

Electronic Thesis and Dissertation Repository

8-23-2023 10:30 AM

Global Cyber Attack Forecast using AI Techniques

Nusrat Kabir Samia, *The University of Western Ontario*

Supervisor: Haque, Anwar, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in
Computer Science

© Nusrat Kabir Samia 2023

Follow this and additional works at: <https://ir.lib.uwo.ca/etd>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Samia, Nusrat Kabir, "Global Cyber Attack Forecast using AI Techniques" (2023). *Electronic Thesis and Dissertation Repository*. 9582.

<https://ir.lib.uwo.ca/etd/9582>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlsadmin@uwo.ca.

Abstract

The advancement of internet technology and growing involvement in the cyber world have made us prone to cyber-attacks inducing severe damage to individuals and organizations, including financial loss, identity theft, and reputational damage. The rapid emergence and evolution of new networks and new opportunities for businesses and technologies are increasing threats to security vulnerabilities. Hence cyber-crime analysis is one of the wide range applications of Data Mining that can be eventually used to predict and detect crime. However, there are several constraints while analyzing cyber-attacks, which are yet to be resolved for more accurate cyber security inspection.

Although there are many strategies for intrusion detection, predicting upcoming cyber threats remains an open research challenge. Hence, this thesis seeks to utilize temporal correlations among attack frequencies within specific time periods to predict the future severity of cyber incidents. The research aims to address the current research limitations by introducing a real-time data collection framework that will provide up-to-date cyber-attack data. Furthermore, a platform for cyber-attack trend analysis has been developed using Power BI to provide insight into the current cyber-attack trend. A correlation was identified in the reported attack volume across consecutive time frames through collected attack data analysis. This thesis introduces a predictive model that forecasts the frequency of cyber-attacks within a specified time window, using solely a historical record of attack counts. The research includes various machine learning and deep learning methods to develop a prediction system based on multiple time frames with an over 15% improvement in accuracy compared to the conventional baseline model. Namely, our research demonstrates that cyber incidents are not entirely random, and by analyzing patterns and trends in past incidents, developed AI techniques can be used to improve cybersecurity measures and prevent future attacks.

Keywords

Web Scrapping, Selenium, Categorical data, Machine Learning, Deep Learning, Power BI, Time Series, Attack Prediction

Summary for Lay Audience

Increasing worldwide connectivity of networks has exacerbated the risk of cyber-attacks, and due to the nature of the networks, many organizations are prone to being robbed of their confidential data by cybercriminals. With the help of machine learning, cybersecurity systems can learn from their patterns and prevent attacks that can help cyber security professionals become more proactive in addressing threats.

This thesis develops a framework for real-time data collection to provide cyber-attack analysts with recent data and a platform for cyber-attack trend analysis to visualize the general patterns globally. Furthermore, the research employs various AI techniques to utilize temporal correlations between attack frequencies within specific time frames in order to predict the future severity of cyber incidents by predicting attack volume for a particular time window. Hence, it will also help cyber security analysts prepare for cyber security measures. Namely, our contribution focuses on developing a data collection process, presenting a visual representation of cyber-attack trends, and predicting different cyber-attack intensities by enhancing prediction accuracy to assist in analyzing future cyber-attack severity in order to prevent potential cyber-crimes.

Acknowledgments

I would like to take the opportunity to express my gratitude to Dr. Anwar Haque, my supervisor, for providing me with the opportunity to conduct my research in the field of machine learning and cyber security. He has been a tremendous source of support and encouragement throughout my entire master's program at western university.

I would like to express my sincere gratitude to Sajal Saha, a dedicated and talented PhD student, for his invaluable support and guidance throughout my research journey. I express deep appreciation to Sahil Rehman, a brilliant master's student, for providing valuable assistance in guiding me towards the appropriate research direction. I also want to thank all my professors for guiding me through my other coursework. I would also like to express my gratitude to my father for the unwavering support he has given me during the entire journey of

my graduate studies. Finally, I want to thank the computer science department of Western University for giving me this opportunity to engage myself with such an esteemed institute.

Table of Contents

Abstract	ii
Summary for Lay Audience	iii
Acknowledgments.....	iii
Table of Contents	v
List of Tables	vii
List of Figures	ix
Chapter 1	1
1 Introduction	1
1.1 Problem Statement	1
1.2 Motivation and Objective	2
1.3 Thesis Contribution.....	4
1.4 Thesis Outline	5
Chapter 2.....	6
2 Background	6
2.1 Cyber-attack	6
2.2 Malware	10
2.3 Cyber Security Analysis	12
2.3.1 Time Series Analysis in Cyber Security	13
2.4 Fundamentals of Machine Learning	13
2.5 Fundamentals of Predictive Modeling	14
2.5.1 Statistical Model	14
2.5.2 Machine Learning Model.....	16
2.5.3 Deep Learning Model	21

2.6	Implementation of ML Model for Predictive Analysis.....	23
Chapter 3	26
3	Literature Review.....	26
3.1	Detection-Based Approach for Cyber-Attack Analysis.....	26
3.2	Prediction-Based Approach for Cyber-Attack Analysis.....	29
3.3	Analysis of Current Research Gap.....	31
3.4	Comparative Analysis and Novel Contributions	32
Chapter 4	35
4	Methodology	35
4.1	High-Level System Architecture	35
4.2	Data Collection Framework and Dataset	36
4.2.1	Data Collection Framework Development	36
4.2.2	Collected Dataset Description.....	39
4.2.3	Data Preparation and Exploratory Analysis.....	41
4.3	Model Development.....	44
4.3.1	Dataset Partitioning and Model Training Protocol.....	44
4.3.2	Baseline Model Development: Statistical Model	45
4.3.3	Machine Learning Model Development	49
4.3.4	Deep Learning Model Development.....	51
4.4	Evaluation Criteria	54
4.4.1	Mean Absolute Percentage Error	54
4.4.2	Root Mean Square Error	55
4.5	Power BI Analysis for Cyber-attack Data Trend.....	55

4.6 Development Environment	60
Chapter 5	61
5 Results and Discussion	61
5.1 Performance Analysis of Developed Models	61
5.2 Explanatory Analysis of Overall Results and Findings	68
Chapter 6	69
6 Conclusion	69
6.1 Limitation and Future Work	70
Bibliography	71
Curriculum Vitae	80

List of Tables

Table 1: Comparative Analysis of the Related Works with Our Implementation	34
Table 2: Auto-correlation Values for different lags for Type 1 and Type 2 Attack	44
Table 3: ADF Test Values	46
Table 4: Optimal Order Values for ARIMA	48
Table 5: Hyperparameter Values for Developed XGBoost Models	49
Table 6: Hyperparameter Values for Developed Random Forest Models	50
Table 7: Hyperparameter Values for Developed KNR Models	50
Table 8: Hyperparameter Values for Developed SVR Models	50
Table 9: Architecture of RNN for Type 1 Attack in US ($\tau = 10$)	51

Table 10: Architecture of RNN for Type 2 Attack in US ($\tau = 10$)	52
Table 11: Architecture of RNN for Type 1 and Type 2 Attack in US ($\tau = 15$)	52
Table 12: Architecture of RNN for Type 1 Attack in US ($\tau = 5$) and UK ($\tau = 15$)	52
Table 13: Architecture of RNN for Type 1 Attack in UK ($\tau = 30$).....	52
Table 14: Architecture of DNN for Type 1 Attack in US ($\tau = 10,15$).....	53
Table 15: Architecture of DNN for Type 2 Attack in US ($\tau = 10$) and Type 1 Attack in UK ($\tau = 15$).....	53
Table 16: Architecture of DNN for Type 2 Attack in US ($\tau = 15$).....	54
Table 17: Architecture of DNN for Type 1 Attack in UK ($\tau = 30$)	54
Table 18: Evaluation Scores for Type 1 Attack in US ($\tau = 10$).....	62
Table 19: Evaluation Scores for Type 2 Attack in US ($\tau = 10$).....	62
Table 20: Evaluation Scores for Type 1 Attack in US ($\tau = 15$).....	62
Table 21: Evaluation Scores for Type 2 Attack in US ($\tau = 15$).....	63
Table 22: Evaluation Scores for Type 1 Attack in US ($\tau = 5$).....	63
Table 23: Evaluation Scores for Type 1 Attack in UK ($\tau = 15$)	63
Table 24: Evaluation Scores for Type 1 Attack in UK ($\tau = 30$)	64
Table 25: Actual Value vs Predicted Value Plots for ARIMA Model	65
Table 26: Actual Value vs Predicted Value Plots for Best Performing ML and DL Models.	66
Table 27: Learning Curve for Developed DL Models.....	67

List of Figures

Figure 1: Simple Random Forest [42]	18
Figure 2: Recurrent Neural Network [53].....	22
Figure 3: Deep Neural Network Architecture [54]	23
Figure 4: Global Cyber-attack Pattern Analysis System Architecture	35
Figure 5: Data Collection Framework	38
Figure 6: Dataset.....	40
Figure 7: Different Attack volumes for United States and United Kingdom	42
Figure 8: Auto-correlation Map for different lag Values.....	43
Figure 9: PCAF Plot.....	47
Figure 10: High-level View of Cyber-attack Trend Analysis with Power BI	56
Figure 11: Dashboard for Global Cyber-attack Trend Analysis.....	57
Figure 12: Top 3 and Top 10 Analysis on Current Cyber-attack Data.....	57
Figure 13: Map Visualization for Countries of Origin Launching Attacks on Target Countries	58
Figure 14: Page with Filtering Mechanism for Corresponding Cyber-attack Information before any selection	59
Figure 15: Page with Filtering Mechanism for Corresponding Cyber-attack Information after Destination Country and Time Range Selection.....	59
Figure 16: MAPE vs Autocorr�lation Plot.....	68

Chapter 1

1 Introduction

Due to the rapid advancements in technology, there are now major threats to organizational data and information. [1]. The rise of cybercrime organizations and paramilitary groups has dramatically changed the landscape of the cyber world [2]. For cyber-attackers, obtaining information and systems resources is a vital part of their goal to gain financial and geopolitical advantage. They are also known to carry out attacks on mission-critical systems to disrupt the operations of a company. Cyber-crimes range from breaching information systems, disseminating computer viruses, stealing identity information, stealing political and industrial secrets, spreading misinformation to influencing global opinions, election results, and more. Consequently, cyber security has evolved as a counter mechanism to secure cyberspace from cyber-attack, damage, misuse, and economic espionage [3]. A wide range of techniques ranging from viruses, worms, botnets, ransomware, and cryptocurrency mining viruses are used for cyber-attacks. Cybercriminals start staking their claim in the digital universe, imposing excessive costs on the economy as organizations go digital. With the increasing sophistication of cybercrime in its attack approaches and techniques, the perception of cyber security has shifted from being solely an "IT issue" to a "strategic management issue" and a "techno-legal-management" issue. Namely, cyber security has become a complex domain with the inclusion of emerging technologies; therefore, it requires more intense research to outpace cyber-crime growth. Secure software development; critical infrastructure security; cyberspace – self-defense technologies; cyber satellite security; data piracy, social media platforms – cyber criminology; cyber security and privacy – ethical, technical, policy, and legal issues fall under cyber security research areas [4].

