of targets and are manually executed (not automated) against a specific mark.
- They frequently seek to infiltrate an entire network rather than a single component.

Perpetrators typically use more conventional techniques like remote file inclusion (RFI), SQL injection, and cross-site scripting (XSS) to gain a foothold in a targeted network. Trojans and backdoor shells are frequently employed to further establish a foothold and establish a persistent presence within the targeted perimeter.

2.5 Protection from Network Attacks

Changing network assaults necessitate a proactive and current network security strategy. The NGFW (Next Generation Firewall) from Forcepoint provides modern businesses with a set of sophisticated tools for detecting and responding to the most devious attacks within their networks.

With a clear breakdown of ongoing activities, the NGFW's real-time monitoring interface allows users to react swiftly to even the smallest network irregularities. NGFW prioritises vital networks and devices while detecting the most devious network assaults that get past traditional firewalls [28].

Furthermore, Forcepoint's next-generation firewall protects user privacy while performing decryption operations that successfully detect possibly stolen or compromised data in SSL and TLS traffic [76].

With a firewall solution designed to close the evasion gap, you can avoid camouflaged network attacks. Learn how to use the Forcepoint technique to improve your company's data security standards as it undergoes digital transformation.

2.6 Datasets

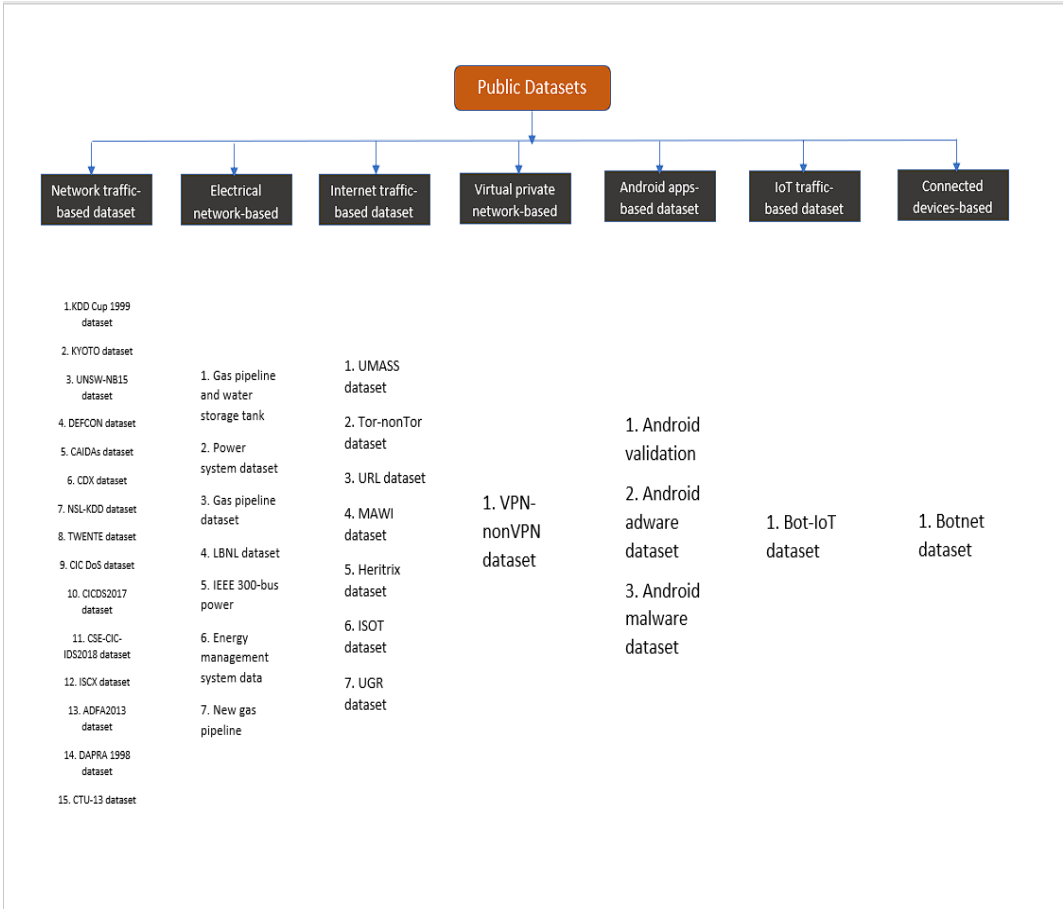

Figure 2.7: Public datasets available

2.6.1 KDD Cup 1999 dataset

The data was collected in 1998 via DARPA's IDS evaluation program [77]. Approximately 4,900,000 vectors were created over seven weeks of network congestion. Four types of simulated assaults have been identified: User to Root (U2R), Remote to Local (R2L), Probing, and Denial of Service (DoS). Each of these attack types is indicated by a distinct acronym (DoS). Each of the 41 characteristics in this dataset may be categorised into one of three distinct classes: (1) fundamental features, (2) traffic features, and (3) content features. To access the fundamental functionality, a TCP/IP connection must be established. Those with the "same host" personality are separated from those with the "same service"

Deep Learning Approach	Dataset Used	Performance Metrics
Deep neural network	NSL-KDD dataset	Precision, Recall, F1-score, Accuracy, ROC Curve
	KDD Cup 1999 dataset	Accuracy, Precision, Recall, F1-score
Feed-forward deep neural network	NSL-KDD dataset	Accuracy, Precision, Recall
Recurrent neural network	KDD Cup 1999 dataset	Detection Rate, FAR, Efficiency
	NSL-KDD dataset	Accuracy, Precision, Recall
	CICIDS2017 dataset	Accuracy, Detection Rate, FAR
Convolutional neural network	KDD Cup 1999 dataset	Accuracy, Precision, Recall, F1-score
	CICIDS2017 dataset	Accuracy, Precision, Recall
Restricted Boltzmann machine	KDD Cup 1999 dataset	Accuracy
	ISCX dataset	Accuracy, TPR, TNR
	NSL-KDD dataset	Accuracy
Deep belief network	KDD Cup 1999 dataset	Detection, Detection rate, FAR
	NSL-KDD dataset	Accuracy, Detection rate, FAR, Precision, Recall
Deep auto-encoder	NSL-KDD dataset	Accuracy, Precision, Recall, False Alarm, F-score
	UNSW-NB15 dataset	Accuracy, Precision, Recall, F-measure, FAR
	NSL-KDD and UNSW-NB15	Accuracy, Precision, Detection rate, Recall, FPR, F1-score
Denoising auto-encoder	KDD Cup 1999 dataset	Accuracy, Test classification error
Deep migration learning	KDD Cup 1999 dataset	Detection rate, FAR, Precision, Missing rate
Self-Taught Learning	KDD Cup 1999 dataset	Accuracy, Precision, Recall, F-measure

Table 2.8: Comparison of different Datasets

personality. There may be questionable behaviour in one of the content items inside the data area. Since its first release in 1999, the KDD'99 data set has mostly been used to evaluate the performance of anomaly detection systems. This data collection was compiled using information collected from DARPA's 1998 IDS evaluation program. DARPA'98 consists of about 4 terabytes of suppressed raw (binary) TCP dump data collected during a seven-week period of network activity. This information may be separated into around 5 million connection records, with each record containing approximately 100 bytes of information. During the two-week trial period, about two million connection registrations were made on this platform. The KDD training dataset has around 4,900,000 distinct connection vectors. Each connection vector includes 41 characteristics that may be attacked in either standard or attack mode. Four distinct forms of imitative attacks may be distinguished as following:

1. An attacker may conduct a denial of service attack, also known as a DoS assault. This attack may include making a computer or memory mode too busy or full to efficiently process genuine requests or preventing authorised users from accessing a system. Both of these methods are considered "denial of service."

2. In a User-to-Root Attack, the attacker first acquires access to a system using password sniffing, a dictionary attack, or social engineering and then exploits a system vulnerability to gain root access (U2R). User-to-root exploits are a kind of attacks that allows any user to get administrative

privileges for a system. This kind of security vulnerability is known as a ”User to Root Attack.”

3. A worm attack occurs when an adversary has the capacity to transmit packets across a network but does not have an account on that system. Such an assault is only successful if the network is not password-protected. A local attacker exploits a vulnerability to get system access in the same way as a user.

4. An endeavour to gather as much information as possible about a computer system or network. This kind of attack aims to circumvent the protections designed to secure the network.

   The choice to draw training data from one probability distribution and test data from a separate probability distribution makes the work seem more realistic. In addition, there are several attack types that were not included in the training data but are included in the testing data. Some experts in the field of network intrusion prevention have theorised that the vast majority of newly reported attacks are simply updated versions of previously observed attacks, and it may be possible to identify new variations of attacks by comparing their signatures to those of previously observed attacks.

2.6.2 NSL-KDD dataset

This compilation of data, prepared by Tavallaee, offers suggestions for enhancing the KDD 1999 dataset. Unlike the previous KDD dataset, the NSLKDD dataset [78] contains the following mileage information: How well does this collection meet the following requirements? The following criteria are met by the work: The number of records are adequate because (1) it does not include any unnecessary records; (2) it does not contain any duplicates of other records, (3) the different numbers of chosen records are expressed as a proportion of the overall number of records, and (4) no records are missing. The great majority of research on intrusion detection indicates that the NSL-KDD dataset yields more accurate findings when comparing the performance of both datasets (the KDD Cup 1999 dataset and the NSL-KDD dataset).

The performance report for the NSL-KDD dataset is one example. The NSL-KDD data set is analysed via a range of clustering approaches accessible through the WEKA data mining platform. The NSL-KDD dataset was divided and organised into four categories, each corresponding to one of the four most prevalent forms of attack. Both the test data and training data are thoroughly analysed. Researchers are now examining the timelines at which various clustering methods may be implemented. Training and testing-related information account for 20% of total data. Utilizing the NSL-KDD dataset, this study illustrates how network-based attacks are executed by attacking the protocol with the most severe security weaknesses. The KDD cup99 dataset has been decommissioned in favour of the more current and comprehensive NSL-KDD data collection. The NSL-KDD dataset has been analysed by a variety of researchers using a variety of techniques and tools. The ultimate goal of these investigations was to create a reliable intrusion detection system. One of these studies, which is included in the

WEKA software, utilises a variety of machine learning techniques to do a comprehensive analysis of the NSL-KDD data set.

The outcomes of this study have been included in the system. K-means clustering is a method for analysing and training responses to a diverse range of threats [79–81]. In this procedure, the NSL-KDD data gathering system is used. Within the framework of the Self-Organization Map, a neural network that was trained using the KDD99 cup data set is used (SOM). This is performed so that comparisons may be made between the NSL-KDD data set and the preceding data set. The NSL-KDD dataset largely consists of four kinds of assaults:

1. DoS is an acronym for ”denial of service,” which refers to an attack in which the target is incapable of responding to or processing valid requests because all of its resources have been depleted. This kind of flooding includes sync floods, for instance. In this context, the terms ”source bytes” and ”percentage of packets with mistakes” are particularly pertinent. Rate of packet error and source byte transmission.

2. The objective of port scanning attacks, along with other types of probing attacks like spying, is to obtain information about a distant target. This section should include references to ”connection time” and ”source bytes.”

3. U2R attacks include unauthorised access to local super user (root) privileges. Using a standard account, an attacker connects to a target system and attempts to get root or administrator capabilities by exploiting a vulnerability in the victim’s system, such as a buffer overflow attack. This is often known as a U2R attack. U2R is an acronym for ”unauthorised access to local super user (root) privileges.” Two crucial characteristics are the ”number of shell prompts triggered” and the ”number of files produced.” Both may be found in the column titled ”number of files created.”

4. The phrase ”remote-to-local” (R2L) signifies that an attacker hacked a remote workstation to get access to a local system. The prevalent practice of ”guessing” passwords is a great example. However, ”the number of unsuccessful login attempts” data at the host level is far more important than ”duration of connection” and ”service requested” data at the network level.

Classification is a branch of data mining in which data instances are allocated to a variety of groupings. Throughout the years, several classification algorithms have been created in an attempt to exceed the competition. Their correct functioning requires mathematical approaches such as decision trees, linear programming, and neural networks. These techniques analyse the given data in a variety of ways in order to provide a prediction. Classification is a type of data mining in which data occurrences are classified into different categories.

Throughout the years, several categorization algorithms have been developed in an effort to outperform one another [82–84]. Mathematical approaches such as decision trees, linear programming, and neural networks are required for best performance. In order to make a forecast, these methods apply a variety of techniques to analyse the provided data. The Decision Tree approach is used to

decompose the classification issue into its component pieces. The process begins with the creation of a decision tree, which is then used throughout the model-building phase. The phrase ”neural network” refers to a certain kind of model that, with adequate training data, may be used to estimate or approximate functions. Neural networks are employed to accomplish this aim.

A person’s ”nearest neighbour” is the one whose residence is physically closest to their own. Each node in a decision tree represents a class acquired from the training data set and is used to decompose the larger classification issue into its component pieces [85–87]. First, a decision tree is constructed, and then it is utilised in a modelling technique that may be used for the classification of diverse items. Neural networks are a kind of statistical learning model that may be used to estimate or approximate functions given adequate quantities of training data [88, 89]. Utilizing neural networks are used for this purpose. A person’s “nearest neighbour” is the person who lives in the house immediately next to their own. This approach considers all of the classes in the original dataset before using a similarity metric to determine how to classify new data [90–92].

2.6.3 DEFCON dataset

The DEFCON8 (2000) and DEFCON-10 editions were used to create the first version of this dataset (2002) [93]. Port scanning and buffer overflow are two of the attacks contained in the DEFCON-8 exploit collection. Not only does the DEFCON-10 database include information on probing attacks, but it also has information on a vast array of other forms of attacks (e.g., bad packets, ports scans, port sweeps, and so on). Nehinbe Ojo Joshua’s [80] classification approach for network intrusions employs these two criteria. Another extensively used dataset for IDS testing is the DEFCON3 data collection. The information was gathered during a Capture the Flag hacking contest (CTF).

In this sport, competitors pick either an offensive or defensive position. This collection was accumulated over the whole tournament. Since CTF traffic consists only of intrusive traffic and none of the regular background traffic is seen on a network, it differs significantly from the traffic observed in the real world. Only in the context of evaluating the alternative alert correlation algorithms has the DEFCON dataset been shown to have any use. This limitation restricts the possible use of the data. The significance of IDS research in securing computer networks has increased.

An intrusion detection system (IDS) can recreate the chronological order of an attack’s chronology. In the case of a security breach, many notifications will be sent. However, its usage has led to an alarming rise of alerts and warnings. According to one research, 95% of these warnings were false positives. The primary causes of false alarms are the limitations of intrusion detection systems (IDS), the specificity of detection signatures, its reliance on the environment, isolation of the IDS from the rest of the system and the network, and the difficulty in distinguishing between abnormal and normal behaviour. To add insult to injury, it may be difficult to distinguish between abnormal and normal processes, which may also result in false alarms. Also, contributing to the frequency of false warnings is the difficulty in differentiating aberrant from regular behaviour.

The primary objectives of these investigations are to reduce the number of false alarms, to investigate the factors that contribute to false positives, to identify high-level attack scenarios, and to provide a consistent response to attacks by understanding the interrelationship between various alerts. The major purpose of this essay is to examine the techniques used to identify complicated attack scenarios. We generate multi-step attacks from raw warnings using a modified version of the Left-to-Right (LR) Parser technique. This allows us to now develop multi-stage assaults. On the secure network, distributed sensors are responsible for generating alerts, which are then relayed to a centralised database. IDS generates tokens without comprehending their connections, similar to lexical analyzers.

Here, the operations of the IDS produce several independent attacks without understanding their interdependence. These connections may be quite hazardous. The complexity of the parser, which must determine the correct sequence of tokens and the grammar to which each token belongs, seems comparable to the difficulty of encountering unique conditions. You must determine not only where in the grammar each token belongs, but also how they should be utilised.

2.6.4 CAIDA dataset

The Center for Applied Internet Data Analysis has provided datasets such as CAIDA DDOS, CAIDA Internet traces 2016, and RSDoS Attack Metadata to this repository (2018-09). The CAIDA DDOS assault comes from the Equinix data centre in Chicago and consists of one hour of DDoS traffic split into pcap files every five minutes [94]. Additionally, DDoS attack traffic is separated. The analysis of UCSD Network Telescope backscatter packets was used to deduce the features of DDoS assaults generated randomly using the RSDoS Attack Metadata. This is what our analysis of the RSDoS Attack Metadata taught us. This 2007 dataset contains data collected from network activity logs after Distributed Denial-of-Service (DDoS) attacks. The usual flow of traffic on a computer or network that is the target of a denial-of-service attack may be disrupted by flooding it with network packets.

In this manner, real messages are prevented from ever reaching their intended destination. A significant shortcoming of the CAIDA dataset is the lack of variety in the assault types. In addition, the collected data lacks components that are consistent throughout the whole network, making it difficult to distinguish between abnormal and normal traffic flows. Our research focuses mostly on the distributed kind of distributed denial of service attacks (DDoS). A bot-master is responsible for providing commands to a collection of infected computers or other electronic devices (”bots”) situated in various regions of the globe to attack a single target. Simply described, a botnet is a huge, geographically scattered, maliciously infected computer network. Botnets are often referred to as zombie networks.

Numerous studies have been undertaken on the subject, yet the Internet community remains at great danger from DDoS attacks, regardless of their form. The first incident of this kind was documented about a decade ago. Damage caused by DDoS assaults has progressively increased over the last many years. This kind of assault has become more complicated and difficult to detect due to advancements in

communication technology, which have increased the difficulty of detecting such attacks. Using a mix of flash crowd agents, slow rate assaults, and amplification attacks, it exploits a DNS server weakness. The speed with which an organisation can identify, react to, and recover from a DDoS attack is critical to its existence. To assess the current status of the network, a DDoS attack simulation was conducted. To get a deeper understanding of the particulars of each step of an assault, researchers used machine learning methods.

There is a detailed daily plan to be implemented. Daily, we gathered raw data from every system, including event logs and network traffic captures (Pcaps) (Windows and Ubuntu event logs). We used the CICFlowMeter-V3 to detect and isolate more than eighty separate characteristics of the traffic as part of the feature extraction process. Then, we created a CSV file containing information about each computer.

2.6.5 CIC DoS dataset

This suite of application layer Denial of Service attacks includes four unique tools and four distinct attacks. According to the CIC DoS dataset made available by Jazi et al. [95], application layer DoS assaults may be either high-volume or low-volume in terms of traffic volume. This category includes high-volume HTTP attacks, such as those used by HTTP Unbearable Load King (e.g., low-volume HTTP attacks generated using HTTP Unbearable Load King). Another word for the same situation is high-volume HTTP assaults (hulk). Low-volume HTTP assaults include the slow-send body attack, the slow-send body (RUDY) attack, the slow-read attack, the slow-send headers attack, and the slow-send headers attack. The slow-send headers attack is a further example. Researchers have begun to focus on application layer Denial of Service (DoS) attacks rather than network-based DoS attacks since their prevalence on the Internet has increased. This is due to the decreasing frequency of DDoS attacks launched from inside a network. Depending on the attack's features, application layer DoS assaults may generate either a large volume of traffic or a negligible quantity.

Flooding refers to a large-scale assault that similar to a natural disaster, overwhelms a system. The vast quantity of application-layer requests supplied to a victim enables for their differentiation. By sending a little quantity of attack traffic to the target, it is possible to execute a low-volume distributed denial of service assault. Three forms of distributed denial of service attacks are distinguished: Sending and receiving data at a slower rate than normal is a common cyberattack strategy that may do substantial damage to the victim. Slow-rate attacks exploit the timing parameters of the server by transmitting and receiving data at a slower pace than normal. A one-shot assault will overwhelm a target with a single connection or request (e.g., Apache Range Header attack).

Due to the limited focus of one-time attacks on a particular vulnerability in an application's protocol or service, we decided to focus on a more broad kind of application. There are two forms of Denial-of-Service slow-rate attacks: Reading and presenting data takes an eternity. For the simple reason that in low-volume production, less is more. Capability to halt a service via a denial of service attack without requiring the attacker to invest considerable time or money. This is due to the fact that

the attacks were designed to generate exactly the quantity of traffic necessary to disrupt the targeted service; in other words, assaults ceased after a server ceased responding to requests. This led us to conclude that a fair amount of traffic for a short period of time was sufficient to execute these attacks effectively. Due to this, we have arrived at this conclusion.