1.1 Problem Statement

According to recent cyber-attack statistics, the costs of corporate cybersecurity are predicted to continue to grow [5]. It has been estimated that 33 billion accounts will be breached in 2023, which would be equivalent to around 2328 incidents every day [6]. The

cost of cybercrime, which is expected to reach \$8 trillion in 2023, has been expected to grow to \$10.5 billion by 2025 [7]. There have been 800,000 cyber-attacks in total, and hacking attacks are carried out every 39 seconds [8]. Amidst growing concern over cybersecurity, the escalation of cybercrime to the position of fifth-highest risk in 2020 has underscored the severity of the threat, a reality further compounded by the widespread adoption of digital technologies across both public and private sectors [7]. These alarming statistics have prompted organizations to prioritize their cybersecurity efforts and develop effective attack prediction strategies.

There are a number of limitations when it comes to recent global cyber-attack prediction frameworks. One limitation is the lack of comprehensive and up-to-date data on cyber-attacks which hinders the development of effective strategies and solutions to combat this pressing issue [9]. Another constraint is the difficulty in predicting global terrorist attacks due to the complexity of the problem [10]. Furthermore, the Global Trends report highlights the challenge of predicting the future accurately, that emphasizes the need for flexibility and adaptability in the face of rapidly changing global trends [11]. Finally, the Global Risks Report 2022 emphasizes the need to consider the interdependence of risks and the potential for cascading effects [12]. Overall, these limitations suggest a need for more research and development to improve the accuracy and reliability of global attack prediction frameworks.

1.2 Motivation and Objective

According to a study conducted by Capgemini, over 60% of organizations believe that they will need to use AI to respond to cyberattacks [13]. An organization can also detect anomalous behavior using machine learning and AI, and by taking advantage of this technology, they can have an understanding of the likelihood of a data breach. Attack prediction frameworks that use AI and ML can help mitigate threats with fewer resources by identifying new attacks and drawing statistical inferences. These systems can help identify and prevent new attacks before they happen. The importance of threat prediction systems lies in their ability to help organizations stay ahead of cyber threats and protect their sensitive data and systems from being compromised. A significant portion of cyber

security research focuses on cyber-attack detection, cyber-attack type prediction, and future security trends in a certain network. If we focus on establishing a system that can forecast the severity of cyber-attacks on a global scale, we will have the opportunity to take preventive measures before the attack occurs, which will assist us to minimize the risk.

The creation of a global alert system for cyberattack severity is one example of a real-world deployment scenario for cyberattack intensity prediction. By indicating the potential attack count for a specific attack type for a given country in the future, this alert system can generate a cyber-attack intensity alert. Using this information, different organizations can take preventive action if they were previously vulnerable to this specific type of attack whenever it occurred on a large scale. The ability to contribute to the field of cyber security has encouraged us to develop our research approach.

As discussed in section 1.1, current research disciplines have a number of limitations that motivate us for our research development in the field of cyber security. Our thesis goals are described below -

- Our primary objective is to address the limitations of current research works and develop techniques to assist cyber-security professionals in assembling insight into cyber-attack trends using machine learning and deep learning algorithms.
- Forecasting events is a crucial part of decision-making in machine learning, and accurate predictions are often difficult to make. Existing research often relies on outdated data, which can lead to inaccurate predictions. We aimed to propose a framework by addressing this limitation of working with obsolete data.
- The complexity of a problem is often a challenge to solve, as it involves various variables that can be difficult to analyze. Our research objective includes considering the complex nature of the problem analysis and developing different techniques to have clear insights from the dataset.
- Furthermore, the thesis aims to provide a comprehensive framework to effectively analyze and predict cyber-attack frequency providing cybersecurity professionals with more accurate and timely insight into global cyber-attack patterns.

1.3 Thesis Contribution

Considering the demand for additional research and development to enhance the accuracy and efficiency of global attack prediction frameworks, this thesis aims to address the following areas -

- We have developed a real-time data collection framework with web scraping to acquire up-to-date data for having a more realistic visualization of cyber-attacks. Our system can collect data in a continuous manner without any interruption for an extended period, and hence our collected data can be leveraged for future investigations into the nature and patterns of cyber-attacks.
- We have developed a platform using Power BI for analyzing threats based on the current data and providing visualizations of attack patterns and trends. This information holds tremendous potential for improving cybersecurity measures and preempting future cyber-attacks.
- We have introduced a novel approach for predicting the intensity of cyber-attacks in a future period depending on counts of previous incidents, which is based on up-to-date data, and our model learning can be easily updated due to the integration of the real-time data collection framework.
- We have performed an exploratory analysis to demonstrate that our collected data is suitable for time series analysis, and we have conducted experiments over varying periods to determine the optimal period length.
- We have explored different data preprocessing and imputation techniques to transform data into a more suitable format for time series analysis with machine learning algorithms to learn efficiently and improve their ability to make accurate predictions. The results show that the performance of the models has significantly improved for effective learning compared to the baseline model.
- We have done comprehensive performance analysis and comparison utilizing the hold-out method and evaluation criteria such as mean absolute percentage error and

root mean square error to identify the optimal model for time series analysis of global cyber-attack data.

1.4 Thesis Outline

The rest of the thesis is organized as follows. Chapter 2 provides a background on cyber-attacks, malware, and machine learning concepts, including an introduction to predictive modeling techniques. Chapter 3 reviews the literature on prediction and detection-based approaches for cyber-attack analysis and compares the related works with our approach, followed by an analysis of the current research gap. Chapter 4 presents the methodology for system architecture, data collection framework and dataset, data preparation, data analysis, model development, and evaluation criteria. Power BI analysis and the development environment are also discussed in this chapter. The results and discussion, including an explanation of the experimental outcomes, are presented in Chapter 5. Finally, Chapter 6 concludes the study with a summary and discussion of the findings, limitations, and future work.

Chapter 2

2 Background

This chapter summarizes the high-level view of background topics relevant to this thesis. Section 2.1 will introduce a detailed exposition of various types of cyber-attacks and their attendant activities. Section 2.2 will provide an in-depth analysis of diverse categories of malware. Section 2.3 will review various aspects of cyber security. Finally, Sections 2.4 and 2.5 will provide an introduction to basic machine learning and predictive modeling concepts, including the implementation techniques.

2.1 Cyber-attack

Cyber-attacks are a growing threat in the modern digital landscape, involving the illegal access of computer networks or systems by one or more individuals. Such attacks can result in the disruption or malfunction of the targeted computer, as well as the theft or compromise of sensitive data. In addition, attackers may exploit breached systems to carry out further attacks. Given the range of tactics available to cybercriminals, it is no surprise that cyber-attacks are becoming increasingly prevalent. This thesis comprises a record of global cyber-attacks in various categories, which in addition includes OWASP (The Open Web Application Security Project) listed cyber-attacks. The Open Web Application Security Project, founded in 2001, is dedicated to providing online users with essential tools and resources to safeguard their websites from malicious attacks, such as viruses and other security threats. Annually, OWASP publishes a list of the top ten most prevalent security flaws in web applications, which are commonly referred to by their OWASP names to signify the associated vulnerabilities. The following is a description of different types of cyber-attacks-

1. **DDoS**: DDoS (Distributed Denial-of-Service) is a type of attack that blocks access to network equipment by saturating the traffic coming from multiple attackers with the goal of disrupting online services [14] [15]. A hacker can easily exploit a wide range of resources, such as computers and IoT devices by creating sudden

unavailability or slowdown of a service or a website which is the most common sign of a distributed denial of service attack. One can easily identify the signs of a DDoS attack, by analyzing the traffic data collected by various tools enabling the identification of common characteristics or patterns indicative of a DDoS attack. Suspicious volumes of incoming traffic from one source or group of sources can indicate a DDoS attack. Furthermore, instant high demands for a single resource or mysterious traffic patterns, such as spikes every ten minutes, can signify a deliberate attack.

2. **API Violation:** API (Application Programming Interface) violation is a cyber-attack that is the malicious or attempted usage of an API by exploiting vulnerabilities in the way an API is designed [16]. This type of attack can lead to unauthorized access to sensitive data or functionality within an application by allowing attackers to use this technique to steal information, manipulate data, or carry out other malicious activities.
3. **RCE / RFI:** RCE (Remote code execution) is a type of cyber-attack that allows an attacker to take control of a target system through unauthorized access by remotely executing malicious code on a computer [17]. Another type of attack is RFI (Remote file inclusion), which targets vulnerabilities in web applications that reference external scripts dynamically by injecting malicious code into a system or an application by including files from an external source [18].
4. **Path Traversal / LFI:** Path Traversal is a cyber-attack that involves manipulating the path of a URL to access files and directories outside of the deliberate scope allowing an attacker to have illicit access to files on a web server[19]. LFI (Local File Intrusion) attack involves accessing the local files of a server by exploiting vulnerabilities in a server application [20].
5. **Automated Threat:** An automated threat is a specific type of security risk that poses a significant danger to both computer systems and web applications by virtue of its ability to initiate an attack in an automated and autonomous manner. The prevalence of online automated attacks is largely due to the fact that they can perform a multitude of repetitive and malicious activities at minimal expense, allowing attackers to cause damage on a large scale with little effort [21]. These

attacks are often used to exploit vulnerabilities in software or to carry out brute-force attacks to gain unauthorized access. This is a process that involves the manipulation of a computer system. Here are some examples of how automated threat cyber-attacks work -

- **Brute Force Attacks:** Brute force attack is usually used to access a computer system or network by trying various passwords. An attacker can use a weak password to access a system that is not secure.
 - **SQL Injection:** Another type of attack is SQL injection, which takes advantage of the database vulnerability allowing an attacker to access the sensitive data of a database. An automated SQL injection attack is very effective because it can rapidly identify many vulnerabilities in web applications.
 - **Phishing:** Phishing is another type of attack which involves tricking people into providing sensitive information. An automated phishing attack can be effective by sending large numbers of phishing messages or emails to potential victims. Due to the nature of the attack, it can be very threatening in a short amount of time.
6. **Automated Threat - Business Logic:** A Business Logic attack is a type of attack that takes advantage of the way an application processes user input. It can potentially have a variety of harmful effects, such as the loss of data or unauthorized access to a user account [22]. By providing malicious software with access to a website's resources, attackers can perform different operations, which would typically be prevented by the site's administrators. These types of attacks can be used to take over weak websites in rare cases. Unlike traditional attacks that target technical vulnerabilities in a system, business logic attacks exploit the way an application is designed to function. This makes them more difficult to detect, as they are often created to mimic legitimate user behavior. Here are some examples of how automated threat business logic attacks work -
- **Account takeover:** This type of attack involves automated tools that attempt to take over user accounts by guessing passwords or using stolen

credentials. Attackers may also use automated bots to bypass two-factor authentication (2FA) mechanisms or to circumvent other security controls.

- **Gift card fraud:** Some automated business logic attacks aim to exploit weaknesses in gift card redemption systems. Attackers use automated tools to generate large numbers of gift card numbers, then use them to purchase high-value items or to transfer funds to other accounts.
- **Credit card fraud:** Like gift card fraud, automated tools can be used to test stolen credit card numbers against an online retailer's payment system. If a card is approved, the attacker can use it to make fraudulent purchases.
- **Fake reviews:** Some automated attacks are intended to create fake reviews or ratings for products or services. These attacks can be used to boost the reputation of a company or damage the reputation of a competitor.

7. **XSS:** XSS (Cross-Site Scripting) is a type of cyber-attack in which malicious scripts are injected into harmless and trusted websites with the ability to access any cookies, session identifiers, or other sensitive information retained by the browser and used with that site [23]. When a web application generates output without validating the user-input, that website becomes vulnerable to a Cross-Site Scripting attack.
8. **SQLi:** SQL Injection (SQLi) is a category of cyber-attack that is initiated by interfering with the queries made from an application to its database [24]. This technique allows an attacker to view, modify or delete any sensitive information by exploiting vulnerabilities in an application's SQL database.
9. **Data Leakage:** Data leakage is a cyber-attack type that occurs through information leakage by unauthorized duplication or transmission of data without affecting source data [25]. However, in some cases, hackers encrypt data to create a denial of access by the data owner leading to a complete loss of data.
10. **Protocol Manipulation:** Protocol Manipulation attack is carried out by exploiting vulnerabilities in the network protocol system [26]. Also, this type of attack can be started by sending malformed messages, which exploits vulnerabilities in protocol vulnerabilities.

11. **Backdoor / Trojan:** The backdoor attack occurs through malicious software programs used to gain complete control over a compromised computer by bypassing the authentication procedures of a system [27]. The trojan attack involves malicious software that opens a backdoor entry to a computer system, disguising itself as an authorized program with the purpose of carrying out malicious activities without the consent of the legitimate user.
12. **File Upload:** Another technique that attacker may utilize is File Upload attacks which is initiated by uploading a malicious file on a database or webserver to gain administrative access [28,29]. If a new file is uploaded with the same extension and name as an existing one, it can be used to overwrite the previous one leading to a harmful server-side attack.
13. **SSRF:** A SSRF (Server-Side Request Forgery) attack is carried out on an application that allows data imports from URLs by altering information to abuse server functionality [29]. Intruders gain access to confidential data by reading the data to the altered URL, including HTTP-enabled databases as well as server configuration data.
14. **Authentication Bypass:** An Authentication Bypass attack exploits system vulnerabilities by altering settings or installing malicious software after successfully providing login credentials, thereby gaining unauthorized access to the targeted system [30]. If an attacker succeeds in compromising a highly privileged account, such as a system administrator, they can control the entire application and gain access to the internal infrastructure.

2.2 Malware

Malware, also known as "malicious software," is an umbrella term that describes any harmful program or code intentionally designed to disrupt a computer, server, or network by gaining unauthorized access. Our compiled dataset includes various types of malwares. These are listed below -

1. **Bot:** A robot, also known as a bot, is a software program that can perform a specific task as part of a computer program. It can be programmed to do so automatically,

eliminating human intervention, and making repetitive tasks easier to complete. When a computer is infected with a malicious bot, it can perform various tasks that allow the attacker to control it. Although these activities can be illegal, they are only considered malicious once they break a website's rules. In most cases, the activities of a bot that engages in unauthorized activities, such as account takeover or identity theft, should be regarded as malicious. When a group of bots launches a distributed denial of service attack (DDoS attack), it can be challenging for the targeted server to identify which one is responsible for the attack.

2. **Browser Automation:** Browser automation malware is a type of malicious software that can automate various actions in a web browser without the user's knowledge or consent. Cybercriminals use automation to speed up and improve the efficiency of their attacks, as automation can send tools to the target and automatically retrieve data and sensitive information.
3. **Hacking Tool:** An application that can be used to break into a computer or network to bypass its security protocols is known as a hacking tool. It is possible to program it to carry out various functions, such as gaining access to a system's internal resources or carrying out other kinds of assaults. These technologies are also available to system administrators for use in the detection of potential dangers.
4. **Vulnerability Scanner:** A vulnerability scanner is a computer program that can be used to detect and analyze various vulnerabilities in a network. It can be used by attackers to locate and analyze systems that are vulnerable. After creating an inventory, the scanner compares the items in it with the known vulnerabilities in a database to find all the systems that are affected.
5. **Masking Proxy:** A proxy hijacking attack is a technique that takes over a genuine web page in search results and index pages. It can be used to gain a competitive edge or redirect users to a malicious website. Through a proxy server, the attacker can access the victim's network traffic and transmit sensitive information to a remote website, and in this case, a man-in-the-middle attack is used.
6. **Worm:** A computer worm is usually created to look for an entry point into a network that can spread through emails or instant messages. Usually, cybercriminals try to trick their victims into running the worm by sending them a

disguised attachment. The attacker uses data names and double file extensions, which seems usually harmless or urgent, to create the illusion of the attachment or link being from a legitimate source. Once the user opens the link or attachment, the computer worm is automatically downloaded and installed onto their system. The worm tries to infiltrate other systems after it is executed. One of its methods is to send an email to all the infected computer contacts with a replica of the worm. Worms can also carry viruses, ransomware, and other harmful programs that can harm the infected system. In the case of extortion, worms can take over the files of the victim and encrypt them. Meanwhile, mixed forms of different malware are frequently used in malware campaigns, for instance, the WannaCry ransomware or the Petya / Not-Petya ransomware. These contain a worm component, which allows the malware to replicate and spread through back doors in other network systems.

2.3 Cyber Security Analysis

Organized crime groups carry out various forms of cyber-attacks and fraud online. The rise of the internet has made it easy for criminals to carry out their crimes, and they are usually involved in white-collar theft. Online stock fraud has been a lucrative venture for thieves causing investors to lose millions of dollars annually. Furthermore, cyber-attacks can have severe consequences on sensitive personal data, often resulting in compromised confidentiality, integrity, and availability of such data. With the advancement of technology, consumers are more reliant on networks and information, and as a result, the risk of being a victim of cybercrime is high. In the field of cyber security analysis, professionals are responsible for analyzing and managing the vast amount of data that is typically acquired and stored by hackers, open-source sources, and individuals. They need to have the necessary theoretical and practical skills to analyze and interpret the data, as efficient management and analysis of such data are essential for identifying and mitigating potential cyber threats. Fundamentally, cyber security analysis is a process that combines risk assessment and vulnerability analysis by helping organizations detect potential threats and improve their security, ensuring the safety and integrity of systems and networks. Time series analysis can help cyber security analysts better understand the impact of past security incidents and use that knowledge to improve their overall security posture.

2.3.1 Time Series Analysis in Cyber Security

Time series analysis focuses on analyzing a dataset that spans many time intervals. It is a popular machine learning technique that cybersecurity professionals can use to prevent breaches and limit data loss by quantitatively fitting data or making predictions to discover anomalies or outliers. It can be used to analyze network data over time to uncover trends and unusual activity to detect potential cyber-attacks [31]. Time-series data is collected by analysts by measuring a characteristic at regular intervals, such as daily, monthly, or yearly. Autocorrelation is a measure of the similarity between a given time series and a lagged version of itself over successive periods, which represents the relationship between the current value of a variable and any past [32] [33]. Time series analysis has the potential to assist cyber security analysts in learning about patterns in data over time, fitting optimal models, and predicting the future. The linear relationship between two observations at various times can be evaluated using the autocorrelation function (ACF) to determine the strength of the correlation between the values in the time series. This concept differs from partial autocorrelation, which isolates the effect of outlying values in a time series and evaluates the correlation between consecutive values.