2.6.6 DARPA 1998 dataset

This dataset, which was made public for the first time in February 1998, was constructed by integrating a computer system's audit logs with network traffic. We collected the necessary test data over the course of two weeks. It took seven weeks of actual network-based assaults to acquire training data. Sharafaldin [96] and his colleagues assert that the figures do not accurately reflect the behaviour of real networks. In 1998, DARPA conducted an offline test and a real-time assessment known as the DARPA Intrusion Detection Evaluation. We simultaneously did both sorts of testing. Researchers studied offline traffic and audit log data from a virtual network to evaluate the efficacy of several intrusion detection solutions.

This was performed in order to extrapolate the findings. Using batch processing, computers analysed data to identify attack sessions that were outside of conventional procedures. Access to intrusion detection equipment for real-time testing has been granted to the Air Force Research Laboratory (AFRL). These devices were installed in the AFRL's network test bed so that researchers could monitor traffic in real time for signs of intrusion attempts. As there are currently a few publicly available datasets, various cloud-based intrusion detection systems (IDS) have used the DARPA dataset for model development and validation.

On the basis of these findings, it is evident that the DARPA dataset is unsuitable for assessing cloud IDS since it lacks crucial statistical features seen in real-world cloud traffic data centres. This dataset is not suitable for testing cloud IDS and should not be used for this purpose. These qualities are essential because they reflect how well the dataset will function in more realistic environments. As an alternative, we propose the creation of a new public dataset that includes both positive and negative data. Our lab cooperated with a cloud service provider that contributes 100 percent of its revenue to charity in order to generate this data collection. As part of a novel hypervisor-based cloud intrusion detection system, we provide an instance-oriented feature model and supervised machine learning algorithms (IDS).

Logical regression, random forest, and support vector machine are among the methodologies being researched. It was constructed using the IDS assessment method developed at the MIT Linco ln Laboratory. In 1998 and 1999, this strategy was used, resulting in the "1998 DARPA Intrusion Detection Evaluation Data Sets," "1999 DARPA Intrusion Detection Evaluation Data Sets," and "2000 DARPA Intrusion Detection Scenario-Specific Datasets," which were all evaluated in 2000. The original DARPA dataset was compiled in 1998, and there have been only minor updates since then. The DARPA Intrusion Detection dataset has been the most popular in academic research since 1999. The DARPA 1999 dataset was divided into two portions for evaluation purposes: online and offline

evaluation. Depending on the kind of evaluation, IDS assessments were done in real-time, offline, or both. DARPA 1999 resulted in the capture of five weeks' worth of data. This collection includes both training and testing data.

2.6.7 CSE-CIC-IDS2018 dataset

This dataset has been authorised by the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity for use in research projects (CIC) [97]. Heartbleed, Brute-force, Distributed Denial of Service, Distributed Denial of Service (DDoS), Web Assaults, Botnet, and Penetration are included in the CSE-CIC-IDS2018 collection. Similar to the CICDS2017 dataset, we use a program called CIC Flow Meter to extract 80 network flow metrics from simulated network traffic. These measures are done to get the desired outcomes. The CSE-CIC-IDS2018 dataset is an example of a dataset constructed utilising profiles in a systematic manner. This data collection will give invasion descriptions and distribution models for applications, protocols, and lower-level network components that are grounded in reality.

As an added bonus, this dataset's profiles will be utilised as one of the datasets for our investigation. These qualities may be used by human operators or automated agents to expedite the development of network events. It has been shown that the produced profiles are compatible with a variety of network protocols and topologies. Because these profiles are so abstract, they cannot be used in practical situations. The incorporation of various profiles may enable the construction of a dataset tailored to a particular need. At minimum, one or more instances may benefit from this.

B-profiles leverage a number of machine learning and statistical analysis techniques to summarise user entity behaviour (such as K-Means, Random Forest, SVM, and J48). Encapsulating qualities include payload pattern uniqueness, payload size, request time distribution, and protocol packet size. Another element of encapsulation is the total number of packets for each flow. Within the constraints of our testbed environment, we shall do the following activities consistently: HTTPS, HTTP, SMTP, POP3, IMAP, SSH, and FTP are among the access protocols. According to the available statistics, HTTP and HTTPS seem to account for the great majority of traffic. Even if HTTP came before HTTPS, this is still true.

An assault scenario should be described in as much detail as feasible inside the M-Profiles. People may read these profiles and act in a manner consistent with what they have learnt even in the most basic of contexts. If these hypothetical circumstances were to arise, they would be grasped and executed with the aid of self-governing agents and compilers. We explored six different assault scenarios:
BRUTEFORCE ATTACK
DOS ATTACK
WEB ATTACK
INFILTRATION ATTACK
BOTNET ATTACK
DDOS + PORTSCAN

It cannot be emphasised enough that a profile will fail if it is not supported by the appropriate infrastructure. Consequently, a profile is ineffective for this purpose. Our testing environment will consist of a network of Windows and Linux desktop computers. On Windows PCs, we'll utilise a variety of service packs (each with its own unique collection of known vulnerabilities), and on Linux PCs, we'll use a version of Linux that can be exploited using Metasploit (because it was designed to be attacked by new penetration testers).

2.6.8 KYOTO dataset

In order to collect this dataset, we used honeypots, darknet sensors, an email server, and a web crawler, and the data spans a period of three years and comprises actual traffic data. The data was collected during this timeframe [98]. 14 of Kyoto's 24 statistical characteristics were taken from the KDD Cup 99 dataset, while the remaining 10 were gathered from other sources. Because honeypots were used to construct this dataset, no human labelling nor any other kind of anonymisation was employed. While it provides some insight into network activity, honeypots are the only targets it displays. This is the situation because hackers purposefully target honeypots.

In comparison to previously accessible datasets, it provides eleven additional elements that are essential for Network Intrusion Detection System research and assessment. In addition, it contains 10 more characteristics than the next largest public dataset. Due to the continuous repetition of normal traffic between attacks, only DNS and email traffic data are collected. Consequently, only two kind of traffic statistics will be accessible. There are no false positives, since these numbers do not reflect typical Internet use. False positives are advantageous since they reduce the amount of false alarms.

In addition to the fourteen statistical characteristics obtained from the KDD Cup '99 dataset, this study provides 10 additional IDS network research and evaluation-relevant properties. Using honeypots, darknet sensors, an email server, and a web crawler, we acquired data from Kyoto 2006+ with the following characteristics: Singh and coworkers (2015), pages 8609 to 8624. He had songs and friends throughout this time period. Honeypots, which are used to identify infiltration attempts in computer networks, and darknet data gathered on a range of physical and digital devices were the target of in-depth research. Honeypots and acquired data were evaluated. They placed honeypots on five distinct networks, some within and others outside of Kyoto University, and collected information on every incoming and outgoing traffic. These networks used a variety of technologies, such as darknets, honeypots, and others. There were 50,033,015 normal sessions, 43,043,225 attack sessions, and 425,719 unknown attack sessions throughout the monitoring period. There were 50,000,000 completed sessions. DDoS assaults and spam emails are two examples of increasingly hazardous and major large-scale cyberattacks facilitated by botnets.

We have depended heavily on network-based security solutions like Network Intrusion Detection System, Intrusion Protection System, and firewalls to prevent unauthorised access to our most vital computer systems, networks, and sensitive data. Because these components are essential to our organisation, they must be treated accordingly. In recent years, Network Intrusion Detection System,

data mining, and machine learning approaches have all gained a great deal of interest due to the correlation between reward and performance. However, the KDD Cup 99' dataset, which is used to gauge network security, has a fatal defect that it does not reflect the current network health or the implemented attack techniques. This is because stats for the KDD Cup 99' were collected before the competition.

Given that its origins were copied on a digital network more than ten years ago, its peculiar mode of operation is not surprising. This is because it was made in a digital environment. There may be further evaluation datasets available, but we are currently unaware of them. As part of our research, we provide Kyoto 2006+, a new assessment dataset comprised of three years of actual traffic data collected from a variety of honeypots (November 2006 to August 2009). The Kyoto 2006+ data collection will let IDS researchers undertake more realistic, relevant, and reliable assessments. Consequently, this is one of the reasons why these experts consider the dataset crucial. In addition, by disseminating the outcomes of our massive honeypot data analysis, we educate security professionals on the most recent online threats and conditions. As a consequence, we will be able to properly train our security employees on evolving cyber risks and operational circumstances.

2.6.9 ISCX dataset

The dataset was developed by Shiravi et al. [99] and focuses on network activity during the preceding week (normal and malicious). Examples of hostile network activity include brute-force login attempts to SSH, HTTP denial of service assaults, distributed denial of service attacks, and network penetration from the inside. Users are able to peruse and choose from two distinct sorts of profiles inside the ISCX dataset. The first kind consists of mathematical distributions or behaviours derived from data. This behaviour and distribution has been modelled analytically. The second kind of profile seeks to explain an attack situation in an intelligible and simple manner.

ISCX has established a methodical plan for the development of the necessary datasets in order to satisfy the requirements of this demand. The method is based on profiles, which not only contain explicit descriptions of intrusions but also abstract distribution models for lower-level network components such as programs and protocols. The profiles are the primary and secondary components of the approach. Agents that generate authentic traffic for protocols such as HTTP, SMTP, SSH, IMAP, POP3, and FTP are profiled using authentic traces. In this fictitious situation, a set of criteria is developed to characterise legal datasets, and the resulting rules serve as the foundation for the generation of profiles. The efficacy of the dataset's realism, evaluation capabilities, total capture, completeness, and malevolent behaviour depends on the achievement of four essential features.

The following are the characteristics of the UNB ISCX 2012 Intrusion Detection Evaluation Data Set:

Realistic network and traffic: In an ideal world, a dataset would contain no undesirable network activity or traffic volume characteristics. This is done so that the real consequences of network attacks and

the responses of workstations to such attacks may be shown with more accuracy. In order to do this, it is vital that the traffic seem and act in accordance with reality. In this sense, ”transportation” refers to both conventional and unconventional ways of movement. After-collection modifications to the raw data, such as the inclusion of false traces, may generate discrepancies into the final dataset. This is because raw data has not been pre-processed or filtered. Therefore, such modifications are absolutely prohibited.

Labeled dataset: To properly evaluate various detection methods, a tagged dataset is required. In order to avoid time-consuming and inefficient manual labelling, it is preferable to construct a dataset in a controlled and predictable environment that facilitates the separation of abnormal behaviour from normal traffic. As a consequence, it is preferable to construct a dataset in a controlled and predictable environment, as this permits the identification of outliers and their subsequent separation from normal traffic.

Total interaction capture : The amount of data that detection systems have access to is a crucial aspect in their capacity to recognise anomalous patterns of behaviour. In a sense, both post-evaluation and proper data interpretation need these particulars. Therefore, all network interactions, both inside and outside of internal LANs, must be included in the dataset. Despite the fact that some of these conversations may occur across distinct local area networks (LANs), this still remains true.

Capture in its entirety: Researchers in network security encounter a number of hurdles, not the least of which is the difficulty of getting acceptable network traces due to privacy concerns. Researchers have also faced new obstacles. When it comes to providing this kind of data, data providers are often reticent. The majority of these traces are either utilised internally, limiting the capacity of other researchers to correctly analyse and compare their systems, or they are completely anonymised, removing any possible use to researchers. These two approaches make it more difficult for academics to perform fair comparisons and assessments of their own systems. These two techniques make it more difficult for other researchers to undertake exhaustive examinations of the systems they are examining. This investigation’s primary purpose is to create network traces inside the boundaries of a controlled testbed environment without contributing to data cleansing. Thus, the original dataset’s integrity is kept in its entirety.

Different types of intrusion scenarios: In the last many years, assaults have risen in frequency, intensity, diversity, and sophistication. This uniformity is seen in all four of these classes. Complex techniques, such as assaults on targeted services and applications, have been added to the list of possible risks. You’ll need a better knowledge of IP services and applications to identify these assaults, which may cause considerably more severe disruptions than brute-force attacks. The objective is to launch a variety of complex, multistage attacks targeted at resolving the most urgent security challenges of the day. This will be achieved by acting out different assault scenarios and displaying unusual behaviour. According to the majority, the vast majority of publicly accessible datasets are either worthless or

insufficient for evaluating study results. These individuals have internet access to the databanks.

2.6.10 UNSW-NB15 dataset

This data collection was compiled with the assistance of the programs: IXIA Perfect Storm, Tcp dump, Argus, and Bro-IDS. These may be used to create malicious software, such as DoS attacks, Exploits, Generic Assaults, Reconnaissance Attacks, Shellcode, and Worms. The UNSW-NB15 dataset is rather large, including nearly 2 million vectors and 544k features. The dataset supplied by Moustafa et al. was divided into a training set and a testing set that remained distinct throughout the experiment (175,341 vectors each) (175,341 each vector).

The word vector appears 82.332 times in the dictionary. Tcpdump is a program with a one hundred GB capacity that captures the raw data flow (e.g., Pcap files). In this area, Ffuzzers, analysis, backdoors, DoS, exploits, generics, reconnaissance, shellcode, worms, and other forms of assaults are prevalent. Also included in this category are fuzzers, analysis, and backdoors. The software programs Bro-IDS and Argus are used to develop the one hundred twelve processes and forty-nine attributes associated with the label. Before trying to evaluate the anomaly detection system of a network, a sufficient dataset must be collected. Here are some details on the UNSW-NB15. The CSV file gives further information on these characteristics.

The needed information is available in four distinct comma-separated values (.csv) files entitled UNSW-NB152.csv,.csv,.csv, and.csv, respectively. These folders contain a total of 2,540,044 records. The file names are UNSW-NB151.csv, UNSW-NB152.csv, UNSW-NB153.csv, and UNSW-NB154.csv. Each and every file is separated by commas. The file UNSW-NB15GT.csv is the industry standard for the ground truth table, whereas the file UNSW-NB15LIST EVENTS.csv is used for the list of events. This dataset allows us to generate two sets: UNSWNB15training-set.csv and UNSWNB15testing-set.csv. While the training set has 175,341 records, the testing set contains just 82,332 records. These statistics contain both offensive and defensive data. The suggested study utilises Anaconda 3, which has the Python 3 distribution, and Jupyter notebook, which contains the sklearnkit module for Python Machine Learning, to eliminate unnecessary features and reduce the dimensionality of the dataset. In its substitute, the Random Forest Classifier, an ensemble classification approach, has been proposed. Ensemble classifiers, or ensemble classifiers, are more accurate than individual classifiers. Ensemble classifiers predict about a target by combining the findings of numerous separate classifiers. To develop classifiers for categorising data, decision trees are created at random. As a final target prediction, the majority-voting decision tree is applied. The predicted target class will be based on the target class that obtained the most votes, all other criteria being equal.

The IXIA Perfect Storm, TCP dump, Argus, and Bro-IDS programs were used to create the dataset. Denial-of-service attacks, exploits, generic attacks, reconnaissance, shellcode, and worms have all used these techniques. The UNSW-NB15 dataset [100] contains information on 49 different kinds of assaults. In addition, Moustafa and his colleagues authored and distributed a piece that was available to everyone. This dataset contains a total of 2,540,044 vectors; 175,341 are used for training and 540,044

are used for testing (82,332 vectors).

After Argus and Bro-IDS analysed raw network packets, 49 characteristics were discovered in total. They feature both packet-based and flow-based communication capabilities. Applications as Argus and Bro-IDS were used. This information is deduced from the facts at hand. For accurate classification, the header and payload of a packet must be evaluated (also called packet data). By sequencing packets as they move from source to destination over a network, flows with unique features may be generated. While the data packets travel from their source to the destination, something occurs. This occurs whenever data is transferred from one place to another.

The operation is executed at each network node. The formulation of flow-based characteristics prioritises the direction, the inter-packet length, and the inter-arrival duration. Two properties of a flow are time-to-live (ttl), which is given as a number of seconds from terminus to origin, and total duration (dur). Additionally, it is possible to reduce the time by substituting ”dur” for ”duration” (dttl) (dttl). Personality characteristics may be divided into three time periods: before the age of 19, after the age of 26, and throughout adulthood (27 to 35). This runs from 27 and 35. Features 36–40 are seen as having a wide use, whereas features 41–47 are regarded as connection features.

General-purpose characteristics contain features that are intended to communicate the function of a single record, whereas connection characteristics include aspects that illustrate the characteristic of a link between one hundred sequentially organised records. These are two excellent examples of ”general purpose traits.” In Parts 2 and 3, we will study the many sorts of assaults and the different ways they might be classified. Analysis, backdoors, denial of service, exploits, fuzzers, generic attacks, reconnaissance, shellcode, and worms are prominent tactics used by attackers. The three cornerstones of harmful cyber activity are generic attacks, surveillance, and shellcode. Signatures for fuzzers, analysis, backdoors, denial of service, exploits, generic code, reconnaissance, shellcode, and worms are found in the record ranges 24246, 2677, 2329, 16535, 44525, 215481, 13987, 1511, and 174, respectively. Eighty-seven percent is categorised as Regular, whilst less than one percent is categorised as Worms. This shows that the component elements of the dataset are significantly unbalanced.

To effectively utilise a dataset in the development of a data-driven classifier model, like the one used for intrusion detection, one must first be able to see the data structures inside it. This is the case since the data structures are already present in the dataset. It provides visual representations of complex data, succinct explanations of crucial characteristics, and instructive recommendations for developing an advanced data-analysis model.

2.6.11 TWENTE dataset

There were 14.2 million flows and 7.6 million warnings registered during the course of six days. The TWENTE dataset displays a division between ICMP, TCP, and UDP, which all belong to the Internet Protocol (IP) protocol family.

2.6.12 CDX dataset

This data collection was produced by Homoliak and coworkers for a network warfare competition, and it includes both malicious and normal TCP traffic on network services. The competition's focus was on network warfare. The dataset was produced by Homoliak et al. These services are vulnerable to a vulnerability known as a buffer overflow vulnerability. CDX 2009 is capable of compromising four distinct types of servers. OpenFire Chat FreeBSD, Postfix Email FreeBSD, Apache Web Server Fedora 10, and BIND DNS FreeBSD are all examples of such programs. Advanced Security Network Metrics CDX-2009 is a subset of the Comprehensive Distributed Computing Environment (CDX) 2009 dataset, which consists of network traffic dumps. This dataset consists of ASNM properties collected from tcpdump records of both malicious and benign TCP connections to network services vulnerable to buffer overflow attacks. These snatches were generated using network services that are susceptible to buffer overflow. The completed dataset may be categorised as follows:

2.6.13 CDX 2009 Dataset Filtering

The objective of network warfare competition was to create a tagged dataset, which was accomplished via the establishment of the CDX 2009 dataset. This was one of the activities we had planned for the meeting. This link will direct you to the CDX-2009 dataset. We examined data from outside the West Point network in addition to NSA data (NSA). With the assistance of SNORT's logs and the boundaries it produces, ASNM data is obtained. The format of these packet captures and segmentation points are Tcpdump (as a source of ground truth).

2.6.14 ADFA2013 dataset

Creech and Hu provide a dataset including payloads and vectors to facilitate the creation and execution of attacks against Ubuntu. A brute-force password attack, a new superuser installation, a Java-based meterpreter, a Linux-based meterpreter payload, and a C100Webshell have been detected. The main subcategories of the dataset are normal training data, normal validation data, and attack data. Training data typically consists of 4,373 traces. The validation data by default comprises 833 traces. There are ten distinct vectors contained in the attack data.

VICTORIA UNIVERSITY

THE NEW WAY TO DO UNI

OFFICE FOR RESEARCH TRAINING, QUALITY AND INTEGRITY

DECLARATION OF CO-AUTHORSHIP AND CO-CONTRIBUTION: PAPERS INCORPORATED IN THESIS

This declaration is to be completed for each conjointly authored publication and placed at the beginning of the thesis chapter in which the publication appears.

1. PUBLICATION DETAILS (to be completed by the candidate)

Title of Paper/Journal/Book:	Movie Recommendation System Using Deep Learning
Surname:	Sama
First name:	Lakshit
Institute:	Institute for Sustainable Industries and Liveab
Candidate's Contribution (%):	70

Status:

Accepted and in press:	☐	Date:	
Published:	☑	Date:	21-01-2022

2. CANDIDATE DECLARATION

I declare that the publication above meets the requirements to be included in the thesis as outlined in the HDR Policy and related Procedures – policy.vu.edu.au.