2.4 Fundamentals of Machine Learning

Machine learning (ML) is a category of AI that uses data collected through past experiences to predict the future. It can be useful in solving problems that are difficult to solve with human brain alone. Through machine learning, computers can efficiently identify patterns in the data they collect. One of the main tasks in machine learning is the creation of algorithms that can analyze and predict data. The algorithms used in machine learning rely on the data collected by the system to make informed decisions and predictions. The datasets are typically divided into multiple sets, which are used in various phases of the development of the model, and they are: Training set, validation set, and test sets.

- **Training Data:** Algorithm learns from the training data to gather experience to make predictions or decisions on new data. The more training data the algorithm

has, the better it can learn and improve its predictions. Each observation is composed of a combination of input and output datasets.

- **Test Data:** A test dataset is composed of observations that are used to perceive how a model might perform in the future. It should not contain any data from the training dataset, as this will make it confounding to tell if the algorithm succeeded in learning through the experience. Performance Metrics are used in the test set to measure the progress of a model.
- **Validation Data:** Another set of observations known as a validation data set is sometimes required during the development of a model. This dataset is used to finetune the variables that the model uses in its learning process.

2.5 Fundamentals of Predictive Modeling

Predictive modeling is a fundamental part of machine learning. This is a structured process that involves the use of statistical and machine learning techniques to develop models capable of forecasting future outcomes based on historical data [34]. The process entails several key steps, including framing the research question, selecting proper data, building and testing candidate models, and evaluating uncertainty. Namely, the predictive modeling technique of data mining looks to the past for insights about the future by analysing data to identify recurring patterns and trends [35]. Predictive modeling, in addition to focusing on the future, can also anticipate outcomes, such as whether or not further inquiry will reveal the transaction to be fraudulent. Machine learning and deep learning models, such as decision trees, boosted regressions, k-nearest neighbors, and deep neural networks, are extensively employed in the modeling process.

We will be providing a brief overview of several predictive modeling techniques which are essential for our thesis experiments.

2.5.1 Statistical Model

A statistical model comprises a set of statistical assumptions that enable the calculation of the probability of any event [36]. This modeling technique consists of a sample space of possible observations and a set of probability distributions. The widely used ARIMA

(Autoregressive Integrated Moving Average) statistical model is a time series analysis tool that takes advantage of data collected over time [37]. It can help predict future trends and obtain insight into a given dataset by integrating the analysis of the relationship between various time-varying variables and a dependent variable. There are three components to the ARIMA model: autoregression (p), integration (d), and a moving average (q). To achieve stationarity in a time series, integration (I) involves differencing raw observations, whereas autoregression (AR) is a model that uses lagging values of a variable to forecast its present value. When applying a moving average (MA) model to historical data, the MA model takes into account the correlation between the residual error and the observation. The parameters p, d, and q define the structure of the ARIMA model and are essential for capturing the basic patterns and dependencies in the time series data. These parameters can be specified as follows:

1. **p (AR order):** p represents the number of lagged observations of the dependent variable that is a symbol indicating the series' linear dependence on its most recent value. The AR coefficient is multiplied by each lagged observation in the model equation.
2. **d (Differencing order):** d represents the number of times a series has to be differenced before it reaches stationarity, hence representing the differencing order. This concept is essential in time series analysis because stationary time series have more predictable patterns and are easier to model.
3. **q (MA order):** The order of the moving average (q) stands for the number of accumulated errors in the model's forecast, which is used as a measure for the interdependence of the error components. The lagged error terms are multiplied in the model equation by their respective coefficients, which are referred to as the MA coefficients.

2.5.2 Machine Learning Model

2.5.2.1 XGBoost

The XGBoost (Extreme Gradient Boosting) algorithm is a machine learning algorithm that uses a gradient-boosted approach to achieve ensemble machine learning. It features advanced regularization techniques that help improve the generalization capabilities of the model. XGBoost is immensely popular in machine learning due to its performance-driven nature. Aside from being useful in machine learning, XGBoost is also accurate and scalable that can boost the performance of tree algorithms by pushing the limits of available computing power. The ensemble learning framework of the algorithm can be used to combine the learning power of multiple models. It can also be used to provide a single model that can aggregate the results of multiple learning systems. XGBoost regression typically includes a regression loss function, such as squared error loss (also known as mean squared error) as its objective function. The objective function is optimized by adding decision trees to the ensemble in an iterative manner, with each tree attempting to minimize the overall loss. The XGBoost feature has a wide range of hyperparameters, which can be tuned to maximize the model's accuracy [38]. The following outlines the essential parameters for analyzing the time series data with the XGB Regression model -

- **Objective:** The objective parameter specifies the loss function that intends to minimize the average error. This parameter is commonly set to "reg:squarederror" for regression analysis.
- **Colsample Bytree:** This parameter specifies the number of columns that will be randomly sampled across all the tree branches.
- **Gamma:** The Gamma parameter is used to specify the minimum loss reduction that needs to be achieved to split the model.
- **Learning Rate:** The learning rate parameter is used to set the shrinkage limit for the steps that are required to prevent overfitting.
- **Max Depth:** The maximum depth of a tree is specified by the Max Depth parameter.

- **Min Child:** The Min Child Weight parameter is used to find the minimum weight that a child should have.
- **N Estimator:** The N estimator is used to specify the number of trees that will be constructed.
- **Reg Lambda:** The Reg lambda parameter specifies the term L2 regularization to reduce the chance of overfitting.
- **Subsample:** The subsample parameter specifies the number of observations that will be considered from each tree.

2.5.2.2 Random Forest

A random forest (RF) is a class of decision tree (DT) that works as an ensemble learning method to produce a class prediction by constructing many individual decision trees [39]. The prediction process involves aggregating the predictions of each individual tree in the forest and selecting the outcome that has the highest number of votes, thereby providing a reliable and accurate prediction [40]. The predicted outcomes of individual trees must have low correlations to one another for the model to perform well. When solving random forest algorithm problems, the method uses MSE (average of the error squares) to determine data distributions from each node [41].

$$\text{MSE} = \frac{1}{N} (f_i - y_i)^2 \dots \dots \dots (2.1) \quad [41]$$

In this equation, N stands for number of data points, f_i stands for the value that the model returns, y_i represents the definite value for i. The formula considers the distance between various nodes and the predicted value to help the user choose the best branch for their forest. For mentioned example, a datapoint, y_i is tested at a particular node and this decision tree returns f_i value. Figure 1 shows a simple Random Forest architecture. Various hyperparameters are used to optimize the performance of the random forest model, and the essential parameter descriptions for the thesis experiments are provided below -

- **N Estimators:** The parameter value specifies the optimal number of decision trees in a random forest. Though the model performance improves as the number of

decision trees increases, it is best to keep this value low first and gradually increase it until its performance stabilizes.

- **Max Depth:** The number of splits for each decision tree is selected with this parameter. The number of splits influences the model's fit, potentially resulting in overfitting or underfitting of the data.
- **Max Features:** This parameter defines the feature number that is used for an individual split.
- **Min Samples Split:** A non-terminal node requires a minimum number of samples to split, which is set to 2 by default.
- **Min Samples Leaf:** This parameter selects the minimum number of samples considered to be a leaf node, which is set to 1 by default.
- **Bootstrap:** Bootstrapping is a process for sampling data points randomly with a replacement which can make the model less prone to overfitting. The bootstrap parameter can allow or deny the use of bootstrapping.
- **Random State:** The parameter is used to ensure reproducible result by using the same seed value.

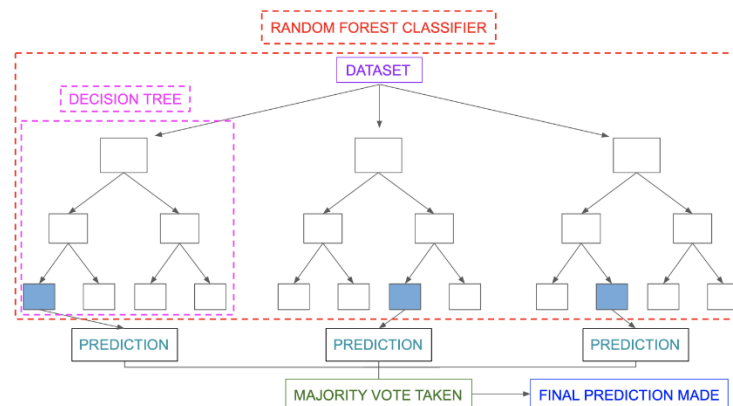


Figure 1: Simple Random Forest [42]

2.5.2.3 Support Vector Regression

Support Vector Regression (SVR) is an algorithm under supervised learning that predicts continuous values based on the theory of Support Vector Machines [43] [44]. SVR

simulates the objective of locating a hyperplane in an n-dimensional space that fits the data points and has the most significant error margin [45]. The data points located closest to the hyperplane influence the position and orientation of the hyperplane. In particular, SVR is a non-parametric technique that utilizes kernel functions to solve regression problems [46]. Additionally, SVR can manage non-linear data due to its adaptability in defining acceptable error values. There are a number of SVR parameters that are typically used to acquire the optimal model. Here are the parameters' descriptions -

- **Regularization Parameter (C):** This value for the hyperparameter creates a margin to maintain a low error rate throughout training and testing. The margin range shifts in the reverse direction as the value of the regularization parameter is adjusted, so decreasing the parameter value could lead to increased training mistakes but improved generalization to test data.
- **Degree:** This parameter value is included to specify the degree of the polynomial kernel function.
- **Epsilon:** The sensitivity of the model to errors can be defined by means of the epsilon parameter, which does this by specifying the width of the epsilon tube that surrounds the regression line. The sensitivity of the model changes reversely with the epsilon value.
- **Gamma:** The parameter determines the extent of influence exerted by a single training example, with smaller values indicating greater distance and larger values indicating closer proximity. The gamma parameters represent the reciprocal of the support vectors' radius of influence selected by the model.

2.5.2.4 K-Nearest Neighbor Regression

K-nearest neighbor regression (KNR) is a machine learning technique that relies on distance-based calculations, which is commonly employed for regression tasks and holds a resemblance to the k-nearest neighbor classifier [47]. The KNR algorithm is utilized to forecast the value of a continuous target variable, which is achieved by finding the nearest

k neighbors in the training set and computing the mean of their target values. The parameter k holds significant importance as a hyperparameter that can be adjusted to enhance the performance of the model. KNN regression uses the same Euclidean and Manhattan distance function as KNN classification.

$$\text{Euclidean} = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \dots \dots \dots (2.2) \quad [48]$$

$$\text{Manhattan} = \sum_{i=1}^k |x_i - y_i| \dots \dots \dots (2.3) \quad [48]$$

The KNR model employs multiple hyperparameters to enhance its performance. The following are the key parameter descriptions for the experiments conducted in the thesis -

- **Number of Neighbors:** In the context of the KNR algorithm, the number of neighbors is denoted by k, which represents the number of nearest neighbors considered when making a prediction for a new data point. The value of k has a significant relationship with the biasness of the model by leading to overfitting or underfitting [1][2].
- **Weights:** In KNN regression, the parameter weights refer to the weight function, which assigns a weight to each neighbor based on its distance from the query point. Closer neighbors receive higher weights, while farther neighbors receive lower weights, allowing KNN regression to prioritize neighbors whose proximity is most similar to the query point, resulting in more accurate predictions.
- **Algorithm:** The parameter algorithm in KNN regression denotes the method used to calculate the nearest neighbors. The prevalent algorithm is the brute-force approach that calculates the distances between every pair of points in the dataset. The k-d tree algorithm is a commonly used method that arranges the dataset into a binary tree structure to enhance the efficiency of the nearest neighbor search [3][4].
- **p:** The parameter p in KNN regression refers to the power parameter used for the Minkowski metric, which is a generalization of the Euclidean and Manhattan distances that allows for varied levels of sensitivity to feature value discrepancies. The value of p in KNN regression can have a substantial impact on model accuracy

since it dictates how much weight is given to distant neighbors in the prediction process.

2.5.3 Deep Learning Model

2.5.3.1 Neural Network

An artificial neural network (ANN) is a deep learning algorithm that draws inspiration from the structure of biological neural networks, featuring nodes that simulate the behavior of cell bodies and are interconnected through connections like axons and dendrites. Analogous to the strengthening of synapses between neurons in a biological neural network when their neurons exhibit correlated outputs, the connections between nodes in an ANN are assigned weights based on their efficacy in achieving a desired outcome. This process involves the continuous adjustment of connection weights through training, resulting in an ANN that can accurately classify and predict outcomes based on the input data.

2.5.3.2 Recurrent Neural Network

The Recurrent Neural Network (RNN) is a category of neural network that is specifically designed to model sequence data, including time series and natural language. Unlike feed-forward neural networks that only allow information to be passed between layers, RNNs enable information to be passed within a layer, making them particularly effective for handling sequential data [49][50]. RNNs can remember information using a cell-level feedback loop to communicate between neurons within the same layer [2].

Different types of RNNs have evolved, with the Simple RNN being the most commonly used due to its effectiveness in processing sequential data [51]. The SimpleRNN layer is a type of RNN layer that has a straightforward structure, such as a single tanh layer [52]. Another type of layer, namely the dense layer, is used to generate an output. The Sequential model allows stacking layers of RNN, i.e., the SimpleRNN layer class and the Dense layer class. The SimpleRNN class can be utilized for RNN implementation by specifying parameters, including unit count, dropout rate, and activation function. When stacking RNN layers, the `return_sequence` parameter of the previous layer must be set to `True`

ensuring the appropriate output format of the layer for the subsequent RNN layer. Finally, a single vector per sample is generated as output from an RNN layer. The RNN layer can be configured to return the complete sequence of outputs for each sample. Figure 2 shows a recurrent neural network architecture.

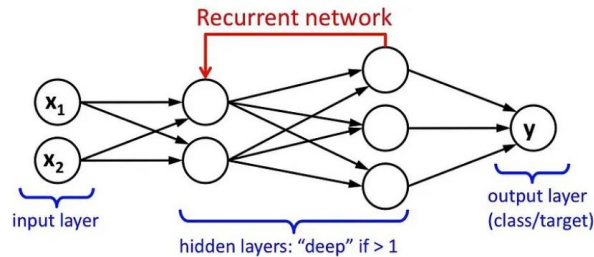


Figure 2: Recurrent Neural Network [53]

2.5.3.3 Deep Neural Network

A deep neural network (DNN) is a part of neural networks, which are a collection of hidden layers that a model can use to improve performance. Typically, a convolutional 1D layer is used for convolutional operations in deep neural network models, especially for processing sequential data such as time series or text for feature extraction. The operation of a deep neural network can be broken down into three key steps: input, processing, and output. Here is a brief overview of each step -

1. **Input:** The input to a deep neural network consists of a set of features or data points that are fed into the network for processing. This input is typically represented as a vector of numbers, with each element of the vector corresponding to a specific feature or attribute. The Flatten layer converts the multidimensional output from the previous layer into a 1D vector, which serves as input for the subsequent layers.

2. **Processing:** Once the input data has been fed into the network, it is processed through a series of layers. Each layer consists of a set of neurons that perform a specific computation on the input data. The dense layer is one of the most popular types of layers, in which each neuron in this layer is connected to every neuron in the preceding layer and performs intermediate calculations to learn complicated representations of the input data. These

computations involve applying a set of weights and biases to the input data, and then applying an activation function to the result. The output of each layer is then fed into the next layer, and the process is repeated until the final layer produces the output of the network.

3. **Output:** The output of a deep neural network can take many forms, depending on the task. For instance, in a classification task, the output can take the form of a set of probabilities, which indicates the degree of certainty that a particular input belongs to each class. For regression, the output is a numerical value that represents the predicted output for a given input. The weights and biases used in the computation of each neuron are initially set to random values but are adjusted during the training process to optimize the performance of the network on a specific task. Figure 3 shows a basic DNN architecture.

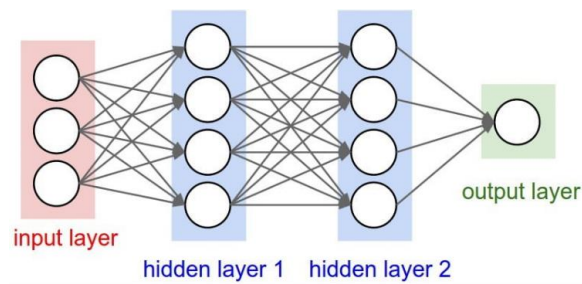


Figure 3: Deep Neural Network Architecture [54]

2.6 Implementation of ML Model for Predictive Analysis

Implementation of Machine learning model refers to the process of developing a framework that can be used to perform predictive analysis or make informed decisions based on the collected data. In most cases, the following procedures are usually applied when implementing machine learning models for predictive analysis -

- The initial decision that needs to be taken before implementing a machine learning model is the choice of a programming language. This decision includes the availability of various standard libraries and APIs.

- The algorithm that will be used in the implementation of a machine learning model should be selected from scratch. The selection includes the type of algorithm, its class, and the specific implementation type.
- The selection of a set of problems or a canonical problem is necessary to test and validate the implementation of the algorithm.
- In order to find the most accurate and up-to-date descriptions of an algorithm, researchers usually conduct extensive search operations on various sources such as books, websites, and libraries. Although they may prefer only a single description, multiple perspectives are needed in the algorithm research process. Multiple perspectives can help internalize the algorithm's description and overcome any roadblocks that arise due to its assumptions or ambiguities.
- One of the most important steps that must be taken during the development of a machine-learning model is the creation of unit tests. This process should be implemented for the purpose of ensuring that the developers understand the various requirements of the algorithm. After the implementation of an algorithm, one can then look into its possible improvements.
- Through experimentation, various decisions made during the development of an algorithm can be exposed and studied in different ways. This process can provide developers with new insights and improve the quality of their implementations.
- Through optimization, one can make use of different libraries, tools, languages, and data structures to improve the efficiency of the implementation. In this case, knowledge about data structures and algorithms from computer science can be beneficial.
- Making an algorithm more general can also create opportunities for developers. Since programmers are skilled at abstraction, they can easily see how it can be used to solve different problems.
- One can also make use of specialized techniques to improve an algorithm's performance. These techniques can be a valuable skill that can be required in the development of production systems. The two most commonly used techniques in machine learning to improve model performance are data preprocessing and feature engineering.

- **Data Preprocessing:** Data preprocessing is a technique involving the preparation of the raw dataset for machine learning model which is a crucial step in the data analysis pipeline as it ensures that the data is accurate, complete, and consistent by improving the quality of the insights derived from it. Additionally, data preprocessing helps to reduce the computational resources required for modeling by removing unnecessary features and instances. It involves techniques such as data cleaning, feature selection, normalization, and transformation to ensure that the data is suitable for analysis.
- **Feature Engineering:** Feature engineering (FE) is a set of preprocessing steps that help transform raw data into valuable features for machine learning. In predictive models, the most important predictor variables are selected and created during the engineering stage. The process of feature engineering involves creating, extracting, transforming, and selecting features that are ideal for developing machine learning algorithms.

Chapter 3

3 Literature Review

We researched previous approaches towards cyber-attack analysis to gain a better understanding of previous contributions in the field. The investigated implemented approaches differ significantly from our research approach. This chapter aims to review the literature related to thesis development, as well as provide a comprehensive comparative analysis among the related works including our proposed approach.