[redacted]	05/08/2022
Signature	**Date**

3. CO-AUTHOR(S) DECLARATION

In the case of the above publication, the following authors contributed to the work as follows:

The undersigned certify that:

1. They meet criteria for authorship in that they have participated in the conception, execution or interpretation of at least that part of the publication in their field of expertise;
2. They take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;

3. There are no other authors of the publication according to these criteria;
4. Potential conflicts of interest have been disclosed to a) granting bodies, b) the editor or publisher of journals or other publications, and c) the head of the responsible academic unit; and
5. The original data will be held for at least five years from the date indicated below and is stored at the following **location**(s):

Name(s) of Co-Author(s)	**Contribution (%)**	**Nature of Contribution**	**Signature**	**Date**
Lakshit Sama	70	Formal analysis, Methodology, Writing	[redacted]	05/08/2022
Hua Wang	15	Resources, Supervision		05/08/22
Aaisha Makkar	15	Validation, Review		05/08/22

Updated: September 2019

Chapter 3

Role of Deep Learning in Network Intrusion

Cyberattacks are increasingly targeting critical national infrastructures (CNI) such as airports, power and water distribution networks, schools, and energy suppliers. CNIs rely on Supervisory control and data acquisition systems (SCADA) or, more broadly, industrial control systems (ICS) for their production management.

Appropriate management of ICSs and CNIs has become a critical issue, and its resolution requires action at the organisational and national levels. Consider the European Union's recent approval of a slew of rules and regulations in response to the growing threat posed by computer network intrusions (CNIs), all of which are aimed at building a uniform framework for network, information, and electronic communications security. To address all elements of cybersecurity, including legal and organisational factors, capacity-building and technological components in addition to enacting legislation and implementing directives and policies, specific security measures must be adopted.

Intrusion detection systems (IDS) [75] serve as an additional line of defence against unauthorised access to a computer system or network. Intrusion detection systems (IDSs) used in conjunction with other security measures such as access control, authentication procedures, and encryption methods may assist in further protecting systems from cyberattacks. We can detect possible vulnerabilities by evaluating traffic patterns and restrictions that are relevant to a certain attack. According to Dewa and Maglaras, traditional intrusion detection systems (IDS) may not be as effective as data mining in dealing with today's high-tech cyber threats. They say this is because data mining can be used to describe how knowledge is found [101].

Several intrusion detection systems are created in the literature using numerous machine learning approaches. Some research, for instance, use single learning approaches, including neural networks, genetic programming, support vector machines [102–105]. On the other hand, several systems are built on the combination of various learning approaches, such as hybrid or ensemble techniques. Specifically, these approaches are designed as classifiers, which are used to categorise or detect whether an incoming Internet connection is a regular connection or an attack. These categories are discussed in Figure 3.1.

1. **Single classifiers**: One machine learning method may be used to solve the intrusion detection

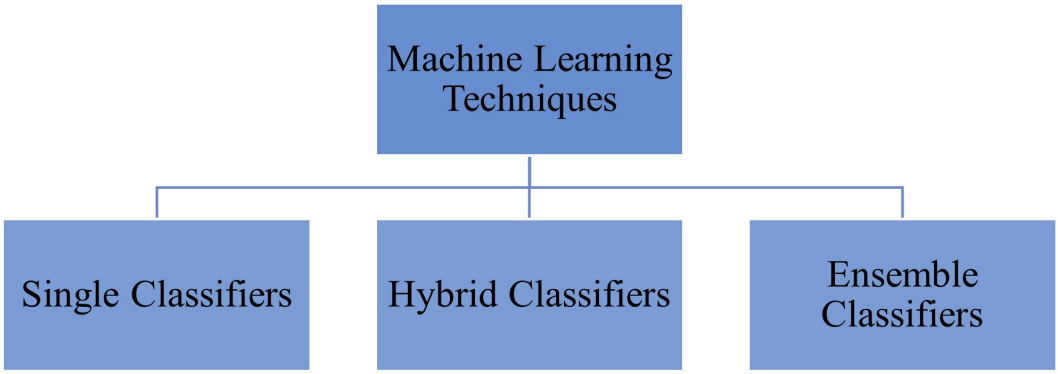

Figure 3.1: Machine Learning techniques for Intrusion Detection

issue. In the literature, these issues have been resolved using machine learning methods as k-nearest neighbour, support vector machines, and many more [106–108].

2. **Hybrid Classifiers**: The ultimate objective in creating an IDS is to obtain the highest level of accuracy for the job at hand. This goal always results in the creation of hybrid approaches to the issue at hand. A hybrid classifier combines many machine learning methods in an effort to dramatically enhance system performance. This method often has two functional components, to be more precise. The first produces intermediate outcomes using raw data as its input. The second will then generate the final results using the intermediate results as input.

3. **Ensemble Classifiers**: To enhance the classification performance of a single classifier, ensemble classifiers were introduced [109, 110]. The combination of several weak learning algorithms or weak learners is referred to as a ”ensemble.” Different training samples are used to instruct the weaker students so that the total performance may be successfully enhanced.
   The ”majority vote” is likely the most often used method of grouping poor learners in the literature. Other combination techniques, like boosting and bagging, rely on resampling training data and then casting a majority vote among the weak learners that arise.

Machine Learning method requires comparative data samples, to create an efficient and successful ML model. Data collection will usually be accomplished either in offline mode or online mode. Data collection enables a vast amount of background records to be obtained and use them for model testing and training. Whereas, in the online process, real-time network collected data can be used as sample to again train the model. Offline information is gathered from different collection, provided that the sample is suitable to the concern problem. The examples of such repositories include UCI Knowledge Discovery, Waikato Internet Traffic Storage (WITS) and many more. Using tracking and analysis software is an efficient way to gather both online and offline data. Monitoring can, however, be passive

or active. Active tracking transforms the traffic measurements, and extracts specific features of the data from this traffic. Passive tracking, by comparison, gathers data by monitoring the real traffic on the network. Evidently, because of bandwidth demand from injected traffic, active control creates excessive overhead. Although passive control minimizes this overhead, also at cost of longer facilities which examine network activity to collect necessary details.

It needs to be broken down into after data cleaning, i.e., training, validation and test datasets once data is collected. The training dataset is used to evaluate the optimal hyper-parameters like Neurons of an ML model. The sample collection, however, is used to select the required architecture a model. Note that if an ML model is fixed with its design in the primarily, then the requirement of further package decreases.

K-fold cross-validation approach can be used to conduct validation and checking. Portion of the usable dataset is set aside in the holdout technique and used as a verification (or testing) set. This prevents the outcome from becoming learned and generalized, resulting to by over and/or under-fitting designs.

Intrusion detection systems are being investigated by researchers to see whether deep learning can be employed in their design or not. Any new work that employs deep learning to detect intrusions would be welcomed, given that the research relied on deep learning to find intrusions in the first place. The literature has a large number of various kinds of deep learning models, and many of these models are now being employed in a variety of modern systems. More in-depth information on the issue will be provided in the next section.

1. Deep neural network
2. Recurrent neural network
3. Convolutional neural network
4. Deep Auto-encoder
5. Restricted Boltzmann Machine
6. Deep Belief Network
7. Miscellaneous
    a) Feedforward neural network
    b) Replicator Neural Network
    c) Deep migration learning
    d) Self-Taught Learning

3.1 Deep Neural Networks

Tang et al. [111] created a deep learning-based intrusion detection system for software-defined networking based on software-defined architecture. The recommended intrusion detection solution is included in the SDN controller, which is in charge of monitoring all OpenFlow switches. The data was divided into four categories: probing attacks, denial-of-service attacks, user-to-resource (U2R) attacks, and remote-code-execution (R2L) attacks. The research found that learning rates of 0.001 outperformed those with the highest receiver operating characteristic (roc) (AUC). To deal with huge amounts of network data, Potluri et al. [112] used a deep neural network approach as the deep-category classifier. They used the NSL-KDD dataset, which has 39 attack types divided into four groups. Their results show that when there are just two classes, detection accuracy is excellent (normal and assault). Kang et al. [113] proposed a deep neural network-based intrusion detection solution for vehicular networks. As part of the attack scenario, malicious data packets were injected into an in-vehicle controller area network. To categorise packets into two groups, the proposed system distributes the feature vector to the input nodes (i.e., a normal packet and an attack packet). The activation function is used to calculate the outputs (e.g., ReLU). The results are then connected to the invisible underlying layers. The suggested approach achieves a detection ratio of 95% when the false positive rate is less than 1–2 %. To enhance the detection of cyber-attacks, Zhou et al. [114] introduced an intrusion detection system based on deep neural networks.

Deep neural network classification is one of the system's three components, which also comprise data collection. When trained, the network's detection rate was 0.01 and the number of input sub units was 86. The network was retrained ten times. In terms of performance, k-nearest neighbour, random forest, and linear regression aren't even close. The authors [115] propose a plug-and-play device for ad hoc networks that uses a capture tool to collect packets and a deep learning detection model to detect network denial-of-service (DoS) and privacy assaults. Deep learning technologies such as convolutional neural networks and long short-term memory are employed in the proposed model to detect cross-site scripting and SQL injection attacks. The suggested method uses a deep neural network to detect denial-of-service threats. The KDD CUP 99 dataset was used in the study, which was separated into two parts: 30% for testing and 70% for training. Using deep neural networks, the study found that the detection rate for XSS attacks was 0.57% and 0.78%.

Zhang et al. [116] employed sophisticated adversarial learning and statistical learning methodologies to identify network intrusions. The research detects a wide range of network attacks by combining data augmentation with complicated classification methods. A discriminator and a generator are important system components. When augmented data from genuine intrusion samples is augmented, the discriminator rejects enhanced data obtained from real intrusion samples.

Kim et al. [117] used the KDD 1999 data set to build a deep neural network for ever-evolving network assaults, which they then deployed to the KDD 1999 data set. The suggested intrusion detection model uses just two parameters: four hidden layers and 100 hidden units, both of which are concealed from view. For deep neural network training, the ReLU function is used as the activation function, and

the training process is optimised using the stochastic optimization technique. The suggested model, according to the authors, has an accuracy of roughly 99% authors. The decision of almost 99%. Zhang et al. [118] developed the CAN IDS, a two-stage intrusion detection system for autonomous vehicle detection. The first step employs a robust rule-based approach, while the second stage employs a deep learning network to identify outliers. The performance of the evaluation is determined using three datasets: Honda Accord, Asian brand, and US brand automobiles. While the training data is exclusively comprised of normal traffic from these three datasets, the testing data includes both normal and malicious traffic that has been exposed to five distinct attack types: drop attack, zero-ID message assault, replay attack, and spoofing attack.

3.2 Recurrent Neural Networks

Kim et al. [119] presents an intrusion detection system based on the KDD Cup 1999 dataset that includes a large short term memory architecture in a recurrent neural network. The technique is validated on the KDD Cup 1999 data set. The experiment used an input vector of (41 characteristics) and an output vector of (4 attacks and 1 non-attack). The time step was 100 milliseconds long, the batch was 50 milliseconds long, and the epoch was 500 milliseconds long. According to the manufacturer, the attack detection performance is 98.8% across all assault scenarios.

Loukas et al. [120] are examining the deployment of a cyber-physical intrusion detection system to detect cyberattacks on autos. This method combines deep multilayer perceptron architectures with recurrent neural networks to achieve more accuracy and consistency than traditional machine learning techniques (e.g., k-means clustering and SVM). To evaluate the system's resilience to risks posed by robotic vehicles in general, command injection, denial of service attacks, and malware aimed at the network interface are all performed. Taylor et al. [121] developed a prototype of a vehicle attack detection system based on an artificial recurrent neural network architecture. The Long Short-Term Memory (LSTM) is a recurrent neural network that has been trained to predict future packet data values and to identify abnormalities utilising its failures as a signal.

Yin et al. [122] used a convolutional neural network to augment a recurrent neural network for automated categorization learning. Three performance metrics (time to first failure, first failure rate, and accuracy) were calculated in the research using the NSL-KDD dataset. According to the authors, anomaly detection is more effective when a learning rate of 0.1 is used in conjunction with 80 hidden nodes. The benefits of utilising a recurrent neural network as an intrusion detection system are also fully covered in this article (ids). Tang et al. [111] suggested the use of a rectified linear unit run for intrusion detection in application networking. Other researchers concurred. According to the study, an 89% recognition rate is feasible with only a few criteria. The network's efficiency must be evaluated using the NSL-KDD dataset and four evaluation metrics: recall, F-measurement, precision, and accuracy. Jiang et al. proposes a multichannel intrusion detection system based on long short term memory in their study. The performance of the proposed detection and prevention system is evaluated using the NSLKDD dataset. The long-short-term memory recurrent neural net has a detection rate of

98.94 %, a false alarm rate of 9.86 %, and an accuracy of 99.94 %. It has a detection rate of 98.94 % and a false alarm rate of 99.23 %.

3.3 Convolutional Neural Networks

Basumallik et al. [123] applied convolutional neural networks to identify packet-data anomalies in a phasor measurement unit-based state estimator. Their research was published in the journal Science. They use a convolutional neural network-based data filter to extract event signatures (features) of phasor measurement units from input data. To communicate between data-gathering equipment, the IEEE-30 and IEEE118 bus systems are employed.
If you have 512 neurons in one layer that are all linked together, you can get a chance of 0.5% and 98% accuracy, according to the research that was done. They say that a completely convolutional network-based filter is better than other machine learning methods.

According to Fu et al. [124], a system that uses a deep neural network to capture the most powerful fraud behaviours, namely in the detection of credit card fraud, has been established. Zhang et al. [125] used a neural network and B2C online transaction data from financial institutions to train and test their solution. A month's worth of data was divided into two groups: training and test sets. According to the research, the accuracy rate is 91% and the recall rate is 94%. When compared to Fu et al.'s [124] research, these findings show a 26 % and 2 % improvement, respectively, over the previous work.

DeepCorr, an intrusion detection system based on a convolutional neural network, was developed by Nasr and colleagues [126]. The system develops a correlation function, which is then used to identify attempted intrusions. DeepCorr is made up of three layers of fully connected neural networks, two layers of convolutional neural networks, and one layer of fully connected neural networks. In investigations, DeepCorr was shown to have a true positive rate of about 0.80, a learning rate of 0.0001, and a false positive rate of 103.
Zhang et al. [122] developed an anomalous traffic detection model that includes two layers of neural networks, the first of which is the updated LetNet-5 convolutional neural network. The second layer makes use of both long-term and short-term memory. The first layer is meant to recover the flow's spatial qualities, while the second layer is intended to recover the flow's temporal properties. On the CICIDS2017 dataset, performance was 94% better than previously. In terms of accuracy, precision, recall, and F1-measure, the proposed technique beats existing machine learning approaches. As a consequence, Zeng et al. [127] provides deep-full range detection, as well as a lightweight framework for uncovering new threats, categorising encrypted data, and detecting intrusions (DFR). Yu et al. [128] used a convolutional autoencoder to analyse network assaults on two intrusion detection datasets, the CTU-UNB dataset and the Contagio-CTU-UNB dataset, to better understand network attacks. The Theano tool is used to design a neural network model. The learning rates for the pretraining and fine-tuning approaches are 0.001 and 0.1, respectively, for each phase. The Contagio-CTUUNB data set is used to divide classes into six and eight categories. An average of 0.99 can be seen on the classification ROC curves for six and eight classes. In addition, the research achieves an astounding

99.59 % accuracy rate in binary classification.

3.4 Deep Auto-encoder

Shone and colleagues [129] relied on the deep auto-encoder throughout their investigation to discover cybersecurity issues. In an effort to improve classification performance, a non-symmetrically large number of hidden layers is being used in place of deep belief networks. According to the conclusions of the study based on the NSL-KDD and KDD Cup datasets, performance is measured using KDD Cup 99 accuracy and F-score metrics. In comparison to prior research, this model earned an average accuracy of 97.85 % in the KDD Cup '99 dataset assessment, which is higher than the previous study's accuracy. When tested against the NSL-KDD dataset, the researchers discovered that the proposed model had an accuracy rate of 85.42 %, a 5% improvement over the deep belief network model.
Khan et al. [130] created a two-stage deep learning-based intrusion detection system that makes use of both deep learning models. Along with a soft-max classifier and a three-layer stacked auto-encoder, this model includes various more components. This model employs a feed-forward neural network, which is comparable to a multilayer perceptron. The researchers conducted their study using publicly available datasets, KDD99 and UNSW-NB15. According to the researchers, the KDD99 dataset generates conclusions with high recognition rates of up to 99.996 %. With a recognition rate of up to 89.134 %, the UNSW-NB15 dataset is suitable for pattern discovery.
Papamartzivanos et al. [131] propose a machine and autonomous abuse intrusion detection system for usage in healthcare settings. According to the proposed system, the four stages are as follows: monitor; analyse; plan; execute, and gather knowledge. This phase involves monitoring for changes that may need the adaption of an intrusion detection system. Using network audit tools, the raw network traffic is turned into network flows, which are then further analysed (e.g., Argus and CICFlowMeter). A sparse autoencoder is utilised during the planning phase to ensure the required quality of input data. The execute phase is in charge of saving all collected data. Meanwhile, performance is examined using two datasets: the KDDCup'99 and the NSL-KDD. According to data, the static model's average accuracy is 59.71%, while the adaptive model's average accuracy is 77.99%.
Yang et al. [132] applied an updated conditional variation auto-encoder and deep neural network for cybersecurity. Three steps are included in the proposed study: training, designing new attack, and detecting assaults. Throughout the training phase, performance losses in the encoder and decoder are minimised to the greatest extent feasible. We use the same multivariate Gaussian distribution as before for the production of fresh attacks.
During the detection phase, a deep neural network is employed to identify potential assaults. Validation of the proposed model is carried out using the NSL-KDD and UNSW-NB15 datasets, with the Adam optimizer's default learning rate of 0.001 being employed. The researchers discovered that the highest accuracy was 89.08 per cent and the detection rate was 95.68per cent on the UNSW-NB15 dataset.

3.5 Restricted Boltzmann Machine

Fiore and colleagues [133] used the Boltzmann machine's limited capabilities to detect intrusions. They achieve the needed integration by combining the expressive capability of probabilistic models with the efficacy of classification using a discriminative restricted Boltzmann machine. This analysis was conducted using the KDD Cup 1999 dataset. In all, 97,278 instances with 41 distinct characteristics were discovered.

Salama and colleagues [134] identify intrusions in their system using the limited Boltzmann machine and the support vector machine. There are a total of 22 different attack types in the training set, whereas there are 17 distinct attack types in the testing set. The study's findings indicate that this combination beats support vector machines in terms of classification accuracy.

Alrawashdeh and Purdy [7] used the limited Boltzmann machine in conjunction with a deep belief network to analyse the large KDD 1999 data set. C++ and Microsoft Visual Studio 2013 were used to develop the detecting algorithm. Following the research, it was determined that a restricted Boltzmann machine could identify 92% of attacks correctly. In comparison to Salama et al. [134], the outcomes of this investigation were positive. They demonstrated an improvement in both detection accuracy and detection speed.

Aldwairi et al. [135] conducted comparative research of restricted Boltzmann machines in order to discover cybersecurity breaches. When limited, Boltzmann machines are proven to be capable of discriminating between normal and aberrant NetFlow traffic in a specific context. When the training data was changed to a precision of 0.004, the proposed research was evaluated on the ISCX dataset, and the precision was determined to be 78.71.9 percent. Both true positive and true negative outcomes exhibit high true positive and true negative rates of 74.9 4.6per cent and 82.4 1.8per cent, respectively, with a learning rate of 0.004.

Gao et al. [136] attempted to combine multilayer unsupervised learning networks to enhance intrusion detection. After learning a limited Boltzmann machine with n layers, the whole restricted Boltzmann machine's parameters are used to train the restricted Boltzmann machine. On the KDD CUP 1999 dataset, the paper reveals that a restricted Boltzmann machine-based deep belief network outperforms alternative deep belief networks.

Alom and colleagues [137] propose an intrusion detection system based on a stack-limited Boltzmann machine. The primary objective of this method is to identify instances of anomalous or dangerous behaviour.

The NSL-KDD dataset contains five distinct types of assaults. It has been demonstrated that when the data is used, only 40% of the data that is used in training is required for the system to achieve 97.5 percent accuracy. In this instance, Boltzmann's bare-bones training gear is used. They recommended varying the number of units in the hidden layers throughout the feature extraction phase. After acquiring the desirable characteristics, the authors recommend training a support vector machine model using gradient descent training parameters. SVM-RBMS, a new method, is expected to have an accuracy rate of up to 80%.

Otoum et al. [138] created RBC-IDS, a clustered intrusion detection system for wireless sensor networks based on the restricted Boltzmann machine. The RBC-IDS system is built up of N clusters, each of which has a set of C sensor nodes that are linked together. To evaluate its performance in a network simulator environment, the Network Simulator-3 (NS-3) and KDD Cup 1999 datasets were employed. When compared to the adaptive machine learning-based IDS (ASCH-IDS) [139], the RBC-IDS system achieves 99 percent content accuracy, whereas the ASCH-IDS system achieves 9 percent accuracy with three hidden layers, demonstrating a significant increasing accuracy.
Aloqaily et al. [140] created a deep belief network and decision tree-based intrusion detection system to safeguard the connection element of connected automobiles. The deep belief network is used to reduce data dimensionality, meanwhile , the decision tree is used to classify attacks based on severity. The D3H-IDS system gathers and analyses data via the use of a cluster-head selection mechanism, which is discussed in detail below. The researchers assessed the system using both the NSL-KDD dataset and NS-3-gathered traffic, and the findings indicate that the system has the greatest detection rate when compared to previous work published in [136], which is encouraging. Additional information about the car dataset is available in the paper [141], which we urge the reader read.
Karimipour et al. [142] used a deep unsupervised machine learning approach to develop a network security intrusion detection system for use in large smart grids. Through the use of symbolic dynamic filtering, dynamic Bayesian networks connect smart grids to give a computationally efficient feature extraction method. This approach may be used to derive causal relationships between intelligent grids. The authors advised that patterns in system behaviour be identified using a constrained Boltzmann machine. The results of a cyberattack on an IEEE 39 bus system show that this model is nearly 100% accurate, with a very high true positive rate and almost no false positives.

3.6 Deep Belief Network

Thamilarasu et al. [143] illustrated how deep belief networks may be used to identify intrusions. The feed-forward deep convolutional neural network for the Web of Things is constructed using a deep belief network. To minimise the simulation's overall cost, the authors added a binary cross-entropy loss function to the IDS model. The Keras framework, the Cooja networking simulator, and the Texas Instrument sensors tags CC2650 are used to analyse the performance of the system under test. Using the Keras tools, we can construct a continuous deep-learning model. The proposed technique is evaluated against five different attack vectors: blackholes, sinkholes, opportunistic services, wormholes, and distributed denial of service (DDoS) assaults. When it comes to identifying DDoS assaults, the data suggests a per cent accuracy rate and a 9 percent recall rate.

Zhao et al. [144] designed an IDS architecture combining deep belief networks and probabilistic neural networks. This research evaluates the intrusion detection system using the KDD CUP 1999 dataset, which has a 10% training dataset and a 10% testing dataset. The method outperforms three models in the trial data: a basic statistical neural network, component analysis coupled with a conventional probabilistic neural network, and an optimized deep belief network mixed with a

probabilistic neural network.

According to Zhang et al. [145], their study is a great instance of how a deep belief network combined with an updated evolutionary algorithm may be utilised to uncover cybersecurity vulnerabilities, and they strongly advise you to read it. The research makes use of a variety of restricted Boltzmann machines. Typically, these machines are employed for supervised techniques of pre-processed data, although they may also be utilised for unsupervised methods. The training procedure for the deep belief network module is often separated into two stages: first, each restricted Boltzmann machine is trained individually, and then the last layer of the deep belief network is transformed to a backpropagation neural network. The NSL-KDD dataset was used to evaluate performance, and the dataset exhibited a detection rate of 99 percent.

3.7 Miscellaneous

3.7.1 Feedforward Neural Network

Kasongo and colleagues [146] identified intrusions in a network of computer systems using feedforward deep neural network. Combining an FFDNN with a filter-based feature selection technique is crucial for obtaining acceptable subsets of features for wireless networks with the least amount of repetition feasible. To train the proposed intrusion detection system, the main training dataset was partitioned into two sets: one for each of the suggested intrusion detection systems and another for the proposed intrusion detection system (i.e., the training dataset and the evaluation dataset). As a result of the aforementioned findings, a technique for feature transformation and two-way normalisation is adopted. Finally, the FFDNN is used in the proposed system for model training and assessment.
The KDDTrain+ and KDDTest+ tests from the NSL-KDD dataset were used in this experiment. The performance assessment findings indicate that the suggested system achieves 99.69 percent accuracy with a learning rate of 0.05 and 30 neurons distributed over three hidden layers.

3.7.2 Deep migration learning

He et al. [147] developed an enhanced deep belief network design for detecting bogus data injection assaults in the supervisor control and data collecting system, which was ultimately implemented. This study employs a conditional Gaussian-Bernoulli restricted Boltzmann machine to extract rapid temporal information and hence, minimise processing time. Taylor et al. [148] argues that his technique has the potential to reduce both the complexity of deep learning architecture training and the time needed to construct the designs. According to the performance assessment network, the detection accuracy rate was 98per cent on both the IEEE 118-bus energy test system and the IEEE 300-bus system. The NSL-KDD dataset was utilised to evaluate performance, and it demonstrated a detection rate of 99 percent when the dataset was employed.

3.7.3 Replicator Neural Network

Cordero et al. [149] use replicator neural networks to identify cybersecurity breaches. The research team utilised a method called dropout to identify anomalies. The entropy extraction technique consists of three stages: In the first step, packets are grouped together. The second step divides the flows into time intervals. This is the last step in which you extract interesting characteristics from the flows. The performance assessment is performed using the MAWI dataset, which includes the additional simulated assaults.

3.7.4 Self-Taught Learning

Li et al. [150] used deep migratory learning to detect cyber security breaches in cyberspace. The study team revealed that the investigation was conducted with the assistance of a deep learning model and an intrusion detection system. This study identified four distinct forms of deep migration learning technologies: parameter migration, sample migration, related knowledge transfer, and feature representation transfer. There are four kinds of deep migration learning techniques: parameter migration, sample migration, related knowledge transfer, and feature representation transfer. Deep migration learning methods may be grouped into a variety of categories, each of which can be split further. The experimental data are derived from a subset of the KDD CUP 99 dataset, with 10% of the training set serving as experimental data. Experimental datasets serve as experimental test sets, and training datasets serve as training sets. Throughout the experiment, 10,000 randomly chosen datasets are used as training sets, while 10,000 randomly selected datasets are utilised as experimental test sets. Experimental datasets serve as experimental test sets, and training datasets serve as training sets. Throughout the experiment, 10,000 randomly chosen datasets are used as training sets, while 10,000 randomly selected datasets are utilised as experimental test sets. According to the data gathered, 91.05 percent of incidents were identified, with a false alarm rate of 0.56 percent.

3.8 Other Applications of Deep Learning

3.8.1 Movie Recommendation System Using Deep Learning

Creating a movie recommendation system using artificial intelligence technologies is a difficult task. It's tough to obtain satisfactory results because not all data on the internet is reliable or valuable. To address this issue, experts advocate employing a recommendation system that delivers relevant information rather than repeatedly looking for the same piece of information. By and large, collaborative filtering algorithms that are based on user information (such as gender, geographic location, or preference) are effective. However, in today's world, where everyone is concerned with ideas or guidelines to help them make a decision, our area of research includes a movie recommendation system that allows us to present the user with a list of related movies. Now that the user has been provided with a selection of movies recommended by our model, he or she merely needs to decide which film to

view next. Rather than perusing the entire list, it's preferable if he/she has some recommendations from which to choose. The entire performance of the purposed framework resulted in a 66 percent accuracy rate, which is quite good for such an endeavour.

Introduction

Nowadays, rather of wasting time debating 'what I should watch next,' individuals want to receive recommendations based on their prior viewing histories/list of watched films. By making recommendations, we aid them in locating related films on a priority basis [151]. For them, this saves a significant amount of time that they would have spent choosing alone. To recommend, we consider the fact that some people watched the same movie and then the movies that the rest of the people watched, in order to predict that he/she may be interested in some of the movies that the rest of the people watched after watching his movie; based on this, we provide the user with a list of movies that are similar to the previous one [152, 153]. Deep learning has been a significant method in a variety of domains, including data analysis on websites and in health-related fields [154–156]. We analyse the movie recommendation problem using a deep learning approach. One criterion that is evaluated is the rating; we prioritise and recommend to users the highest rated films [157].

Deep Learning is a component of artificial intelligence that is frequently used to create algorithms based on data trends and historical relationships between data. This article makes the following contributions: 1) A single dataset is preprocessed using the feature engineering approach. 2) Using KNN clustering methods, the suggested scheme will recommend further films to consumers. 3) A deep learning model was used to validate this strategy (Matrix factorization).

Literature review

The traditional linear algebra lies at the heart of machine learning [139, 141, 158]. Numerous machine learning applications are not used in Recommendation because datasets without NaNs are not available (incomplete entries). For instance, when constructing a model, NaN or missing data is removed during data preparation because most algorithms cannot operate on unpopulated values. Without values, features such as principal component analysis cannot be defined. However, removing NaN will cause the recommendation system to fail to function properly. There are valid reasons for these NaNs: not every user rates every item, and expecting them to is a bit illogical. Dealing with sparse data may be fairly challenging, which is why Recommendation is such an intriguing subject [159, 160]. Sparsity further complicates matters. Without defined terms in the matrix, the analysis of single values (that is, the factorization of the m x n matrix's singular and orthogonal values) becomes questionable. This means that we cannot explicitly compute a matrix in order to determine which diagonal (or prospective) elements have the most weight in the data.

Non-Personalized Recommendations are straightforward but quite valuable. We may provide recommendations to the user without knowing anything about them because they resolve the customer's cold start issue [161, 162]. Following the receipt of user input or the acquisition of further information

about the user, We can now move on to some of the more sophisticated models detailed below. The approach supplied by IMDB was used to find the best films in various categories that may be recommended to a new user. This recommender evaluates all users who have viewed a particular film and then counts the number of times the group's most popular film is returned. Without regard for context (and only on the basis of user ratings), we utilise the KNN algorithm to discover related movies based on user ratings for distinct films. The k-nearest neighbours (KNN) technique is a straightforward supervised machine learning algorithm that can be used to address classification and regression forecasting problems. However, it is more likely to be used in industrial categorization problems. When analysing any technique, we often consider three key factors [163]:

1. Simplicity of interpretation of the output

2. Speed of calculation

3. Predictive Power

We attempted matrix factorisations (Collaborative Filtering); because the low-rank method is incredibly inefficient, the sparse matrix was factored using scipy's SVD. NMF is a relatively new technique for discovering linear representations of non-negative data based on its constituents. While this technique has been successfully applied to a wide number of applications, it does not always result in a representation based on parts. The fact that NMF often delivers data in a sparse representation is one of its most attractive qualities.

This format uses a small number of 'active' components to encode a great quantity of data, allowing for easy decoding of the coding. [164] The matrix factorization technique is a well-known technique for constructing recommendation systems. It is predicated on the assumption that we can learn the low-dimensional representations of users and films (embedding). For instance, we can ascertain the required number of actions and the time of each film. We may code the duration of each user's enjoyment of the action, or the duration of their enjoyment of the film, and so on. As a result, we can forecast the ratings of unreleased films using a combination of user ratings and built-in movies. Alternatively, consider the following: We wish to estimate low-dimensional matrices (W [M X k] and H [M X k]) from a matrix (A [M X N]) comprising users and videos.

There is an area that requires extensive learning and merits additional exploration. For the time being, we will examine many major directions in terms of data, objective, methodology, explanation, and security. [165–167]. The recommendation of projects, as well as the recommendation system itself, necessitates a multi-objective evaluation, including accuracy, diversity, and even contingency, as well as the efficiency and quality of online networking services for large-scale projects. As a result, using auxiliary data to construct increasingly intricate goal functions with multiple assessment indicators is extremely stimulating.

Formulation of the Problem

The objective of this project is to develop software that will allow users to receive movie recommendations. People suffer far too much trying to determine what to watch next; it consumes far too much of their time [168]; it will be much easier for them if some recommendation system takes care of this work and leaves them with only a small selection of movies from which to choose their next viewing [169]. Our model will propose films based on their previous interests.

The proposed research aims to conduct work that will result in the development of a method for Deep learning-based prediction The proposed target will be accomplished by segmenting the effort into the following goals:

1. Collect a solid dataset
2. Use deep learning methods to achieve high accuracy/better results
3. Validate scheme using deep learning and cross-check the findings.

Dataset analysed: This notebook makes use of the Movie Lens 1M rating data collection. This section contains 1 million movie ratings from 7120 films and 14,025 users. This data set contains the following [170]:

- Each film is assigned a unique identifier called a movie ID; this identifier can be used to identify any film.
- Additionally, whatever methodology we apply will be based on the movie ID, as the movie ID is the unique identifier for each film.
- Each user has a unique ID, which enables us to refer to users by their IDs. If the same user ID appears in two different test sets, it indicates that both are the same person.
- Rating: A rating is a numerical value assigned to a film out of ten. It is determined by a variety of elements, including user feedback, budget, and profitability of a particular film. We determine whether a film is a smash or a flop at the box office based on its rating.

Additionally, a movie data set comprises the title, tags, and genres of the film [171, 172]. The title is the film's name; in some circumstances, the title may be same, but identification of the film is accomplished by the use of the Movie ID. The title of the film explains a great deal about what the movie is all about [173, 174].

Genres: This term refers to the sort of film, such as romantic, action, drama, thriller, or adventure, etc., that we have to recommend, and thus the genre plays a significant role in the recommendation system.

Tags: Producers and directors can use tags to describe what makes this film unique, for example, any actor's name or genre can be included in tags, and any quote can also be included in tags.

Step-by-step Procedure

- Imported the essential libraries; the library contains preset functions that are critical for the proper implementation of our strategy.
- We need to read the CSV file and evaluate the required field. If a user rates a movie twice or more, we consider the maximum rating. This increases accuracy [175, 176].
- Visualize the data and categorise the ratings based on the statistics contained in the dataset, as well as calculate the proportion of each rating for better comprehension.
- Obtained a genre dataset and converted it to a list of genres so that we can simply construct a matrix factorization approach.
- Consider the concept of a weighted rating score based on the number of reviews def weighted rating(x,m=minimum reviews,C=average rating across all): Apply join() appropriately, for instance, to join movie score and movie rating.
- Obtain the best films according to genre-based on the weighted score generated using the IMDB method, we now have a list of the greatest films in each genre that we can propose to them, which is critical for our recommendation system.
- Find the latent features using low-rank matrix factorization.
- Using pd.merge, I combined the rating and movie data frames ().
- Extracted the top ten films watched by those who saw this film, for example: Obtaining further top ten films that are watched by those who saw 'Gone Girl' obtain other films('Gone Girl (2014)').
- Created a matrix table with rows for movie IDs and columns for user IDs, and replaced NAN values with 0.
- Using 'kNN', determines the top ten nearest neighbours of any movie (k-nearest neighbors).
- Using tags and other features, determines the top ten movies based on their content.
- Determined the rmse score for SVD using various values of k (la-tent features).
- Implemented a matrix factorization scheme that generates recommendations based on the aforementioned characteristics.
- A neural network model capable of performing matrix factorization was returned.
- Analyzed the recommendation and verified its accuracy; accuracy is critical.

Outcome from Experiments

Using deep learning, we can classify movies according to their genre. For example, if we want to extract the greatest movies in a certain genre, we can call best movie by genre('genre', List size) as shown in Figure 3.2.

```
best_movies_by_genre('Musical',10)
```

	title	count	mean	weighted_score
824	Duck Soup (1933)	280	4.217857	4.151220
580	Singin' in the Rain (1952)	542	4.097786	4.067969
3909	Dr. Horrible's Sing-Along Blog (2008)	185	4.148649	4.062224
1777	Stop Making Sense (1984)	119	4.142857	4.019316
2647	Fiddler on the Roof (1971)	165	4.048485	3.968606
600	Wizard of Oz, The (1939)	1229	3.947518	3.937552
588	Gay Divorcee, The (1934)	57	4.149123	3.935381
1438	Nashville (1975)	128	4.015625	3.923279
624	Top Hat (1935)	120	4.004167	3.909188
3297	Day at the Races, A (1937)	51	4.117647	3.899730

Figure 3.2: Extraction of best movies

We can get the list of movies which are similar to the movie we decided to apply filter and the list of similar movies based on the content and specific genres as shown in Figure 3.3.

```
get_similar_movies_based_on_content(19338)
```

```
[(6260, 4.0), (6822, 4.0), (11018, 4.0), (11826, 4.0), (15203, 4.0), (16733, 4.0), (18315, 4.0), (18349, 4.0), (19338, 4.0)
[6260, 6822, 11018, 11826, 15203, 16733, 18315, 18349, 19338, 20615, 9]
```

	title	genres
6260	Matrix Reloaded, The (2003)	Action|Adventure|Sci-Fi|Thriller|IMAX
6822	Matrix Revolutions, The (2003)	Action|Adventure|Sci-Fi|Thriller|IMAX
11018	Poseidon (2006)	Action|Adventure|Thriller|IMAX
11826	Spider-Man 3 (2007)	Action|Adventure|Sci-Fi|Thriller|IMAX
15203	Iron Man 2 (2010)	Action|Adventure|Sci-Fi|Thriller|IMAX
16733	Sanctum (2011)	Action|Adventure|Drama|Thriller|IMAX
18315	Bourne Legacy, The (2012)	Action|Adventure|Drama|Thriller|IMAX
18349	Mission: Impossible - Ghost Protocol (2011)	Action|Adventure|Thriller|IMAX
19338	Skyfall (2012)	Action|Adventure|Thriller|IMAX
20615	G.I. Joe: Retaliation (2013)	Action|Adventure|Sci-Fi|Thriller|IMAX
9	GoldenEye (1995)	Action|Adventure|Thriller

Figure 3.3: Extraction of similar movies

For example, we desire to get similar movies as 'Gone Girl' so our model provides this facility too. By using this model we can also get similar movies according to content of the previous movie as shown in Figure 3.4.

```
# Getting other top 10 movies which are watched by the people who saw 'Gone Girl
get_other_movies('Gone Girl (2014)')
```

title	userId	perc_who_watched
Gone Girl (2014)	61	100.0
Matrix, The (1999)	54	88.5
Inception (2010)	53	86.9
Fight Club (1999)	52	85.2
Shawshank Redemption, The (1994)	52	85.2
Dark Knight, The (2008)	52	85.2
Lord of the Rings: The Fellowship of the Ring, The (2001)	51	83.6
Lord of the Rings: The Return of the King, The (2003)	50	82.0
Pulp Fiction (1994)	48	78.7
Silence of the Lambs, The (1991)	47	77.0

Figure 3.4: Extraction of other movies of same content

By calculating the rmse score of SVD with various values of k, matrix factorization is performed and the resulting list of movies predicted by the movie recommendation system is obtained as shown in Figure 3.5.

```
#Calculate the rmse sscore of SVD using different values of k (latent features)
rmse_list = []
for i in [1,2,5,20,40,60,100,200]:
    #apply svd to the test data
    u,s,vt = svds(train_data_matrix,k=i)
    #get diagonal matrix
    s_diag_matrix=np.diag(s)
    #predict x with dot product of u s_diag and vt
    X_pred = np.dot(np.dot(u,s_diag_matrix),vt)
    #calculate rmse score of matrix factorisation predictions
    rmse_score = rmse(X_pred,test_data_matrix)
    rmse_list.append(rmse_score)
    print("Matrix Factorisation with " + str(i) +" latent features has a RMSE of " + str(rmse_score)

Matrix Factorisation with 1 latent features has a RMSE of 1.8695042787105645
Matrix Factorisation with 2 latent features has a RMSE of 1.443510955312454
Matrix Factorisation with 5 latent features has a RMSE of 1.372482165500306
Matrix Factorisation with 20 latent features has a RMSE of 2.1698736654999347
Matrix Factorisation with 40 latent features has a RMSE of 2.214808709690625
Matrix Factorisation with 60 latent features has a RMSE of 2.4373540250166057
Matrix Factorisation with 100 latent features has a RMSE of 2.6117524246055877
Matrix Factorisation with 200 latent features has a RMSE of 1.3935050497840982
```

Figure 3.5: Prediction of movies by the movie recommendation system

Then we can provide the user with a list of movies to watch as shown in Figure 3.6.

This recommendation system is 65.8 percent accurate, which is quite impressive given the lack of an ideal recommendation system.

`op 10 predictions for User 1`

	ratings	movieId	title	genres
0	5.071427	1142	Get Over It (1996)	Drama
1	4.746949	5622	Charly (2002)	Comedy|Drama|Romance
2	4.350751	312	Stuart Saves His Family (1995)	Comedy
3	4.338692	2796	Funny Farm (1988)	Comedy
4	4.213159	994	Big Night (1996)	Comedy|Drama
5	4.153872	1146	Curtis's Charm (1995)	Comedy|Drama
6	4.113652	2605	Entrapment (1999)	Crime|Thriller
7	4.077832	254	Jefferson in Paris (1995)	Drama
8	4.038245	1159	Love in Bloom (1935)	Romance
9	4.012471	287	Nina Takes a Lover (1994)	Comedy|Romance

Figure 3.6: Extraction of best movies to watch

Conclusion

In general, each strategy is applicable to any type of assistance data, though different strategies may result in varying degrees of efficiency and effectiveness. To fully exploit the complementary nature of various strategies, we believe that designing deep learning strategies will typically result in superior performance. Used in conjunction with deep learning help data for collaborative recommendation, including the ”kNN and matrix decomposition” algorithm. Finally, so far, KNN and matrix factorisation have been implemented successfully on the schema, with quite satisfactory and effective results. Deep learning techniques will increasingly be used to solve other difficult problems in the future, including video classification, social network analysis, image classification taken to a new level, and logical inferences. Additionally, we may sell our scheme as a product or service to businesses in need of an effective recommendation system.