3.1 Detection-Based Approach for Cyber-Attack Analysis

Arora et al. use the Random Forest Algorithm to investigate cybercrime on social media [55]. They used WEKA, a software to support several data mining tasks, to compare the algorithms based on the F-measure value corresponding to the Accuracy and Precision rating. The researchers have proposed a model that will aid in the development of functionality to classify the threat and capture the user automatically. Considering the accuracy and ease of use with multiple datasets at once, the authors chose the Random Forest Algorithm for their implementation. NLTK and Wordnet libraries are used in their model. The researchers show that the NLTK library helps them to process the expressions that are given in the dataset. Wordnet is a sub-library that informs the system about the importance of certain words in the dataset. The dataset used for this implementation is based on many factors like Synonyms, Age, Location, Gender, Hashtags, Sarcasm and more. The implementation is composed of three main modules. The first module is data collection, which is focused on collecting public data from various social media platforms such as Twitter and Facebook. The second module is data cleaning, which is designed to remove unnecessary information. The third module is the classification section, which is designed to classify the data according to the training set. The following conclusions were drawn from the paper: 1) The Random Forests Algorithm for parameter-based threat detection is more than 80% accurate. 2) This algorithm has a precision of 0.81, which makes it the best fit.

Azizan et al. presented a model that considers the performance of three different algorithms against a network anomaly detection framework [56]. The objective of their method was to perform knowledge discovery and pattern evaluation using the IDS (Intrusion Detection System) framework. The research was carried out using the Knowledge Discovery in Databases (KDD) method. The KDD method is an automatic discovery process that involves finding hidden rules and patterns in substantial amounts of data. The researchers determined either positive or negative detection of an attack and, afterward, utilized the Intrusion Detection Evaluation Dataset from CIC-IDS2017 to evaluate the performance of classifiers. The results of the study revealed that the SVM has the highest accuracy of 98.18%, followed by the DJ (Decision Jungle) and the RF (Random Forest).

Elmrabit et al. aimed to evaluate the capabilities of different ML algorithms to identify anomalous behaviors in real-world networks [57]. The general structure of their proposed method involves data cleaning, feature scaling, binary classification label normalization, and multi-classification label scaling. The researchers carried out the evaluation on three public datasets: the Cyber-attack dataset of the Industrial Control System, the UC-NB15 dataset, and the CICIDS-2017 dataset having different features such as types of events, names of attacks, timing, and scenario patterns. The results of their evaluation revealed that the Random Forest algorithm performed best in terms of accuracy, recall, precision, and ROC curves on the mentioned three datasets. Random forest showed 99% accuracy for the CICIDS-2017 dataset, and it was their best achieved accuracy.

RM et al. present a deep neural network that can be used to develop efficient and effective IDS for identifying and detecting cyberattacks [58]. The network is designed to perform various tasks, such as preprocessing and hyperparameter tuning. They present a deep neural network model that is a hybrid of PCA (Principal Component Analysis) and the GWO (Grey Wolf Optimization) metaheuristic-based intrusion detection approach. When compared to other similar swarm intelligence-based techniques, the Grey Wolf Optimization Technique is used because it is a mathematically proven model with superior exploration characteristics. The main advantage of this optimizer is that it is a natural leader with the ability to control the search agents in the environment over time. Whereas the other optimization techniques do not have a natural leader, the user must decide the

leadership characteristic through trial and error, which reduces the optimizer performance. Performance evaluation on DNN and other ML algorithms has been done by the researchers on the benchmark intrusion detection dataset. The researchers show that their proposed model can outperform existing machine learning techniques. They evaluated their proposed model using a dataset that has been universally benchmarked from Kaggle and achieved almost 99.9% accuracy.

Zhang et al. provide a comprehensive overview of the latest developments in the field of cyber-attack detection using DL solutions [59]. The researchers focus on analyzing cyber-physical system (CPS) data as cyberattacks continue to pose a threat to CPS security. This survey proposes a six-step DL-driven method for summarizing and analyzing the twenty recently published papers. Their approach shows a deep understanding of the various steps involved in the deep learning process, which include analyzing the scenario, developing a problem model, performing an evaluation, and acquiring data. A comprehensive view of a CPS scenario is obtained by inspecting it, identifying the cybersecurity issues, developing a problem model, and preparing datasets for evaluation. Through rigorous analysis, the authors suggest that DL models can be useful to analyze and utilize the data collected from CPSs.

Sengan et al. focuses on the issue of data integrity in the smart grid system [60]. Their first contribution is an attack exposure metric that can be used to measure the vulnerability of the system to cyber-attacks. The research then explores how to decentralize the security of the system by implementing an agent-based approach. Their proposed approach includes a measurement meter and uses an artificial neural network to process data. The proposed AFN (Artificial Feed-forward Neural Network) model's input layer receives the meter reading, which is then passed through four hidden layers with assigned weights to train and transform the data. The researchers conducted a series of tests on the effectiveness of the proposed model and compared their efficiency with that of existing deep-learning models. The results of the investigations revealed that the Artificial Feed Forward Neural Network model was able to detect 98.19% of false data. The authors concluded that deep neural networks could be effective to analyze the cyber data collected from smart grids to detect attacks and incidents.

Al-Abassi, and colleagues proposed a deep learning model that uses a combination of DNN and DTC (Decision Tree Classifier) classifiers to detect cyber-attack activities [61]. Their proposed model consists of multiple unsupervised SAE (Stacked Autoencoder) that use multiple Autoencoders (AE) to extract a new representation from unlabeled data to obtain different patterns with the purpose of creating a new balanced representation of an imbalanced dataset. The output from SAE is then passed to a DNN via a super vector and concatenated using a fusion activation vector. The new model is then fed into an attack detection framework that is designed for industrial control systems. The researchers evaluate the performance of the proposed model against various conventional classifiers, such as AdaBoost, Random Forest, and DNN. The evaluation results reveal that the proposed model performed well compared to others, with 96% accuracy. In addition, the authors note that the proposed model is able to detect 99.67 percent of attacks from the SwaT dataset.

Bilen et al. analyzed various machine learning methods on cyber-crime data collected from the occurrence in the province of Elazig between 2015 and 2019 to detect assailants and cyber-attack types [62]. Their research methodology involved selecting useful features by analyzing the correlation of data with the target variable and rescaling them using the StandardScaler function in the python library. The researchers were able to identify the characteristics of the assailants based on their model, and they observed that the likelihood of these attacks depends on the income and education level of the victim. They found that the Support Vector Machine linear was the most accurate method for identifying cyber-attack types, with an accuracy rate of 95.02%. The Logistic Regression method was the most accurate method for detecting attackers, with 65.2% accuracy. Furthermore, the authors presented a comprehensive review of their accuracy ratios.

3.2 Prediction-Based Approach for Cyber-Attack Analysis

To accurately predict the type of cyber-attack at any given time, Ben Fredj et al. suggested two neural network-based approaches, the Basic Model, and the Looking Back Model, which include LSTM (Long Short-Term Memory), RNN (Recurrent Neural Network), and MLP (Multilayer Perceptron) [63]. While the Basic model uses the source IP and

destination IP of the attack at a given time as inputs, the second model considers the current type of attack with the preceding attack type. The authors utilized the Ctf'17 dataset, sourced from Defcon, the largest internet security community, which contained traffic data generated during the CtF'17 competition in 2009. To ensure the robustness and generalizability of the models, the researchers experimented with different configurations and hyperparameter values. They evaluated the performance of developed models under several evaluation metrics, including recall, precision, and the F-measure, and observed that the LSTM outperformed the other models.

The developed model by Ansari et al., which is based on the Gated Recurrent Unit (GRU), can learn from various security event sequences and output future threats based on the history of attacks [64]. The authors used alert data, including generic fields such as attack detection time, attack volume, target IP, port, etc., obtained from the Warden threat sharing platform, an open-source platform designed to automatically share detected security events between Cyber Security Incident Response Teams (CSIRT). They presented a hybrid model that included different classifiers such as J48 (Decision Tree classification algorithm), Random Tree, REP Tree (Reduce Error Pruning Tree), AdaBoost (Adaptive Boosting), Decision Stump, and Naive-Bayes, as well as a voting algorithm with Information Gain that combined the probability distributions of the used learners to select the best classifier. Prediction quality is estimated by comparing predicted alerts for all test vectors to their corresponding ground truth alerts, where the dissimilarity metric between actual and predicted alerts (error value) is computed as a weighted sum of individual field dissimilarities.

Sokol et al. implemented NSSA (Network Security Situation Awareness), a method of recognizing, analyzing, and responding to potential security risks in a network, to develop techniques for predicting the future trend of network security issues [65]. They used a dataset that contained category, IP address, port, protocol of network traffic source data, and target data, detection time, and interruption time from Warden collected between December 2017 and December 2018. The researchers used the quantitative forecasting method, which involves predicting the future tendency of network security situations using statistical and neural network models, including the performance comparison of multiple

statistical and neural network models using a 95% prediction interval. Combining encoder-decoder with DNN yielded promising results when implementing various neural network types and data scaling methods to determine the optimal model.

Yin et al. proposed a framework based on the TCN-Combined Transformer model to forecast longer-term changes in the current state of network security, which incorporates both the Temporal Convolutional Network (TCN) model and the Transformer model [66]. A temporal convolutional network (TCN) is a neural network architecture designed explicitly for processing sequential data, and transformer model is a category of neural network architecture which is a sequence-to-sequence model that processes input data using self-attention mechanisms. For the experiments, the researchers used two datasets: the UNSW-NB15-2015 dataset and the CSE-CIC-IDS2018 dataset, both of which contain detailed descriptions of intrusion and abstraction distribution models for applications, protocols, or lower-level network entities, as well as network traffic for each computer, including the victim, log files, and 80 network traffic features. The authors demonstrate through experiments that the proposed model outperforms five baseline models in terms of temporal sequence prediction accuracy and robustness.