3.8.2 Image and Video Recognition

To process image and video processing generally neural network is involved which role several channels of images then collect them one layer after another in order to reduce the size of the image before passing. For example, the area of bioinformatics where three layered architecture extracted from gamma images and fed comparator network and finally utilised during radiotherapy. Other category of image processing is on human action recognition is identified using various techniques like LSTM has used regression analysis is used, Affective-MIT Facial expression dataset where author has trained group of frames is expressed by author through model [177]. In the same problem to a set-to-set matching problem Deep Embedding Network is solved by increasing decreasing and increasing inter-object and intra-object respectively. Using CNN model, GoogleNet Inception-v3 has developed image-based search engine. To compress video sensing, deep neural network architecture is

used. To improve deep image learning network on architecture and constraints many applications were proposed.

1. Audio processing:

    - Noise power spectral density estimation has conducted to reduce noise.
    - Cost-sensitive deep ensemble learning mechanism proposal was proposed for speech separation in a noisy environment.
    - For automatic speech recognition, a deep neural network is used.

2. Natural Language Processing and Text Analysis:

    a) For natural dialog simulation, on two virtual agents an architecture a LSTM encoder-decoder is applied.

    b) Construction of a natural language descriptions is constructed by utilising CNNs where classification is classified from aerial landscape and image detection.

3. Autonomous Systems and Robotics:

    - Through deep reinforcement learning and off-policy updates a robot's arm is trained to perform manual task independently.
    - User interacts with the robot and robot learns simultaneously in the application of Supervised Progressively Autonomous Robot Competencies(SPARC)
    - A deep learning pipeline is used to train robot through tele-operation.
    - The proposed model Grasp Quality Convolutional Neural Network(GQ-CNN) and it predicts the output based on probability from depth images.
    - A single deep CNN model for pose estimation and 3D object detection over state-of-the-art system is proposed.

To detect an unsupervised object by utilising stereo vision as an input and based on a Deep Boltzmann support vector machines, Deep Boltzmann machines (DBMs) and auto-encoders(AEs). Deep flow tool used to obtain the information of data flow from sensitive source to sinks. Using deep learning techniques, the real time catches the detection of ongoing attack by preventing SQL. Apart from these many other application areas have been identified such as Economics, Finance, Market Analysis and others, Physical science, and Computational Biology [177].

The authors (Ledig et al.) focused on image super resolution, present the first framework of generative adversial model (SRGAN) and acknowledged that it is capable of highlights the limitation of PSNR-focused images as well as infer photo-realistic natural images for 4 X upscaling factors. To do this, a proposal of perceptual loss function consist of content and adversial loss is used. The perceptual similarity motivates the content loss and adversial loss forces the proposed using discriminator network and provides a solution to the natural image manifold. That network is trained to differentiate between

original photo-realistic and super-resolved image. Perceptual quality is gained heavily by testing MOS(mean-opinion-score) using SRGAN. The use of deep residual network recovers the texture of on public benchmarks from densed down sampled images [178].

3.8.3 Fake videos and images

Fake videos can be automatically detected with the support of deep learning and to do this, CNN(Convolutional Neural Network) that extracts frame-level features is used. RNN (Recurrent Neural Network) and its features are used to learn to classifies whether a video has been modified or not. To do this, multiple 600 manipulated videos are calculated and evaluated using LSTM structure to accurately predict if the video are manipulated or not within 2 sec using a simple pipeline architecture [179]. A new method [180] has been introduced. A new method has been introduced to work on to expose the deep neural network fake videos. The detection method represents physiological signal and is based on the blink of eye in the synthesised videos. Previous work has studied in eye blinking where the existing work is on CNN-based and not it is extended to LSTN. The training of new model is based on the datasets image of eye open states. Later, tested the proposed algorithms, in deepfake algorithm detection is based on the blink eyes to detect fake and authenticated videos The authors (Liu et al.) used datasets are CEW that includes 1193 and 1232 images for closed and open respectively.

One of the application of DL which is deep fake algorithm named, deepfake algorithms is proposed. This framework creates similar fake videos and images like authenticated videos and images. Through this approach, a technological phenomenon is proposed which is implemented in a manner that it automatically assess and detect the integerity of digital visual media.

Another approach of fake images detection is in which the presentation of Text Conditioned Auxiliary Classifier Generative Adversial Network (TAC-GAN) is presented such that the synthesising images from their text descriptions where 102 dataset of flowers have used. Also, the proposal evaluates the discrimination of the generated images with diversity using the Multi-Scale Structural Similar index as well as with Inception-Score. Although various other proposal have proposed but their performance is outperformed with its inception-score is 3.45 [181]. The representation of the state-of-art method has applied on unsupervised domain application. The PixelDA models has given outperformed performance on the previous set of unsupervised domain adaptation scenario as well as brings challenge of ”Synthetic Cropped Linemod to Cropped Linemod”. This method has proved that the number of errors for pose estimation produces a better result as compared to previous one.In the proposal GAN-based techniques is used that stabilize the task-specific loss and novel content. The proposed model disconnects from the task-specific architecture to the process of domain adaptation to give advantages to make the process easy to understand via visualisation of the adapted images outputs of the models [182]. The model process begins its process in the context of image for pixel-level domain adaptation (PixelDA) . At the source side, a labeled dataset is given and unlabeled dataset is in target domain which aims from the source side domain. A classifier gets trained on the data so to generalise the target domain. In this work, the authors (Nousmalis et al.) have decoupled the primary

function of domain adaptation process from the task-specific classification process to capture images in such a way as if they were sampled from the target domain.

Another techniques name inversion in which using a pretrained GAN to the latent space projects the images is sampled in data. Based on reconstruction loss the performance of GAN is quantified as well as the ability of identifying which attributes of a data set is used to train a GAN model is able with the techniques of an inversion technique. This proposal also demonstrates how an inversion technique can be used to quantify the performance of various GAN models which is trained on three Image datasets. The characteristics of GAN model is considered as a difficult as well as well trained model.In regards to difficult model, it captures few features of the target image as it only has Z* of latent representation. And in terms of well trained model it is able to find the representation Z* in response to the image X which is given to the target image X .Z* after passing through the generator produces an image G(Z*) as similar to target image [183]. Another approach using contrastive loss is deep learning based approach that detects fake image. Secondly, the structure of reduced Dense Net in a two-streamed network accepts pairwise information as an input. Thirdly, to distinguish between fake and real images a proposal of common fake feature network is trained. Fourthly, The contenation of classification layer with proposed common fake feature is able to detect that input image is real or fake

VICTORIA UNIVERSITY

THE NEW WAY TO DO UNI

OFFICE FOR RESEARCH TRAINING, QUALITY AND INTEGRITY

DECLARATION OF CO-AUTHORSHIP AND CO-CONTRIBUTION: PAPERS INCORPORATED IN THESIS

This declaration is to be completed for each conjointly authored publication and placed at the beginning of the thesis chapter in which the publication appears.

1. PUBLICATION DETAILS (to be completed by the candidate)

Title of Paper/Journal/Book: Enhancing system security by Intrusion Detection using Deep learning [redacted]

Surname: Sama First name: Lakshit

Institute: Institute for Sustainable Industries and Liveab Candidate's Cont [redacted] 70

Status:

Accepted and in press: ✔ Date: 26/06/2022

Published: ☐ Date:

2. CANDIDATE DECLARATION

I declare that the publication above meets the requirements to be included in the thesis as outlined in the HDR Policy and related Procedures – policy.vu.edu.au.

[redacted] 05/08/2022

Date

3. CO-AUTHOR(S) DECLARATION

In the case of the above publication, the following authors contributed to the work as follows:

The undersigned certify that:

1. They meet criteria for authorship in that they have participated in the conception, execution or interpretation of at least that part of the publication in their field of expertise;
2. They take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;

3. There are no other authors of the publication according to these criteria;
4. Potential conflicts of interest have been disclosed to a) granting bodies, b) the editor or publisher of journals or other publications, and c) the head of the responsible academic unit; and
5. The original data will be held for at least five years from the date indicated below and is stored at the following **location**(**s**):

Name(s) of Co-Author(s)	**Contribution (%)**	**Nature of Contribution**	**Signature**	**Date**
Lakshit Sama	70	Formal analysis, Methodology. Writing	[redacted]	05/08/2022
Hua Wang	15	Resources, Supervision		05/08/22
Paul Watters	15	Validation, Review		05/08/22

[redacted]

Updated: September 2019

Chapter 4

Proposed work: Enhancing system security by Intrusion Detection using Deep learning

Network intrusion detection has become a hot topic in cybersecurity researchers for better advancements in deep learning. The research is lacking an objective comparison of the various deep learning models in a controlled setting, notably on recent intrusion detection datasets, even though several outstanding studies address the growing body of research on the subject. In this paper, a network intrusion scheme is developed as a solution of the discussed issue. The four different models are built and are experimented with NSL-KDD dataset. These deep learning models are LightGBM, XGBoost, LSTM, and decision tree. For the validation of the proposed scheme, the proposed scheme is also experimented with UNSW-NB15 dataset and CIC-IDS2017. However, the experiments concluded that the proposed scheme outperforms and the discussion is also illustrated.

4.1 Introduction

Over the previous ten years, the size, use, and complexity of computer networks have all increased dramatically, and lots of technologies to secure data have been developed [184, 185]. Internet of Things have evolved as new sorts of devices and network infrastructures. The safety of these networks and systems has grown increasingly important as they grow in size and complexity [186, 187]. According to CyberEdge's 2021 Cyberthreat Defense Report, attacks on major enterprise networks have increased over the past five years. Among these attacks, malware, intrusion, spam, and denial of service (DoS) are most common and prominent attacks.

Network intrusion detection is one of the most difficult aspects of network security to deal with. Though tremendous progress has been made, the bulk of solutions still use less competent signature-based techniques instead of anomaly detection methods. Many factors contribute to this reluctance, including as the high false error rate (and related costs), difficulty in acquiring trustworthy training data, length of training data, and system dynamics. There will come a point where the current scenario

will lead to inefficient and inaccurate detection as a result of such tactics. For this problem, we must come up with an acceptable anomaly detection method that can keep pace with the ever-changing nature of current networks.

The implementation of machine learning and deep learning techniques like Naive Bayes, Support Vector Machines, and Decision Trees has been a major focus in Network intrusion research in recent years. Improved detection accuracy has been achieved by using these strategies. Expert knowledge is necessary to process data e.g. to find important data and patterns, but these techniques have some drawbacks, such as the requirement of relatively high level of human engagement. In addition to the fact that it takes a lot of time and money, this method is also prone to mistakes. Similar to this, in order to operate, a large volume of training data is necessary (together with the corresponding time overheads), which can be difficult in a dynamic context.

Deep learning is a research area that is currently receiving a lot of attention from researchers in a variety of fields because of the above limitations. An advanced branch of machine learning, this can help overcome the problems with shallow learning. Deep learning's remarkable layer-wise feature learning has so far been shown to outperform shallow learning techniques, if not match them. Using this tool, it is possible to conduct a deeper analysis of network data and identify any anomalies more quickly.
It is possible to analyse network traffic flows through the use of misuse detection, anomaly detection, and signature based analysis techniques. The attacks are detected through the use of predefined signatures and filters, which are known as misuse detection. It is dependent on human inputs to keep the signature database up to date on a continuous basis. This method is accurate when it comes to identifying known attacks, but it is completely ineffective when it comes to identifying unknown attacks. Anomaly detection is a technique that employs heuristic mechanisms to identify previously unidentified malicious activities.

Detection system produces a high number of false positives in the majority of the scenarios. Most organisations employ a combination of misuse detection and detection techniques in their commercial delivering in order to combat this issue. When compared to the other detection methods, signature based analysis is the most effective because it operates at three different layers: the network layer, the application layer, and the transportation layer. This makes use of the vendor specification settings that have already been defined in order to detect deviations from the appropriate services and algorithms. Even though investigations in this work have recently been proposed to improve the intelligence of intrusion detection systems, there has been no research to compare the performance of such supervised learning models against publicly available datasets.

The most typical problems with existing approaches based on machine learning designs are as follows: first, the frameworks produce a false - positive rate with a wider range of attacks; second, the frameworks are not generalizable because existing studies have primarily used a single dataset to review the performance of the model learning model; third, the models researched so far have completely overlooked today's massive network traffic; and finally, the solutions are required to withstand today's

high-volume network traffic. The primary motivation for this work is to evaluate the efficacy of various deep learning classifiers, with a particular emphasis on identifying those that are most effective. A new deep learning model is proposed in this paper to allow intrusion detection to operate within the context of today's networks.

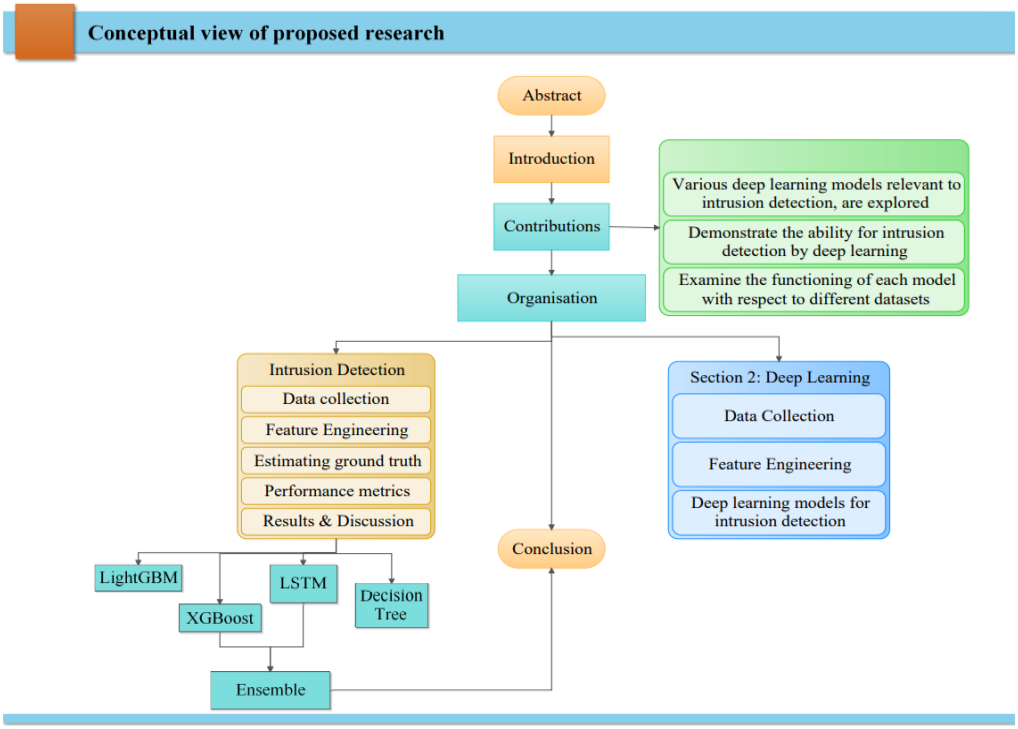

Figure 4.1: Conceptual view

4.2 Contributions

The following are the contributions done to propose the intrusion detection system:

1. The deep learning models are designed with the objective of intrusion detection.
2. The network intrusion system is built with the deep learning models and validated with the dataset.
3. The validation of the proposed scheme is done by experimenting the proposed scheme with benchmark datasets UNSW-NB15 Dataset and CIC-IDS2017.

4.3 Related Work

Security, forensics, and anomaly detection have all relied heavily on network monitoring techniques. Despite this, recent technological advancements have posed numerous challenges for network intrusion detection. There are a number of pressing concerns, including:

Author	Technique	Model	Dataset	Results
Shyu et al., 2011 [12]	Outlier Detection	Principal Component Analysis	KDD'99	99% Accuracy
Revathi et al., 2013 [9]	Network Intrusion Detection System	Random Forest	NSL-KDD dataset	Improved Accuracy
Khaled et al., 2016 [1]	Unsupervised Dimension reduction	Adaboost	NSL-KDD dataset	97.9 percent
Yin et al. [13]	Intrusion detection	Recurrent Neural Networks	NSL-KDD dataset	Improved accuracy
Shone et al. [11]	Intrusion detection system	Deep Auto Encoder	KDD Cup'99 and NSL-KDD	High precision
Sama et al. [10]	Network Intrusion Detection	Deep learning	KDD dataset	Improved accuracy

Table 4.2: Summary of existing techniques

1. **Diversification** - In recent years, the number of new or customised protocols used in modern networks has increased. Devices with network and Internet access are a factor in this, to some extent. The outcome is that the ability to distinguish between regular and abnormal traffic and/or behaviour is getting increasingly difficult as a result of these changes. The dynamic nature of modern networks makes it impossible to forecast how they will behave. As a result, it is difficult to establish a trustworthy behavioural norm. Learning models' usefulness is called into question as a result.

2. **Storage** - As data storage and network traffic continue to grow, so does the volume of data. According to current predictions, the amount of data in existence will reach its peak in 2022. As a result, modern networks' traffic capacity has been dramatically expanded in order to cope with the high volume of traffic currently witnessed. Faster than 100 Gigabits per second (Gbps) wirespeeds are currently common on many backbone links. At this rate, it is difficult to provide a network system that meets acceptable standards of accuracy, efficacy, and efficiency.

3. **Adaptability changes** - Adaptability Dynamic technologies like virtualization and Software Defined Networks are becoming more common as a result of this and the ramifications of such innovations.

4. **Low-frequency attacks** - These attacks have typically defied earlier anomaly detection methods, including artificial intelligence systems. An imbalance in the training dataset causes Network Intrusion Detection System to be less accurate when it comes to detecting low-frequency attacks. Many new technologies have been introduced to lessen the network's dependence on old technology and management methods.

5. **Cost** - In order to maintain the accuracy levels stated above, one cannot depend upon. Consequently, a comprehensive and accurate view calls for a greater level of granularity as well as depth and contextual awareness of the issues involved. As a result, the opportunity cost is high in terms of both money and time.

The majority of the work done for the intrusion detection computational modeling section is done using datasets that are similar in both training and testing. It is difficult to generalise real-time events

from these datasets because of the nature of the data. As a result, when the bulk of these predictive are exposed to real-world network traffic, their performance measures deteriorate. Different methods for classifying connections with abnormalities in order to detect network intrusions have been proposed, including the use of heuristics.

– According to Shyu et al. [188], an unique scheme based on Principal Component Analysis was presented, with anomalies being treated as outliers. With the KDD'99 dataset, the anomaly detection system performed significantly better. The detection rate increased to ninety nine percent, but the percentage of false alarms plummeted to as low as 1 percent, according to the results.

– According to Revathi et al. [189], a detailed analysis of the NSL-KDD dataset was carried out using only relevant features, both with and without feature reduction of the dataset, on different classification algorithms, among others. Across both cases, Random Forest was found to have the highest accuracy in test accuracy. Deep learning techniques make it possible to create Network Intrusion Detection System that are both versatile and resilient.

– Khaled et al. [190] proposed an integrated approach for intrusion detection that included one hidden layer of Deep Boltzmann Device for unsupervised dimension reduction and Adaboost with multi-class gentle for classification, as well as one hidden units of Restricted Boltzmann Equipment for unsupervised dimension reduction. Using the entire 10% KDD-Cup'99 test dataset, the model achieved a detection rate of 97.9 percent, surpassing the industry standard. The KDDCup'99 dataset does not represent a challenge that is comparable to that of real-world network traffic in any way. Niyaz and colleagues used regression to classify data. The model was tested against the benchmark NSL-KDD dataset for classification in two classes, five classes, and 23 classes, and the findings were encouraging, with the model demonstrating improved performance in all three classifications.

– Yin et al. [191] suggested a deep learning strategy for intrusion detection based on Recurrent Neural Networks, which they believe will be successful (RNN). The experimental results revealed that the model's performance in both binary and multiclass classification was promising, and that the model was capable of classifying with high accuracy.

– According to Shone et al. [129], a novel deep learning categorization model was suggested that developed utilising layered Non-Symmetric DeepAuto Encoder (NDAE) in unlabeled data learning and the RF classification algorithm to classify the results. The model has been implemented in Tensor Flow and tested on the benchmark datasets KDD Cup'99 and NSL-KDD. With the reduction in training time, the model was able to maintain a consistent level of classification results while also achieving a high degree of precision and recall.

– The difficulty in developing a robust Network Intrusion Detection System is the lack of real-time network data patterns that include both intrusions and normal network usage, the presence of

constantly evolving and changing known attacks, the need for lengthy training periods, and a lack of knowledge about the modifications that should be made to existing datasets. Even though a model may attain great accuracy when compared to test datasets, the model's accuracy always appears to decline when compared to real-world network traffic.

– The Network Intrusion Detection System, which was developed utilising deep learning, was able to buck this trend to some extent. The majority of deep learning-based studies discovered in the literature had a significant detection rate and were able to detect abnormalities that were previously undetected in some cases [12]. Although, the vast majority of the work in the intrusion prevention field remains to be done in order to develop a practical and effective Network Intrusion Detection System, this appears to be the most promising path forward. A Self-Taught Training (STL) technique for unsupervised training and a soft-max algorithm for unsupervised feature training [5].

The above discussed existing works are summarised in Table.

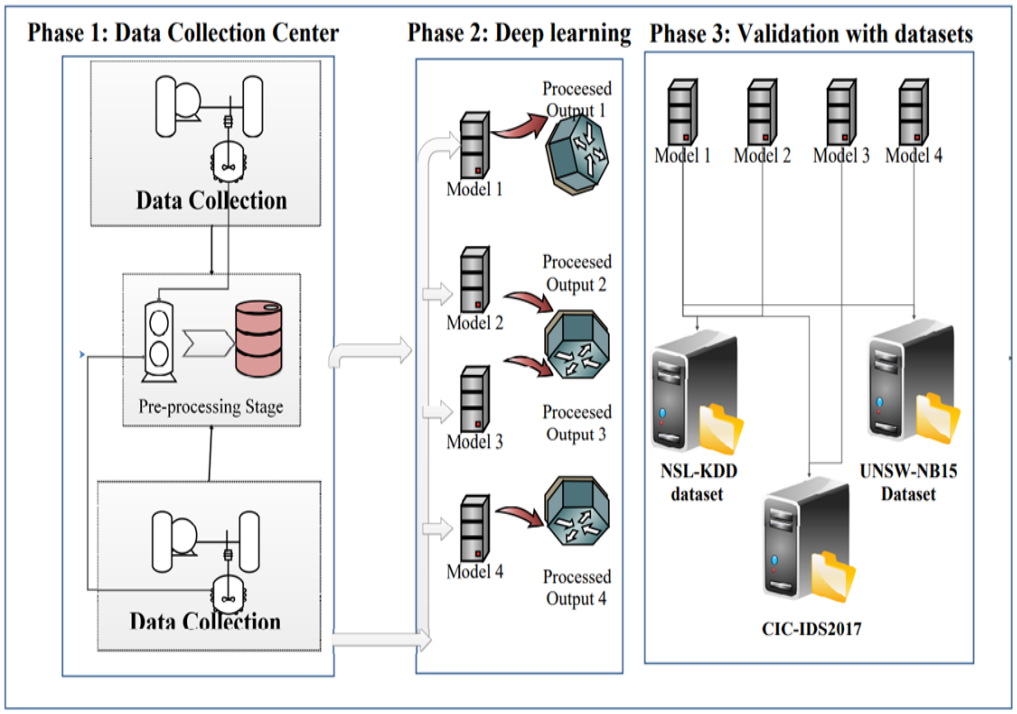

Figure 4.3: Steps followed in the proposed work

4.4 Proposed Scheme

The proposed scheme comprises of building a network intrusion detection system (NIDS). The system is designed with four deep learning models, LightGBM, XGBoost, LSTM, and decision tree. Each model is trained with the patches of data and is able to detect the attacks. We have developed the scheme by its validation on NSL-KDD dataset. However, the scheme has been validated with other

datasets as well as discussed in next section. Below is the discussion of four models deployed at the proposed scheme. The steps followed are discussed in Figure 2.

– LightGBM: Light Gradient Boosting Machine (LightGBM) is a gradient boosting platform which uses algorithms based on tree learning. The segment where LightGBM shines is in the case of big datasets, such as the one used in our implementation. Light is the product of running at a very high speed and using much less memory in LightGBM. LightGBM works to allow complete, effective use of the gradient boosting system by first processing the dataset and making it lighter.
Every loss to the network is computed by the above equation. Here, x is the number of features of the proposed scheme.

– XGBoost: EXtreme Gradient Boosting (XGBoost), originated from GBDT. Owing to its accuracy and relatively fast speed compared to traditional machine learning algorithms, it was introduced earlier than LightGBM and is commonly used in machine learning. For performance, XGBoost chooses an algorithm based on histograms. The histogram-based algorithm uses bins that are separated by data point characteristics into discrete types. It is therefore, more accurate than the presorted process, but all the potential split points have to be enumerated as well.

– LSTM: Long-Short Term Memory (LSTM) is commonly used as a deep neural network for Time series data processing, which is an enhanced Recurrent Neural Networks (RNNs) based model. RNN uses an internal state to represent previous input values, allowing temporal background to be captured. For long input sequence, it is not easy to train LSTM. However, compared to RNN, LSTM can capture the background of longer time series. In our problem, the census parameters are lengthy in nature, so to minimise the length, we used LSTM. LSTM work on gates which provides the gateway for the timestamp to control it. The inputs for the proposed scheme are the various features of dataset (as discussed in next Section). The outputs produced using tanh function on the basis of different timestamps.

– Decision tree: An example of a non-parametric machine learning model, Decision Trees are useful both for classification and regression. If we build them appropriately (which we should), decision trees can generate either a categorical or a numerical forecast depending on the number of characteristics we include in them. Two types of elements, nodes and branches, are used to build them. Data features are examined at each node in the training process or while creating predictions in order to separate observations or to follow a specific path for an individual data point in training.

Entropy and information gain are the two parameters computed by decision tree. Gain is computed using these two parameters, which decides the position for each instance.

As every classifier is capable of performing in its own strategy, So, lets analyse the working of each model with the different sets of features.

4.5 Experimental Setup

To evaluate the performance of the proposed framework, experiments were conducted on Intel Xeon® CPU e5-1620, 64GB of RAM running on Windows 10. We experimented with the models on Google Collaboratory using Python and Pytorch library. The training data is portioned as 60% of the total data fetched from each of the four datasets.

4.6 Results and Discussion

4.6.1 Dataset-1: NSL-KDD dataset

Kaggle has a detailed description of the NSL-KDD dataset. When it was first developed in 1999, it was widely used in 2009 for detecting intrusions. The authors [11, 7] now use this dataset as the benchmark dataset. These datasets are named KDDTrainC and KDDTestC, respectively, and are used for training and testing. 41 features are listed for each traffic record and 1 class label, including basic features (1-10) and content - based features (11- 22), as well as traffic-based characteristics (23-41) as shown in Table . U2R, R2L, DoS, and Probe were the four types of attacks that were shown. Training samples and testing samples are separated in the data, as shown in the table.

Class	Count of testing samples	Count of training samples
DoS	23985	38145
R2L	1632	100
Probe	428	400
U2R	25	5
Normal	6065	9721

Table 4.4: Samples of dataset-1

4.6.2 Dataset-2: UNSW-NB15 Dataset

It is possible to categorise the attacks in the UNSW-NB15 dataset into one normal class and nine attack classes, namely the Analyze class and the Backdoor class and the DoS class and the Exploits class and the Generic class and the Reconnaissance class and the Worms class. Table lists the eleven data types and their descriptions. The UNSWNB15 dataset has 257,673 data instances, including 175,341 training data instances and 82,332 testing data instances .

4.6.3 Dataset-3: CIC-IDS2017

There are 2,830,743 records in the CIC-IDS2017 database, and each record has 78 features associated with it. The CIC-IDS2017 database contains the latest attacks and the outcomes of network traffic analysis to termed flows depending on the source and destination access technologies and time stamps in the database.

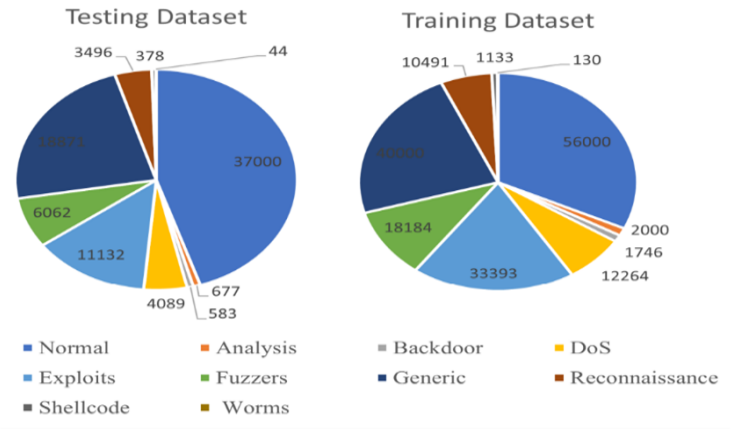

Figure 4.5: Training and Testing data of dataset-2

Model 1		Model 2		Model 3		Model 4	
LightGBM	**Settings**	**XGBoost**	**Settings**	**LSTM**	**Settings**	**Decision Tree**	**Settings**
Epoch	50	Epoch	100	Epoch	150	Epoch	250
Error rate	0.001	Error rate	0.001	Objective function	binary	Level depth	8 nodes, 10 leafs
Batch rate	28	Batch rate	56	Folds	3	Number of splits	5
Data dimension	41	Data dimension	41	Early stopping rounds	50	Nodes shuffling	True

Table 4.6: Setting of hyper parameters for each deep learning model

An updated intrusion detection database, CIC-IDS2017, covers important criteria such as Botnet, SQL injection, port scan and DDoS. It also includes XSS and XSS injection. The below are the steps required to build the network intrusion detection system:

- **Data preprocessing**: The data should be clean before it is trained with the model. The major considerations are that firstly there should not be any NULL values in the data, second is that the data form should be same, i.e., normalisation.
- **Feature selection**: The selection of the features is crucial because the irrelevant features may lead to inaccuracies. There two categories of feature selection techniques, filter methods and wrapper methods.
- **Deep learning model**: The convolutional neural network in adopted in the proposed scheme which detects the intrusion of network layer by layer. The different layers like activation, hidden and polling layer form the network.

4.7 Impact of deep learning on the proposed scheme

The deep learning models are deployed in the proposed scheme. The experimentation of the models is being done using different datasets. The hyper parameters settings for each model in illustrated

Attacks	Normal	U2R	R2L	DoS	Probe
LightGBM					
Normal	9212	90	30	46	248
U2R	18	45	20	22	70
R2L	67	31	560	88	164
DoS	0	22	54	6470	620
Probe	25	82	17	61	3529
XGBoost					
Normal	9316	66	20	66	218
U2R	38	25	10	32	70
R2L	82	16	610	68	134
DoS	22	0	74	6270	643
Probe	5	62	8	70	3530
LSTM					
Normal	9516	66	10	66	118
U1R	58	15	10	51	70
R1L	81	16	610	68	154
DoS	11	0	74	6170	645
Probe	5	61	8	70	5550
Decision Tree					
Normal	9343	33	40	33	447
U4R	37	43	40	34	70
R4L	74	43	340	37	434
DoS	44	0	74	3470	343
Probe	3	34	7	70	3330

Table 4.7: Attack matrix by different models on NSL-KDD dataset

in Table 4. Firstly, the models are experimented with the NSL-KDD dataset. The four categories of attacks are identified by each of the model and the performance of each model with respect to different attacks is illustrated below. The table 5 presents the attacks identified by LightGBM on NSL-KDD dataset. The Table 6 presents the attacks identified by XGBoost on NSL-KDD dataset.The table 7 presents the attacks identified by LSTM on NSL-KDD dataset. And, the table 8 presents the attacks identified by Decision Tree on NSL-KDD dataset.

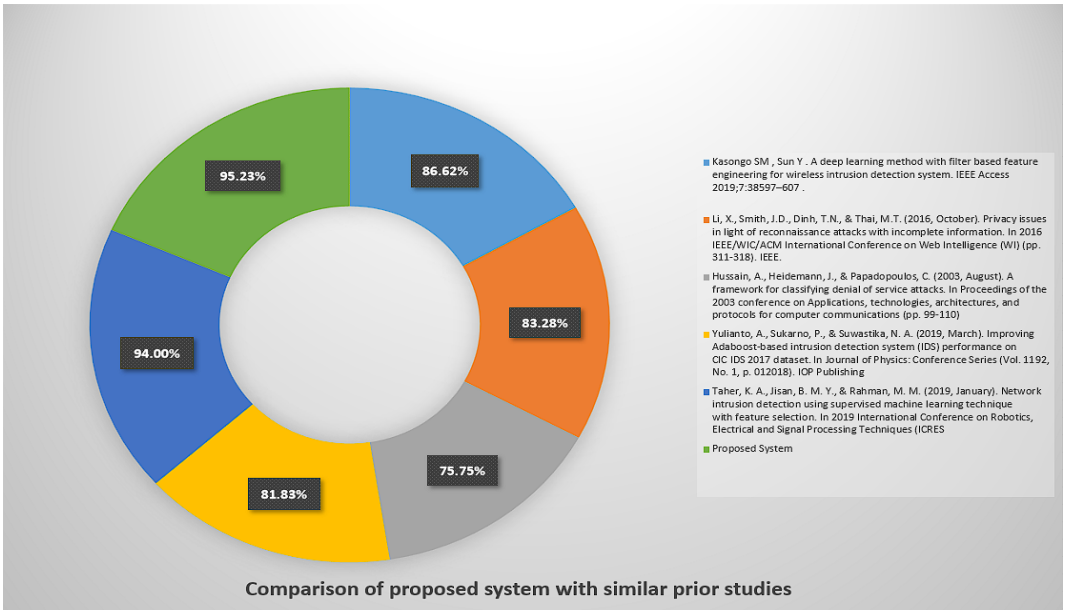

Figure 4.8: Comparison of proposed scheme

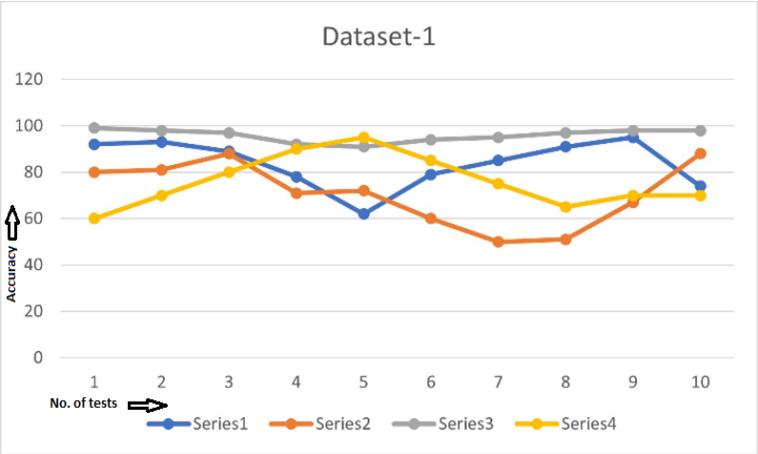

Figure 4.9: Accuracy of four models with Dataset-1

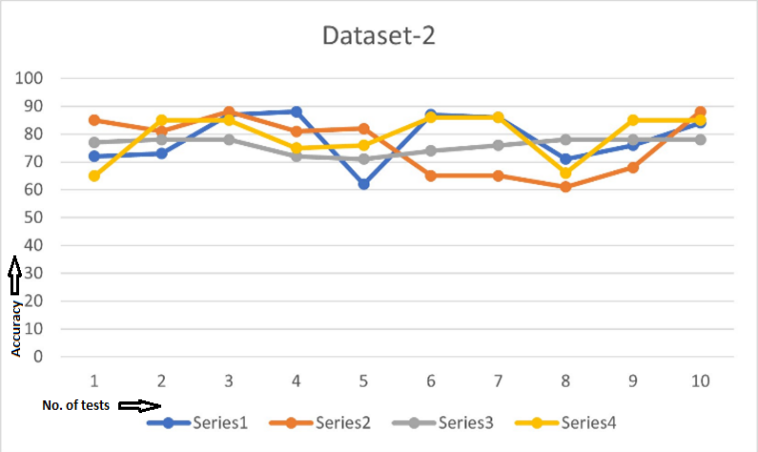

Figure 4.10: Accuracy of four models with Dataset-2

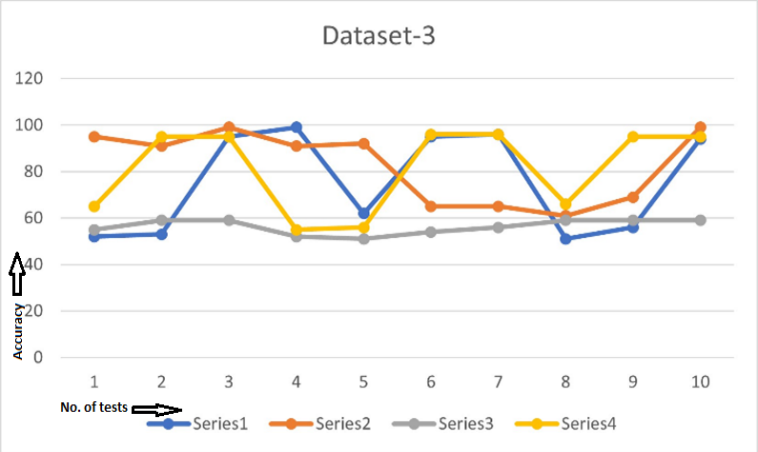

Figure 4.11: Accuracy of four models with Dataset-3

4.8 Impact of different datasets on the proposed scheme

The different models experiment with different datasets for the validation of the proposed scheme. Each model performed differently with every data. In the below discussion, series 1 refers to LightGBM, series 2 refers to XGBoost, series 3 refers to LSTM and series 4 refers to Decision Tree. These are the

observations:

- NSL-KDD dataset: Series 3 outperforms when experimented with this dataset. However, series 1 also performs well with some classes as shown in Figure 4.8.
- UNSW-NB15 dataset: Series 1 and series 4 almost performed same when experimented with this dataset. However, series 2 and series performed constant as shown in Figure 4.9.
- CIC-IDS2017: Series 2 and series 4 equally contributed when experimented with this dataset as shown in Figure 4.10.

4.9 Summary

In this chapter, we have discussed the difficulties that currently available Network Intrusion Detection System techniques face. In response to this, we have proposed our novel Network Intrusion Detection System method to the scientific community. We have developed a TensorFlow implementation of our proposed model and conducted extensive evaluations of the model's capabilities. We used the benchmark NSL-KDD dataset for our evaluations, and we were able to achieve very promising results. The four deep learning models are deployed and further experimented with on other benchmark datasets in addition to the one used for this experiment.

Chapter 5

Conclusion

5.1 Summary of the thesis

Building a safe network communication environment requires network intrusion detection. To improve an Network Intrusion Detection System's capacity to find more unauthorised and hostile network activity, effective detection methods are crucially needed. The proposed work presents the scientific community to our cutting-edge Network Intrusion Detection System technique. The scheme is being created using a TensorFlow and the implementation of suggested model is thoroughly assessed for the model's performance. For our assessments, the benchmark NSL-KDD dataset is employed, and the findings are quite encouraging. In addition to the benchmark dataset utilised for the experiment, the four deep learning models are deployed and further tested on additional benchmark datasets.

Along with the proposed work, the thesis is structured as follows:

In **Chapter 1**, the introduction is discussed covering the various network security parameters. The brief of network intrusion techniques is also given in this chapter. It is crucial to understand the need of detection of intrusion detection, so the case studies are presented. The first case is about the data breach in which the hackers gained access to the personal information of over 533 million Facebook users because of a data breach. This breach occurred when fraudsters exploited a flaw in Facebook's contact importer to scrape data from the company's servers.
The second case study is about the first ransomware assault on the government of Costa Rica which began during the week of April 10. Conti first publicised its assault on the finance ministry on its blog, where it exposes the identities of its victims and the data it has taken from them if they fail to pay the ransom. The chapter finally contributed towards discussing the preventive measures.

In **Chapter 2**, the discussion is all about the network intrusion and different categories of attacks. The categories of attacks such as network attacks, persistent threats. The attack discussion is mainly related to malware, botnet, ransomware. As the literature of attack detection system uses the datasets, So the various network intrusion datasets are also discussed. The standard dataset such as KDD 1999, NSL-KDD , UNSW-NB15 are discussed in detail.