3.3 Analysis of Current Research Gap

The current research strategies in the field of cyber-attack prediction have been observed to possess certain limitations. These are listed below –

- One of the major shortcomings identified in the current research strategies is the dependence on datasets from the past, which may not necessarily reflect the current scenario and can lead to inaccurate predictions.
- In addition, the existing literature lacks a comprehensive visualization of the current global cyber-attack trends, which can affect the development of effective preventive measures.
- Moreover, most research conducted in the cyber security domain concentrates on detecting the existence of cyber-attacks, predicting cyber-attack types, and

identifying future security trends in a specific network by analyzing incident characteristics. While these approaches have been successful to some extent, it fails to address the growing complexity of global cyber-attacks, which can take various forms and operate in dynamic environments.

- While time series analysis techniques have the potential to improve threat detection, response, and prevention mechanisms in cybersecurity, only a handful of studies have been conducted in this area.

3.4 Comparative Analysis and Novel Contributions

Table 1 shows a comparative analysis of the related works with our approach. As illustrated in the table, most of the work related to cyber threat analysis has been done to determine the presence and type of cyber-attack or to predict the probable future network security alert from the pattern of specific occurrences.

We have listed below how our research differs from the previously cited works by addressing the identified limitations in the current state of research, which therefore indicates our novel contributions in the cyber security research area –

- A substantial part of related works includes datasets from the past, in contrast, our research includes recent data collected from an up-to-date website, and our data collection framework can continue collecting the latest updated data. Additionally, our collected data can be a valuable resource for future cyber-attack analysis.
- Compared to related works, our thesis has made it easier to have a proper visualization of current global cyber-attack trends with the implementation of the Power BI tool and the integration of an up-to-date data collection system.
- Furthermore, the vast majority of cyber security studies concentrate on cyber-attack detection, cyber-attack type prediction, and the forecasting of future security trends within a given network; conversely, with the incorporation of the real-time global cyber-attack data collection framework, we have developed a method for anticipating the severity of attacks in a future period based on counts of previous incidents.

- In addition, our developed data preparation and time series analysis techniques differ from related implementations, which will aid in future research development in similar areas through understanding the methods and processes. We conducted an exploratory analysis to establish that the data we collected is suitable for time series analysis and the assessment of data validity presented here is absent in the pertinent literature.
- We introduced a novel approach by incorporating the time window parameter in our research and demonstrated that by choosing an appropriate time window, we may improve the model's performance by taking advantage of the enhanced correlation between various time lags. Overall, our results demonstrate that elevated autocorrelation values coupled with extended time windows enhance both predictive capabilities and precision of the model, representing an unexplored avenue in prior research.
- Also, another current research gap in the field of prediction modeling is the need for more clarity in achieving optimal results, while our research offers an in-depth analysis of various time windows and model architectures to explain our step-by-step approach to developing an optimal model.

Additionally, the findings presented in Table 1 showed that Decision Tree, Support Vector Machine, and Deep Neural Network were encountered to be beneficial for cyber-attack analysis. Correspondingly, we have included SVR, Random Forest, XGBoost, and Deep Neural Network for our research development.

Table 1: Comparative Analysis of the Related Works with Our Implementation

Ref	Approach	Dataset	Task	Best Model
[55]	Multi-factor Analysis	Data Collected from social media	Parameter based threat detection	RF
[56]	KDD method	Intrusion Detection Dataset (CIC-IDS-2017)	Network intrusion detection by analyzing pattern	SVM
[57]	Data Cleaning, Scaling and Normalization	Industrial Control System, UC-NB15, CICIDS-2017 datasets	Identifying cyber-attack in real world networks	RF
[58]	PCA and GWO Optimization	Kaggle	Identifying the presence of cyber-attack	DNN
[60]	FNN-based Measurement Meter	Data from smart grid system	Detecting false data in smart grid system	FNN
[61]	Unsupervised Stacked Autoencoder	ICS datasets obtained in 2015 and 2018	Cyber-attack detection in ICS infrastructure	Combined DNN and DTC
[62]	Feature Selection and Rescaling	Cyber-crime Data Occurred in the province of Elazig between 2015 and 2019	Identifying if the same assailant carried out the cyber-attack or not by analyzing the characteristics	LR
[62]	Feature Selection and Rescaling	Cyber-crime Data Occurred in the province of Elazig between 2015 and 2019	Detection of cyber-attack type	SVM
[63]	Basic and Looking Back Model	Ctf'17 dataset in 2009	Predicting potential type of cyber attack	LSTM
[64]	GRU-Based Deep Learning Approach	Cyber security alert data collected from Warden	Predicting potential future cyber security alert based on previous security alert	Hybrid model combining J48, Random Tree, REP Tree, AdaBoost, Decision Stump and Naive-Bayes
[65]	Time Series Analysis based on statistical and neural network model	Dataset containing security alert collected from Warden between 2017 and 2018	Predicting future trend of network security issue	DNN
[66]	TCN-Combined Transformer Approach	UNSW-NB15-2015 and CSE-CIC-IDS2018 datasets	Forecasting longer-term changes in the current state of network security	DNN
Our Approach	Time Series Analysis based on machine learning and deep learning approach	Imperva's Website	Predicting potential cyber-attack intensity for different periods of time by analyzing up-to-date data pattern	CNN and RNN

Chapter 4

4 Methodology

In this chapter, our developed global cyber-attack pattern analysis approach is described. Namely, for our prediction model creation, we will explain our data collection framework, Power BI platform development for cyber-attack trend analysis, statistical method, machine learning, and deep learning-based techniques.

4.1 High-Level System Architecture

The importance of a global cyber-attack pattern analysis approach has been made apparent in the previous chapters. Considering the shortcomings of current cyber-attack analysis approaches, the developed architecture looks to address several key areas. Figure 4 shows the elementary architecture of our implementation.

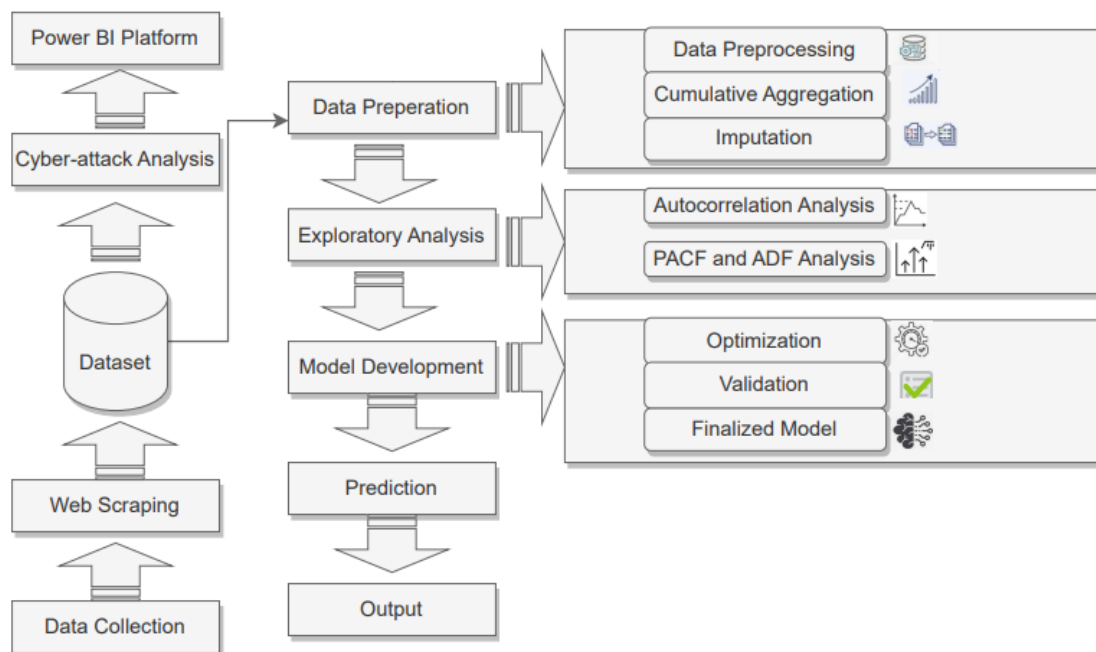


Figure 4: Global Cyber-attack Pattern Analysis System Architecture

The steps include data collection through web scraping, cyber-attack trend analysis, data preparation, exploratory analysis, model development and prediction. Data preparation includes data preprocessing: a process of preparing the raw data, cumulative aggregation: building features based on the cumulative sum over the period and imputation: a method to estimate missing values. The exploratory analysis includes autocorrelation analysis, partial autocorrelation (PACF), and augmented dickey-fuller analysis (ADF) which will be described in more detail later in this chapter. Finally, model development includes optimization: a technique to improve the model performance iteratively, model validation, and finalizing the model.

4.2 Data Collection Framework and Dataset

4.2.1 Data Collection Framework Development

The first step in our research is to gather current data for analysis; therefore, we have collected data from the website of Imperva, which is constantly updated with new cyber-attacks from around the world [67]. Imperva is a company that provides unrivaled end-to-end application and data security that safeguards critical applications, APIs, and data anywhere. Our developed data collection framework will simplify the assembling process of current data that will contribute to making informed decisions and taking appropriate actions based on the current situation. We compiled the collected data into an Excel file using web scraping.

4.2.1.1 Web Scraping

The number of explanatory substances that can be found on the web today is enormous. Due to the nature of the web pages, the users will be focused on the contradiction between the facts and the visual representations. Websites do not allow users to save their data to a directory using a web browser, and this is because they do not offer the capability to do so. The only way to do this is by manually copying and pasting the data into a hard drive. Web scraping is a technique that involves automatically extracting data from websites and storing the collected information in a central database [68]. The process is carried out by web scrapers, which are software programs that are specially designed to perform this type

of operation. It can be created for specific websites, or it can be an organized tool that can work with any website. The main goal of a web scraper is to extract data from the websites so that it can be stored in an organized database. Some of the procedures used in web scraping include HTML Parsers, DOM Parsers, and HTTP programming. These allow us to get error-free data, which is very beneficial as it enables us to perform quick analysis and retrieve information from websites. The ease of access that web scraping provides makes it very beneficial for users. Some popular frameworks and libraries used for this include Selenium, BeautifulSoup, and Scrapy. We have used selenium for our data collection to scrape the relevant data from the webpage.