In **Chapter 3**, firstly we have discussed that how machine learning techniques are used for network intrusion and the emphasis of deep learning in the system, which is the basis for our research. As we have used deep learning models for proposing the NID, so we have presented the deep learning models in detail. The models discussed are, mainly deep neural networks, recurrent neural network, convolutional neural network, auto encoder and deep belief network. Along with the various deep learning models, the application of deep learning, i.e., movie recommendation is also presented. This application is published in one of our research papers. However, in today's world, where everyone is concerned with ideas or guidelines to help them make a decision, this area of research includes a movie recommendation system that allows to present the user with a list of related movies. The user has been provided with a selection of movies recommended by our model, he or she merely needs to decide which film to view next.

In **Chapter 4**, we have presented our proposed framework for the detection of network intrusion system that comprises of four deep learning models and validated using three different datasets. The results are discussed in detail.

5.2 Future work

The network intrusion detection is developed in the proposed work. The proposed system used deep learning models and validated using standard datasets. However, there are several issues which could be addressed in future.

Reducing the complexity of models: According to studies, the reason why DL-based intrusion detection systems are growing in popularity is because they are good at identifying dangerous threats since they can learn deep characteristics. This is one reason why increasing numbers of individuals are adopting these systems. They have become better at recognising potentially harmful situations, which has increased their popularity. For these models to function, a substantial amount of computing power, storage space, and time is required to execute the DL algorithms that make them function. Because of these designs, it is now considerably more difficult to implement IDS in real time. On the other hand, the cost of current GPUs may be prohibitive for the majority of individuals. There must be a balance between how well a solution performs and how much it costs. It is feasible that GPU-based platforms or cloud-based services will reduce the cost of training models.

Improving the Efficiency of Network Intrusion Detection System: According to a recent study, detecting zero-day attacks is challenging owing to a large rate of false positives. Consequently, IDS may perform better if it has access to a more rigorous, up-to-date, and balanced dataset. Researchers have only tried a comprehensive and efficient Network Intrusion Detection System design a handful of times. The architecture necessitates a way to frequently update attack definitions stored in a dataset in order for the model to recognise new properties. The model is then only capable of obtaining fresh information. Due to this method, it will be easier to detect zero-day attacks and less false alarms would

be created.

Utilizing algorithmic frameworks constructed with DL technology: On the basis of current research, it has been recommended that DL-based algorithms should be used while developing IDS systems. Before an IDS solution can be developed for an IoT network, further research on deep reinforcement learning (DRL), and other deep learning (DL) approaches is required. Currently, the usage of DL in IDS applications is still in its infancy. In this area of study, it may be conceivable to investigate a hybrid technique that combines deep learning for feature extraction with machine learning for classification. It is probable that more study will be conducted on this topic. This will make it much simpler to comprehend the model as it now stands.

Bibliography

[1] Y. Qin, Q. Z. Sheng, N. J. Falkner, S. Dustdar, H. Wang, A. V. Vasilakos, When things matter: A survey on data-centric internet of things, Journal of Network and Computer Applications 64 (2016) 137–153.

[2] P. Vimalachandran, H. Liu, Y. Lin, K. Ji, H. Wang, Y. Zhang, Improving accessibility of the australian my health records while preserving privacy and security of the system, Health Information Science and Systems 8 (1) (2020) 1–9.

[3] M. Abomhara, G. M. Køien, Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks, Journal of Cyber Security and Mobility (2015) 65–88.

[4] H. Wang, Y. Wang, T. Taleb, X. Jiang, Editorial: Special issue on security and privacy in network computing, World Wide Web 23 (07 2019). doi:10.1007/s11280-019-00704-x.

[5] F. Zhang, Y. Wang, S. Liu, H. Wang, Decision-based evasion attacks on tree ensemble classifiers, World Wide Web 23 (5) (2020) 2957–2977.

[6] A. Makkar, M. S. Obaidat, N. Kumar, Fs2rnn: Feature selection scheme for web spam detection using recurrent neural networks, in: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, 2018, pp. 1–6.

[7] S. Biswas, J. Bicket, E. Wong, R. Musaloiu-e, A. Bhartia, D. Aguayo, Large-scale measurements of wireless network behavior, in: Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, 2015, pp. 153–165.

[8] H. Wang, J. Cao, Y. Zhang, Ticket-based service access scheme for mobile users, Aust. Comput. Sci. Commun. 24 (1) (2002) 285–292.

[9] W. Buchanan, Transmission control protocol (tcp) and internet protocol (ip), in: Applied Data Communications and Networks, Springer, 1996, pp. 87–109.

[10] A. Makkar, N. Kumar, M. S. Obaidat, K.-F. Hsiao, Qair: Quality assessment scheme for information retrieval in iot infrastructures, in: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, 2018, pp. 1–6.

[11] J. Yin, M. Tang, J. Cao, H. Wang, Apply transfer learning to cybersecurity: Predicting exploitability of vulnerabilities by description, Knowledge-Based Systems 210 (2020) 106529. doi:https://doi.org/10.1016/j.knosys.2020.106529.

[12] J. Yin, M. Tang, J. Cao, M. You, H. Wang, M. Alazab, Knowledge-driven cybersecurity intelligence: Software vulnerability co-exploitation behaviour discovery, IEEE Transactions on Industrial Informatics (2022) 1–9doi:10.1109/TII.2022.3192027.

[13] Y. Zhang, Y. Shen, H. Wang, J. Yong, X. Jiang, On secure wireless communications for iot under eavesdropper collusion, IEEE Transactions on Automation Science and Engineering 13 (3) (2016) 1281–1293. doi:10.1109/TASE.2015.2497663.

[14] M. Romagna, Hacktivism: Conceptualization, techniques, and historical view, in: The Palgrave Handbook of International Cybercrime and Cyberdeviance, Springer, 2020, pp. 743–769.

[15] M. You, J. Yin, H. Wang, J. Cao, Y. Miao, A minority class boosted framework for adaptive access control decision-making, in: Web Information Systems Engineering – WISE 2021, 2021, pp. 143–157. doi:10.1007/978-3-030-90888-1_12.

[16] M. You, J. Yin, H. Wang, J. Cao, K. Wang, Y. Miao, E. Bertino, A knowledge graph empowered online learning framework for access control decision-making, World Wide Web (2022) 1–22doi:10.1007/s11280-022-01076-5.

[17] M. Uma, G. Padmavathi, A survey on various cyber attacks and their classification., Int. J. Netw. Secur. 15 (5) (2013) 390–396.

[18] J. M. Olson, To catch a spy: The art of counterintelligence, Georgetown University Press, 2021.

[19] E. Kabir, J. Hu, H. Wang, G. Zhuo, A novel statistical technique for intrusion detection systems, Future Generation Computer Systems 79 (2018) 303–318. doi:https://doi.org/10.1016/j.future.2017.01.029.

[20] J. Yin, M. J. Tang, J. Cao, H. Wang, M. You, A real-time dynamic concept adaptive learning algorithm for exploitability prediction, Neurocomputing 472 (2022) 252–265. doi:https://doi.org/10.1016/j.neucom.2021.01.144.

[21] S. Thapa, A. Mailewa, The role of intrusion detection/prevention systems in modern computer networks: A review, in: Conference: Midwest Instruction and Computing Symposium (MICS), Vol. 53, 2020, pp. 1–14.

[22] V. Kouliaridis, G. Kambourakis, E. Chatzoglou, D. Geneiatakis, H. Wang, Dissecting contact tracing apps in the android platform, Plos one 16 (5) (2021) e0251867.

[23] J. Zhang, X. Tao, H. Wang, Outlier detection from large distributed databases, World Wide Web 17 (4) (2014) 539–568.

[24] M. Goyal, M. Dutta, Intrusion detection of wormhole attack in iot: A review, in: 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), IEEE, 2018, pp. 1–5.

[25] A. Daly, The introduction of data breach notification legislation in australia: A comparative view, Computer Law & Security Review 34 (3) (2018) 477–495.

[26] H. Wang, J. Cao, Y. Zhang, Access Control Management in Cloud Environments, Springer, Germany, 2020. doi:https://doi.org/10.1007/978-3-030-31729-4.

[27] M. E. Kabir, H. Wang, E. Bertino, Efficient systematic clustering method for k-anonymization, Acta Informatica 48 (1) (2011) 51–66.

[28] M. E. Kabir, H. Wang, E. Bertino, A conditional purpose-based access control model with dynamic roles, Expert Systems with Applications 38 (3) (2011) 1482–1489.

[29] A. Makkar, N. Kumar, A. Y. Zomaya, S. Dhiman, Spami: A cognitive spam protector for advertisement malicious images, Information Sciences 540 (2020) 17–37.

[30] L. Sun, H. Wang, J. Soar, C. Rong, Purpose based access control for privacy protection in e-healthcare services, Journal of Software 7 (11) (2012) 2443–2449.

[31] L. Sun, H. Wang, Access control and authorization for protecting disseminative information in e-learning workflow, Concurrency and Computation: Practice and Experience 23 (16) (2011) 2034–2042.

[32] X. Sun, H. Wang, J. Li, Satisfying privacy requirements: One step before anonymization, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, 2010, pp. 181–188.

[33] M. Alazab, S.-H. Hong, J. Ng, Louder bark with no bite: Privacy protection through the regulation of mandatory data breach notification in australia, Future Generation Computer Systems 116 (2021) 22–29.

[34] A. Makkar, U. Ghosh, D. B. Rawat, J. H. Abawajy, Fedlearnsp: preserving privacy and security using federated learning and edge computing, IEEE Consumer Electronics Magazine 11 (2) (2021) 21–27.

[35] C. Gutiérrez Espada, Counter-drone defense systems in the light of international law., Journal of Applied Business & Economics 22 (11) (2020).

[36] J. Silva, L. López, Á. Caraguay, Lv, & hernández-álvarez, m.(2019). a survey on situational awareness of ransomware attacks—detection and prevention parameters, Remote Sensing 11 (10).

[37] X. Sun, H. Wang, J. Li, Y. Zhang, Injecting purpose and trust into data anonymisation, Computers & security 30 (5) (2011) 332–345.

[38] X. Sun, M. Li, H. Wang, A family of enhanced (l, α)-diversity models for privacy preserving data publishing, Future Generation Computer Systems 27 (3) (2011) 348–356.

[39] M. Cano, J. Jeimy, Managing uncertainty and complexity: Emerging challenges in cyber security, in: World Conference on Information Systems and Technologies, Springer, 2022, pp. 192–203.

[40] H. Wang, Y. Zhang, J. Cao, Effective collaboration with information sharing in virtual universities, IEEE Transactions on Knowledge and Data Engineering 21 (6) (2008) 840–853.

[41] H. Wang, J. Cao, Y. Zhang, A flexible payment scheme and its role-based access control, IEEE Transactions on knowledge and Data Engineering 17 (3) (2005) 425–436.

[42] H. Wang, Y. Zhang, J. Cao, V. Varadharajan, Achieving secure and flexible m-services through tickets, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 33 (6) (2003) 697–708.

[43] J. Ma, L. Sun, H. Wang, Y. Zhang, U. Aickelin, Supervised anomaly detection in uncertain pseudoperiodic data streams, ACM Transactions on Internet Technology (TOIT) 16 (1) (2016) 1–20.

[44] A. El-Kosairy, M. A. Azer, Intrusion and ransomware detection system, in: 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), IEEE, 2018, pp. 1–7.

[45] A. Javaid, Q. Niyaz, W. Sun, M. Alam, A deep learning approach for network intrusion detection system, Eai Endorsed Transactions on Security and Safety 3 (9) (2016) e2.

[46] S.-H. Choi, D.-H. Hwang, Y.-H. Choi, Wireless intrusion prevention system using dynamic random forest against wireless mac spoofing attack, in: 2017 IEEE Conference on Dependable and Secure Computing, IEEE, 2017, pp. 131–137.

[47] A. Youssef, A. Emam, Network intrusion detection using data mining and network behaviour analysis, International journal of computer science & information technology 3 (6) (2011) 87.

[48] M. Sun, M. Zheng, J. C. Lui, X. Jiang, Design and implementation of an android host-based intrusion prevention system, in: Proceedings of the 30th annual computer security applications conference, 2014, pp. 226–235.

[49] M. E. Kabir, H. Wang, Y. Zhang, A pairwise-systematic microaggregation for statistical disclosure control, in: 2010 IEEE International Conference on Data Mining, IEEE, 2010, pp. 266–273.

[50] M. Li, X. Sun, H. Wang, Y. Zhang, Multi-level delegations with trust management in access control systems, Journal of Intelligent Information Systems 39 (3) (2012) 611–626.

[51] J. R. Vacca, Network and system security, Elsevier, 2013.

[52] J. Yin, M. Tang, J. Cao, H. Wang, M. You, Y. Lin, Vulnerability exploitation time prediction: an integrated framework for dynamic imbalanced learning, World Wide Web (2021) 1–23.

[53] R. ur Rasool, K. Ahmed, Z. Anwar, H. Wang, U. Ashraf, W. Rafique, Cyberpulse++: A machine learning-based security framework for detecting link flooding attacks in software defined networks, International Journal of Intelligent Systems 36 (2021) 3852 – 3879.

[54] M. Li, X. Sun, H. Wang, Y. Zhang, J. Zhang, Privacy-aware access control with trust management in web service, World Wide Web 14 (4) (2011) 407–430.

[55] R. ur Rasool, H. Wang, U. Ashraf, K. Ahmed, Z. Anwar, W. Rafique, A survey of link flooding attacks in software defined network ecosystems, Journal of Network and Computer Applications (2020) 102803doi:https://doi.org/10.1016/j.jnca.2020.102803.

[56] J. Shu, X. Jia, K. Yang, H. Wang, Privacy-preserving task recommendation services for crowdsourcing, IEEE Transactions on Services Computing 14 (1) (2018) 235–247.

[57] L.-X. Yang, X. Yang, Q. Zhu, L. Wen, A computer virus model with graded cure rates, Nonlinear Analysis: Real World Applications 14 (1) (2013) 414–422.

[58] O. M. Radostits, I. R. Littlejohns, New concepts in the pathogenesis, diagnosis and control of diseases caused by the bovine viral diarrhea virus, The Canadian Veterinary Journal 29 (6) (1988) 513.

[59] A. Makkar, U. Ghosh, P. K. Sharma, A. Javed, A fuzzy-based approach to enhance cyber defence security for next-generation iot, IEEE Internet of Things Journal (2021).

[60] A. Alzahrani, Coronavirus social engineering attacks: Issues and recommendations, International Journal of Advanced Computer Science and Applications 11 (5) (2020).

[61] A. Makkar, N. Kumar, Protector: an optimized deep learning-based framework for image spam detection and prevention, Future Generation Computer Systems 125 (2021) 41–58.

[62] G. L. Nguyen, B. Dumba, Q.-D. Ngo, H.-V. Le, T. N. Nguyen, A collaborative approach to early detection of iot botnet, Computers & Electrical Engineering 97 (2022) 107525.

[63] A. Makkar, J. H. Park, Securecps: Cognitive inspired framework for detection of cyber attacks in cyber–physical systems, Information Processing & Management 59 (3) (2022) 102914.

[64] H. Kaur, S. Behal, K. Kumar, Characterization and comparison of distributed denial of service attack tools, in: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), IEEE, 2015, pp. 1139–1145.

[65] X. Sun, L. Sun, H. Wang, Extended k-anonymity models against sensitive attribute disclosure, Computer Communications 34 (4) (2011) 526–535.

[66] X. Sun, H. Wang, J. Li, T. M. Truta, Enhanced p-sensitive k-anonymity models for privacy preserving data publishing, Transactions on Data Privacy 1 (2) (2008) 53–66.

[67] Y. Zhang, Y. Shen, H. Wang, Y. Zhang, X. Jiang, On secure wireless communications for service oriented computing, IEEE Transactions on Services Computing 11 (2) (2015) 318–328.

[68] A. R. Chordiya, S. Majumder, A. Y. Javaid, Man-in-the-middle (mitm) attack based hijacking of http traffic using open source tools, in: 2018 IEEE International Conference on Electro/Information Technology (EIT), IEEE, 2018, pp. 0438–0443.

[69] M. S. Kumar, J. Ben-Othman, K. Srinivasagan, An investigation on wannacry ransomware and its detection, in: 2018 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2018, pp. 1–6.

[70] B. Kuang, A. Fu, W. Susilo, S. Yu, Y. Gao, A survey of remote attestation in internet of things: Attacks, countermeasures, and prospects, Computers & Security 112 (2022) 102498.

[71] F. Q. Kareem, S. Y. Ameen, A. A. Salih, D. M. Ahmed, S. F. Kak, H. M. Yasin, I. M. Ibrahim, A. M. Ahmed, Z. N. Rashid, N. Omar, Sql injection attacks prevention system technology, Asian Journal of Research in Computer Science 6 (15) (2021) 13–32.

[72] M. Fichtenkamm, G. F. Burch, J. Burch, Isaca journal cybersecurity in a covid-19 world: Insights on how decisions are made (2022).

[73] A. Makkar, T. W. Kim, A. K. Singh, J. Kang, J. H. Park, Secureiiot environment: Federated learning empowered approach for securing iiot from data breach, IEEE Transactions on Industrial Informatics (2022).

[74] A. Ahmad, J. Webb, K. C. Desouza, J. Boorman, Strategically-motivated advanced persistent threat: Definition, process, tactics and a disinformation model of counterattack, Computers & Security 86 (2019) 402–418.

[75] A. Makkar, Secureengine: Spammer classification in cyber defence for leveraging green computing in sustainable city, Sustainable Cities and Society 79 (2022) 103658.

[76] N. Kumar, A. Makkar, Machine learning in cognitive IoT, CRC Press, 2020.

[77] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani, A detailed analysis of the kdd cup 99 data set, in: 2009 IEEE symposium on computational intelligence for security and defense applications, Ieee, 2009, pp. 1–6.

[78] L. Dhanabal, S. Shantharajah, A study on nsl-kdd dataset for intrusion detection system based on classification algorithms, International journal of advanced research in computer and communication engineering 4 (6) (2015) 446–452.

[79] Y.-H. Zhang, Y.-J. Gong, Y. Gao, H. Wang, J. Zhang, Parameter-free voronoi neighborhood for evolutionary multimodal optimization, IEEE Transactions on Evolutionary Computation 24 (2) (2020) 335–349. doi:10.1109/TEVC.2019.2921830.

[80] Z. Zhang, L. Teng, M. Zhou, J. Wang, H. Wang, Enhanced branch-and-bound framework for a class of sequencing problems, IEEE Transactions on Systems, Man, and Cybernetics: Systems 51 (5) (2021) 2726–2736. doi:10.1109/TSMC.2019.2916202.

[81] Z.-J. Wang, Z.-H. Zhan, Y. Lin, W.-J. Yu, H. Wang, S. Kwong, J. Zhang, Automatic niching differential evolution with contour prediction approach for multimodal optimization problems, IEEE Transactions on Evolutionary Computation 24 (1) (2019) 114–128.

[82] Y.-F. Ge, W.-J. Yu, J. Cao, H. Wang, Z.-H. Zhan, Y. Zhang, J. Zhang, Distributed memetic algorithm for outsourced database fragmentation, IEEE Transactions on Cybernetics 51 (10) (2021) 4808–4821. doi:10.1109/TCYB.2020.3027962.

[83] F. Liu, X. Zhou, J. Cao, Z. Wang, T. Wang, H. Wang, Y. Zhang, Anomaly detection in quasi-periodic time series based on automatic data segmentation and attentional lstm-cnn, IEEE Transactions on Knowledge and Data Engineering 34 (6) (2022) 2626–2640. doi:10.1109/TKDE.2020.3014806.

[84] J.-Y. Li, Z.-H. Zhan, H. Wang, J. Zhang, Data-driven evolutionary algorithm with perturbation-based ensemble surrogates, IEEE Transactions on Cybernetics 51 (8) (2021) 3925–3937. doi:10.1109/TCYB.2020.3008280.

[85] T. Huang, Y.-J. Gong, W.-N. Chen, H. Wang, J. Zhang, A probabilistic niching evolutionary computation framework based on binary space partitioning, IEEE Transactions on Cybernetics 52 (1) (2022) 51–64. doi:10.1109/TCYB.2020.2972907.

[86] W.-L. Liu, Y.-J. Gong, W.-N. Chen, Z. Liu, H. Wang, J. Zhang, Coordinated charging scheduling of electric vehicles: A mixed-variable differential evolution approach, IEEE Transactions on Intelligent Transportation Systems 21 (12) (2019) 5094–5109.

[87] T. Huang, Y.-J. Gong, S. Kwong, H. Wang, J. Zhang, A niching memetic algorithm for multi-solution traveling salesman problem, IEEE Transactions on Evolutionary Computation 24 (3) (2019) 508–522.

[88] M. Peng, W. Gao, H. Wang, Y. Zhang, J. Huang, Q. Xie, G. Hu, G. Tian, Parallelization of massive textstream compression based on compressed sensing, ACM Transactions on Information Systems (TOIS) 36 (2) (2017) 1–18.

[89] Q. Xie, J. Huang, M. Peng, Y. Zhang, K. Peng, H. Wang, Discriminative regularized deep generative models for semi-supervised learning, in: 2019 IEEE International Conference on Data Mining (ICDM), IEEE, 2019, pp. 658–667.

[90] Y.-F. Ge, J. Cao, H. Wang, Z. Chen, Y. Zhang, Set-based adaptive distributed differential evolution for anonymity-driven database fragmentation, Data Science and Engineering (2021) 1–12.

[91] J. Huang, M. Peng, H. Wang, J. Cao, W. Gao, X. Zhang, A probabilistic method for emerging topic tracking in microblog stream, World Wide Web 20 (2) (2017) 325–350.

[92] M. Peng, G. Zeng, Z. Sun, J. Huang, H. Wang, G. Tian, Personalized app recommendation based on app permissions, World Wide Web 21 (1) (2018) 89–104.

[93] I. Sharafaldin, A. Gharib, A. H. Lashkari, A. A. Ghorbani, Towards a reliable intrusion detection benchmark dataset, Software Networking 2018 (1) (2018) 177–200.

[94] M. H. Bhuyan, D. Bhattacharyya, J. K. Kalita, An empirical evaluation of information metrics for low-rate and high-rate ddos attack detection, Pattern Recognition Letters 51 (2015) 1–7.

[95] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, H. Janicke, Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study, Journal of Information Security and Applications 50 (2020) 102419.

[96] S. T. Brugger, J. Chow, An assessment of the darpa ids evaluation dataset using snort, UCDAVIS department of Computer Science 1 (2007) (2007) 22.

[97] J. L. Leevy, T. M. Khoshgoftaar, A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data, Journal of Big Data 7 (1) (2020) 1–19.

[98] J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, K. Nakao, Statistical analysis of honeypot data and building of kyoto 2006+ dataset for nids evaluation, in: Proceedings of the first workshop on building analysis datasets and gathering experience returns for security, 2011, pp. 29–36.

[99] A. Shiravi, H. Shiravi, M. Tavallaee, A. A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detection, computers & security 31 (3) (2012) 357–374.

[100] N. Moustafa, J. Slay, Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set), in: 2015 military communications and information systems conference (MilCIS), IEEE, 2015, pp. 1–6.

[101] Z. Dewa, L. A. Maglaras, Data mining and intrusion detection systems, International Journal of Advanced Computer Science and Applications 7 (1) (2016).

[102] J. Zhang, H. Wang, X. Tao, L. Sun, Sodit: An innovative system for outlier detection using multiple localized thresholding and interactive feedback, in: 2013 IEEE 29th International Conference on Data Engineering (ICDE), IEEE, 2013, pp. 1364–1367.

[103] Y.-F. Ge, J. Cao, H. Wang, J. Yin, W.-J. Yu, Z.-H. Zhan, J. Zhang, A benefit-driven genetic algorithm for balancing privacy and utility in database fragmentation, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2019, pp. 771–776.

[104] Y.-F. Ge, M. Orlowska, J. Cao, H. Wang, Y. Zhang, Knowledge transfer-based distributed differential evolution for dynamic database fragmentation, Knowledge-Based Systems (2021) 107325doi:https://doi.org/10.1016/j.knosys.2021.107325.

[105] Y.-F. Ge, Z.-H. Zhan, J. Cao, H. Wang, Y. Zhang, K.-K. Lai, J. Zhang, Dsga: A distributed segment-based genetic algorithm for multi-objective outsourced database partitioning, Information Sciences 612 (2022) 864–886. doi:10.1016/j.ins.2022.09.003.

[106] R. ur Rasool, M. Najam, H. F. Ahmad, H. Wang, Z. Anwar, A novel json based regular expression language for pattern matching in the internet of things, Journal of Ambient Intelligence and Humanized Computing 10 (4) (2019) 1463–1481.

[107] Y.-F. Ge, M. Orlowska, J. Cao, H. Wang, Y. Zhang, Mdde: multitasking distributed differential evolution for privacy-preserving database fragmentation, The VLDB Journal (2022) 1–19doi:10.1007/s00778-021-00718-w.

[108] M. E. Kabir, H. Wang, Microdata protection method through microaggregation: a systematic approach, Journal of Software 7 (11) (2012) 2415–2422.

[109] W. Shi, W.-N. Chen, S. Kwong, J. Zhang, H. Wang, T. Gu, H. Yuan, J. Zhang, A coevolutionary estimation of distribution algorithm for group insurance portfolio, IEEE Transactions on Systems, Man, and Cybernetics: Systems (10.1109/TSMC.2021.3096013) (2021) 1–15.

[110] J.-Y. Li, K.-J. Du, Z.-H. Zhan, H. Wang, J. Zhang, Distributed differential evolution with adaptive resource allocation, IEEE Transactions on Cybernetics (2022) 1–14doi:10.1109/TCYB.2022.3153964.

[111] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, M. Ghogho, Deep learning approach for network intrusion detection in software defined networking, in: 2016 international conference on wireless networks and mobile communications (WINCOM), IEEE, 2016, pp. 258–263.

[112] S. Potluri, C. Diedrich, Accelerated deep neural networks for enhanced intrusion detection system, in: 2016 IEEE 21st international conference on emerging technologies and factory automation (ETFA), IEEE, 2016, pp. 1–8.

[113] M.-J. Kang, J.-W. Kang, Intrusion detection system using deep neural network for in-vehicle network security, PloS one 11 (6) (2016) e0155781.

[114] L. Zhou, X. Ouyang, H. Ying, L. Han, Y. Cheng, T. Zhang, Cyber-attack classification in smart grid via deep neural network, in: Proceedings of the 2nd international conference on computer science and application engineering, 2018, pp. 1–5.

[115] F. Feng, X. Liu, B. Yong, R. Zhou, Q. Zhou, Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device, Ad Hoc Networks 84 (2019) 82–89.

[116] H. Zhang, X. Yu, P. Ren, C. Luo, G. Min, Deep adversarial learning in intrusion detection: A data augmentation enhanced framework, arXiv preprint arXiv:1901.07949 (2019).

[117] A. Darem, J. Abawajy, A. Makkar, A. Alhashmi, S. Alanazi, Visualization and deep-learning-based malware variant detection using opcode-level features, Future Generation Computer Systems 125 (2021) 314–323.

[118] L. Zhang, L. Shi, N. Kaja, D. Ma, A two-stage deep learning approach for can intrusion detection, in: Proc. Ground Vehicle Syst. Eng. Technol. Symp.(GVSETS), 2018, pp. 1–11.

[119] J. Kim, J. Kim, H. L. T. Thu, H. Kim, Long short term memory recurrent neural network classifier for intrusion detection, in: 2016 international conference on platform technology and service (PlatCon), IEEE, 2016, pp. 1–5.

[120] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, D. Gan, Cloud-based cyber-physical intrusion detection for vehicles using deep learning, Ieee Access 6 (2017) 3491–3508.

[121] A. Taylor, S. Leblanc, N. Japkowicz, Anomaly detection in automobile control network data with long short-term memory networks, in: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2016, pp. 130–139.

[122] Y. Zhang, X. Chen, L. Jin, X. Wang, D. Guo, Network intrusion detection: Based on deep hierarchical network and original flow data, IEEE Access 7 (2019) 37004–37016.

[123] S. Basumallik, R. Ma, S. Eftekharnejad, Packet-data anomaly detection in pmu-based state estimator using convolutional neural network, International Journal of Electrical Power & Energy Systems 107 (2019) 690–702.

[124] K. Fu, D. Cheng, Y. Tu, L. Zhang, Credit card fraud detection using convolutional neural networks, in: International conference on neural information processing, Springer, 2016, pp. 483–490.

[125] Z. Zhang, X. Zhou, X. Zhang, L. Wang, P. Wang, A model based on convolutional neural network for online transaction fraud detection, Security and Communication Networks 2018 (2018).

[126] M. Nasr, A. Bahramali, A. Houmansadr, Deepcorr: Strong flow correlation attacks on tor using deep learning, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 1962–1976.

[127] Y. Zeng, H. Gu, W. Wei, Y. Guo, $deep - full - range$: a deep learning based network encrypted traffic classification and intrusion detection framework, IEEE Access 7 (2019) 45182–45190.

[128] Y. Yu, J. Long, Z. Cai, Network intrusion detection through stacking dilated convolutional autoencoders, Security and Communication Networks 2017 (2017).

[129] N. Shone, T. N. Ngoc, V. D. Phai, Q. Shi, A deep learning approach to network intrusion detection, IEEE transactions on emerging topics in computational intelligence 2 (1) (2018) 41–50.

[130] F. A. Khan, A. Gumaei, A. Derhab, A. Hussain, A novel two-stage deep learning model for efficient network intrusion detection, IEEE Access 7 (2019) 30373–30385.

[131] D. Papamartzivanos, F. G. Mármol, G. Kambourakis, Introducing deep learning self-adaptive misuse network intrusion detection systems, IEEE Access 7 (2019) 13546–13560.

[132] Y. Yang, K. Zheng, C. Wu, Y. Yang, Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network, Sensors 19 (11) (2019) 2528.

[133] U. Fiore, F. Palmieri, A. Castiglione, A. De Santis, Network anomaly detection with the restricted boltzmann machine, Neurocomputing 122 (2013) 13–23.

[134] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, A. E. Hassanien, Hybrid intelligent intrusion detection scheme, in: Soft computing in industrial applications, Springer, 2011, pp. 293–303.

[135] T. Aldwairi, D. Perera, M. A. Novotny, An evaluation of the performance of restricted boltzmann machines as a model for anomaly network intrusion detection, Computer Networks 144 (2018) 111–119.

[136] N. Gao, L. Gao, Q. Gao, H. Wang, An intrusion detection model based on deep belief networks, in: 2014 Second international conference on advanced cloud and big data, IEEE, 2014, pp. 247–252.

[137] M. Z. Alom, V. Bontupalli, T. M. Taha, Intrusion detection using deep belief networks, in: 2015 National Aerospace and Electronics Conference (NAECON), IEEE, 2015, pp. 339–344.

[138] S. Otoum, B. Kantarci, H. Mouftah, A comparative study of ai-based intrusion detection techniques in critical infrastructures, ACM Transactions on Internet Technology (TOIT) 21 (4) (2021) 1–22.

[139] H. Wang, L. Sun, Trust-involved access control in collaborative open social networks, in: 2010 fourth international conference on network and system security, IEEE, 2010, pp. 239–246.

[140] M. Aloqaily, S. Otoum, I. Al Ridhawi, Y. Jararweh, An intrusion detection system for connected vehicles in smart cities, Ad Hoc Networks 90 (2019) 101842.

[141] F. Khalil, J. Li, H. Wang, A framework of combining markov model with association rules for predicting web page accesses, in: Proceedings of the 5th Australasian Data Mining Conference (AusDM 2006): Data Mining and Analytics 2006, ACS Press, 2006, pp. 177–184.

[142] H. Karimipour, A. Dehghantanha, R. M. Parizi, K.-K. R. Choo, H. Leung, A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids, IEEE Access 7 (2019) 80778–80788.

[143] G. Thamilarasu, S. Chawla, Towards deep-learning-driven intrusion detection for the internet of things, Sensors 19 (9) (2019) 1977.

[144] G. Zhao, C. Zhang, L. Zheng, Intrusion detection using deep belief network and probabilistic neural network, in: 2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC), Vol. 1, IEEE, 2017, pp. 639–642.

[145] Y. Zhang, P. Li, X. Wang, Intrusion detection for iot based on improved genetic algorithm and deep belief network, IEEE Access 7 (2019) 31711–31722.

[146] S. M. Kasongo, Y. Sun, A deep learning method with filter based feature engineering for wireless intrusion detection system, IEEE access 7 (2019) 38597–38607.

[147] S. Sengan, V. Subramaniyaswamy, V. Indragandhi, P. Velayutham, L. Ravi, Detection of false data cyber-attacks for the assessment of security in smart grid using deep learning, Computers & Electrical Engineering 93 (2021) 107211.

[148] G. W. Taylor, G. E. Hinton, S. Roweis, Modeling human motion using binary latent variables, Advances in neural information processing systems 19 (2006).

[149] C. G. Cordero, S. Hauke, M. Mühlhäuser, M. Fischer, Analyzing flow-based anomaly intrusion detection using replicator neural networks, in: 2016 14th Annual Conference on Privacy, Security and Trust (PST), IEEE, 2016, pp. 317–324.

[150] D. Li, L. Deng, M. Lee, H. Wang, Iot data feature extraction and intrusion detection system for smart cities based on deep migration learning, International journal of information management 49 (2019) 533–545.

[151] S. Subramani, H. Wang, H. Q. Vu, G. Li, Domestic violence crisis identification from facebook posts based on deep learning, IEEE access 6 (2018) 54075–54085.

[152] H. Hu, J. Li, H. Wang, G. Daggard, Combined gene selection methods for microarray data analysis, in: International conference on knowledge-based and intelligent information and engineering systems, Springer, 2006, pp. 976–983.

[153] R. Singh, Y. Zhang, H. Wang, Y. Miao, K. Ahmed, Investigation of social behaviour patterns using location-based data – a melbourne case study, ICST Transactions on Scalable Information Systems 8 (2020) 166767. doi:10.4108/eai.26-10-2020.166767.

[154] F. Khalil, H. Wang, J. Li, Integrating markov model with clustering for predicting web page accesses, in: Proceeding of the 13th Australasian world wide web conference (AusWeb07), AusWeb, 2007, pp. 63–74.

[155] R. Sarki, K. Ahmed, H. Wang, Y. Zhang, K. Wang, Convolutional neural network for multi-class classification of diabetic eye disease, EAI Endorsed Transactions on Scalable Information Systems 9 (4) (12 2021). doi:10.4108/eai.16-12-2021.172436.

[156] D. Pandey, H. Wang, X. Yin, K. Wang, Y. Zhang, J. Shen, Automatic breast lesion segmentation in phase preserved dce-mris, Health Information Science and Systems 10 (05 2022). doi:10.1007/s13755-022-00176-w.

[157] X. Sun, H. Wang, J. Li, J. Pei, Publishing anonymous survey rating data, Data Mining and Knowledge Discovery 23 (3) (2011) 379–406.

[158] X. Sun, M. Li, H. Wang, A. Plank, An efficient hash-based algorithm for minimal k-anonymity, in: Conferences in Research and Practice in Information Technology (CRPIT), Vol. 74, Australian Computer Society Inc., 2008, pp. 101–107.

[159] J. Huang, M. Peng, H. Wang, Topic detection from large scale of microblog stream with high utility pattern clustering, in: Proceedings of the 8th Workshop on Ph. D. Workshop in Information and Knowledge Management, 2015, pp. 3–10.

[160] K. Cheng, L. Wang, Y. Shen, H. Wang, Y. Wang, X. Jiang, H. Zhong, Secure k-nn query on encrypted cloud data with multiple keys, IEEE Transactions on Big Data 7 (4) (2017) 689–702.

[161] H. Li, Y. Wang, H. Wang, B. Zhou, Multi-window based ensemble learning for classification of imbalanced streaming data, World Wide Web 20 (6) (2017) 1507–1525.

[162] F. Khalil, H. Wang, J. Li, Integrating markov model with clustering for predicting web page accesses, in: Proceeding of the 13th Australasian world wide web conference, 2007, pp. 63–74.

[163] H. Wang, L. Sun, E. Bertino, Building access control policy model for privacy preserving and testing policy conflicting problems, Journal of Computer and System Sciences 80 (8) (2014) 1493–1503.

[164] G. Bargshady, X. Zhou, R. C. Deo, J. Soar, F. Whittaker, H. Wang, Enhanced deep learning algorithm development to detect pain intensity from facial expression images, Expert Systems with Applications 149 (2020) 113305.

[165] Z.-G. Chen, Z.-H. Zhan, H. Wang, J. Zhang, Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems, IEEE transactions on evolutionary computation 24 (4) (2019) 708–719.

[166] D. Pandey, X. Yin, H. Wang, Y. Zhang, Accurate vessel segmentation using maximum entropy incorporating line detection and phase-preserving denoising, Computer Vision and Image Understanding 155 (2017) 162–172.

[167] W. Gao, M. Peng, H. Wang, Y. Zhang, Q. Xie, G. Tian, Incorporating word embeddings into topic modeling of short text, Knowledge and Information Systems 61 (2) (2019) 1123–1145.

[168] M. Peng, J. Zhu, H. Wang, X. Li, Y. Zhang, X. Zhang, G. Tian, Mining event-oriented topics in microblog stream with unsupervised multi-view hierarchical embedding, ACM Transactions on Knowledge Discovery from Data (TKDD) 12 (3) (2018) 1–26.

[169] J. Zhang, H. Li, X. Liu, Y. Luo, F. Chen, H. Wang, L. Chang, On efficient and robust anonymization for privacy protection on massive streaming categorical information, IEEE Transactions on Dependable and Secure Computing 14 (5) (2015) 507–520.

[170] M. E. Kabir, A. N. Mahmood, H. Wang, A. K. Mustafa, Microaggregation sorting framework for k-anonymity statistical disclosure control in cloud computing, IEEE Transactions on Cloud Computing 8 (2) (2015) 408–417.

[171] Y. Wang, Z.-O. Wang, A fast knn algorithm for text categorization, in: 2007 international conference on machine learning and cybernetics, Vol. 6, IEEE, 2007, pp. 3436–3441.

[172] D. Lee, H. S. Seung, Algorithms for non-negative matrix factorization, Advances in neural information processing systems 13 (2000).

[173] R. U. Rasool, U. Ashraf, K. Ahmed, H. Wang, W. Rafique, Z. Anwar, Cyberpulse: a machine learning based link flooding attack mitigation system for software defined networks, IEEE Access 7 (2019) 34885–34899.

[174] Y. Wang, Y. Shen, H. Wang, J. Cao, X. Jiang, Mtmr: Ensuring mapreduce computation integrity with merkle tree-based verifications, IEEE Transactions on Big Data 4 (3) (2016) 418–431.

[175] L. Sun, J. Ma, H. Wang, Y. Zhang, J. Yong, Cloud service description model: an extension of usdl for cloud services, IEEE Transactions on Services Computing 11 (2) (2015) 354–368.

[176] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (10) (2009) 1345–1359.

[177] W. G. Hatcher, W. Yu, A survey of deep learning: platforms, applications and emerging research trends, IEEE Access 6 (2018) 24411–24432.

[178] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[179] P. Skocir, P. Krivic, M. Tomeljak, M. Kusek, G. Jezic, Activity detection in smart home environment., in: KES, 2016, pp. 672–681.

[180] Y. Li, M.-C. Chang, S. Lyu, In ictu oculi: Exposing ai created fake videos by detecting eye blinking, in: 2018 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2018, pp. 1–7.

[181] A. Dash, J. C. B. Gamboa, S. Ahmed, M. Liwicki, M. Z. Afzal, Tac-gan-text conditioned auxiliary classifier generative adversarial network, arXiv preprint arXiv:1703.06412 (2017).

[182] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, D. Krishnan, Unsupervised pixel-level domain adaptation with generative adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3722–3731.

[183] A. Creswell, A. A. Bharath, Inverting the generator of a generative adversarial network, IEEE transactions on neural networks and learning systems 30 (7) (2018) 1967–1974.

[184] X. Sun, H. Wang, J. Li, Y. Zhang, Satisfying privacy requirements before data anonymization, The Computer Journal 55 (2012) 422–437. doi:10.1093/comjnl/bxr028.

[185] E. Kabir, H. Wang, Conditional purpose based access control model for privacy protection, in: Proceedings of the Twentieth Australasian Conference on Australasian Database, Vol. 92, 2009, pp. 137–144.

[186] S. Chenthara, K. Ahmed, H. Wang, F. Whittaker, Z. Chen, Healthchain: A novel framework on privacy preservation of electronic health records using blockchain technology, Plos one 15 (12) (2020) e0243043.

[187] E. Kabir, A role-involved purpose-based access control model, Information Systems Frontiers 14 (2012) 809–822.

[188] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, L. Chang, Principal component-based anomaly detection scheme, in: Foundations and novel approaches in data mining, Springer, 2006, pp. 311–329.

[189] S. Revathi, A. Malathi, A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection, International Journal of Engineering Research & Technology (IJERT) 2 (12) (2013) 1848–1853.

[190] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, K. Han, Enhanced network anomaly detection based on deep neural networks, IEEE access 6 (2018) 48231–48246.

[191] X. Wang, S. Yin, H. Li, J. Wang, L. Teng, A network intrusion detection method based on deep multi-scale convolutional neural network, International Journal of Wireless Information Networks 27 (4) (2020) 503–517.