4.2.1.2 Selenium

The suite of automation tools and techniques known as Selenium is immensely popular among testers due to its various advantages [42]. It allows testers to run tests on a target browser and drive interactions on web pages without requiring manual input. In 2004, the emergence of Selenium made it easier for automation testers to perform their tasks, and it reduced the time it took to detect and analyze failures. It can run tests on multiple operating systems and browsers which allows enterprises to deliver high-quality web applications. Developers can easily create and run Selenium test scripts in various languages, such as C#, PHP, Perl, Python, and Java which makes it quite easy for testers to use it. It can be executed on different operating systems and browsers, such as Google Chrome, Mozilla Firefox, Opera, and Safari. We have used the selenium package to interact with web browser from python automatically. We have used chrome driver to interface with the preferred browser, as Selenium also requires a web driver to control browser.

4.2.1.3 Chrome Driver

The Chrome Driver is a utility that is used by Selenium WebDriver to control Google Chrome [69]. It is maintained by the team of Chromium and is expected to be installed in the default location of browser. Setting a special capability allows us to force Chrome Driver to use a customized location. We installed Chrome Driver in our platform and included its path in our environment variable.

Given the information presented above regarding our data collection method, we have summarised the complete procedure in Figure 5. Our data collection process starts with a continuous loop until the kernel is interrupted. The program continues to collect data through web scraping automation and saves the data in an excel file.

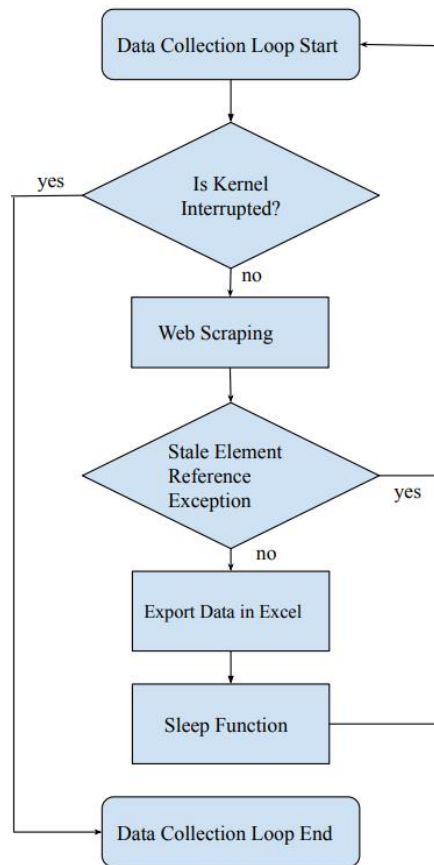


Figure 5: Data Collection Framework

We encountered various challenges during the data collection process through web scraping. As shown in Figure 5, we introduced the necessary steps to resolve these issues. The issues and solution strategies are demonstrated below:

- **Dynamic change in website:** When a page is dynamically updated or when elements are removed or added, Stale Element Reference Exception can occur. This is a type of error that occurs while using the Selenium WebDriver to scrape a web page. It occurs when the previously found element no longer exists in the document

object module of the page. We have handled Stale Element Reference Exception by calling the pass statement which allows the program to continue with the next loop without crashing.

- **Unresponsive Website:** At times our data collection process was overwhelming the server with requests which caused the website to slow down or become unresponsive by blocking our request to the server, making it difficult or impossible to access the data. We resolved this issue by calling the time sleep function allowing the program to sleep for a specified number of seconds.

Algorithm 1 demonstrates step by step procedures for our whole data collection technique.

ALGORITHM 1: ALGORITHM FOR DATA COLLECTION FRAMEWORK

- 1 *Import necessary libraries*
 - 2 *Define an infinite loop to collect data from the web page until kernel is interrupted*
 - 3 *Locate the required elements on the webpage by using their XPATHs*
 - 4 *Create empty lists for each data column*
 - 5 *Loop through each element in required column and append value to respective list*
 - 6 *Use the datetime module to get the current date and time*
 - 7 *Create a DataFrame with the collected data with date and time*
 - 8 *Write the DataFrame to a CSV file in append mode to continue adding new data*
 - 9 *Sleep the program for a specified number of seconds*
 - 10 *Ignore the Stale Element Reference Exception and continue with the next iteration*
-

4.2.2 Collected Dataset Description

We worked on the dataset collected for seven days from the website of Imperva which includes live attack update. We collected nearly 144k data points over a period of seven

days that may serve as a valuable resource for future research endeavors. Imperva updates cyber-attack information every few seconds. While confirmation research is conducted on the attacks update, it is likely that some attacks go unreported or are reported with inaccurate information, such as the update of the attack. This website does not claim to be exhaustive, but it provides a comprehensive overview of the threat landscape. The data we collected contains categorical features, a type of variable that takes into account a set of fixed and limited possible values. Figure 6 shows a section of our collected dataset in excel file. The features in our dataset are listed below-

1. **Source Country:** The source countries are those countries that launched the attack. Our dataset includes 233 unique countries.
2. **Destination Country:** Destination countries are those countries that were attacked by the source countries. Our dataset holds 148 destination countries.
3. **Industry:** The various industries affected by the attack include education, healthcare, financial services, IT, retail, and government. Our dataset contains 20 different industrial sectors.

A	B	C	D	E	F	G
Source_Country	Destination_Country	Industry	Type_of_Attack	Type of Malware	Date/Time	
Serbia	United States	Computing & IT	Automated Threat - Bus	Hacking Tool	11/05/2023 00:00:03	
China	United States	Society	Automated Threat - Aut	Browser Automa	11/05/2023 00:00:03	
Hong Kong	Italy	Business	Automated Threat - Aut	Vuln. Scanner	11/05/2023 00:00:03	
Aruba	United States	Travel	OWASP - API Violation	Browser Automa	11/05/2023 00:00:03	
South Korea	United States	Computing & IT	Automated Threat - Bus	Browser Automa	11/05/2023 00:00:03	
Germany	France	Retail	Automated Threat - Aut	Hacking Tool	11/05/2023 00:00:14	
Czech Republic	Canada	Financial Services	OWASP - API Violation	Hacking Tool	11/05/2023 00:00:14	
United States	United States	Automotive	Automated Threat - Bus	Browser Automa	11/05/2023 00:00:14	
Sweden	United States	News	Automated Threat - Aut	Bot	11/05/2023 00:00:14	
United States	United States	Financial Services	OWASP - XSS	Browser Automa	11/05/2023 00:00:14	
United States	Hong Kong	Computing & IT	OWASP - RCE/RFI	Hacking Tool	11/05/2023 00:00:25	
United States	Australia	Healthcare	OWASP - API Violation	Hacking Tool	11/05/2023 00:00:25	
United States	United States		OWASP - Protocol Mani	Hacking Tool	11/05/2023 00:00:25	
Canada	United States	Sports	Automated Threat - Bus	Hacking Tool	11/05/2023 00:00:25	
Australia	United States	Computing & IT	Automated Threat - Bus	Bot	11/05/2023 00:00:25	
United Kingdom	United States	Healthcare	Automated Threat - Aut	Bot	11/05/2023 00:00:35	
United States	United States	Law & Governme	OWASP - Data Leakage	Hacking Tool	11/05/2023 00:00:35	
Ireland	United States	Healthcare	OWASP - API Violation	Bot	11/05/2023 00:00:35	
Hong Kong	Germany	Financial Services	Automated Threat - Aut	Hacking Tool	11/05/2023 00:00:35	
United States	United States	Financial Services	Automated Threat	Browser Automa	11/05/2023 00:00:35	
United States	United States	Business	Automated Threat - Bus	Bot	11/05/2023 00:00:46	
Ireland	Ireland	Financial Services	OWASP - Path Traversal	Vuln. Scanner	11/05/2023 00:00:46	
Mexico	United States	Automotive	OWASP - API Violation	Browser Automa	11/05/2023 00:00:46	
Pakistan	Singapore	Travel	Automated Threat - Bus	Browser Automa	11/05/2023 00:00:46	
Spain	United States	Computing & IT	Automated Threat - Bus	Browser Automa	11/05/2023 00:00:46	
France	United States	Travel	OWASP - API Violation	Browser Automa	11/05/2023 00:00:56	
Thailand	France	Computing & IT	OWASP - API Violation	Browser Automa	11/05/2023 00:00:56	
Brazil	United States	Gaming	Automated Threat - Bus	Browser Automa	11/05/2023 00:00:56	
India	Kenya	Travel	Automated Threat - Bus	Browser Automa	11/05/2023 00:00:56	
Germany	United States	Financial Services	OWASP - Path Traversal	Browser Automa	11/05/2023 00:00:56	
Canada	United States	Financial Services	OWASP - XSS	Browser Automa	11/05/2023 00:01:07	
Canada	United States	Business	OWASP - API Violation	Browser Automa	11/05/2023 00:01:07	
South Korea	United States	Business	OWASP - API Violation	Browser Automa	11/05/2023 00:01:07	

Figure 6: Dataset

4. **Type of Attack:** The type of attack that was carried out can define the scope and capabilities of the attacker. Our dataset includes eighteen types of cyber-attacks.
5. **Type of Malware:** Malware refers to the software or technique that is used to carry out successful cyber-attacks. We have observed eight different types of malwares in our dataset.
6. **Date-Time:** Date and time is the cyberattack occurrence time.

4.2.3 Data Preparation and Exploratory Analysis

Initial preprocessing of the collected dataset consisted of removing duplicates and null values. Type of attack, Destination Country, and Date-Time are the three characteristics from the dataset that we have primarily utilized for our prediction analysis. For the purpose of our experiment, we have initially chosen to focus on the most recent highly targeted country based on our collected data, namely the United States (US), and the top attack type, which is Automated Threat- Business Logic as showed in Figure 7. By including the second most targeted country, the United Kingdom (UK), and retaining the second most attack type, namely API Violation, we aim to demonstrate the versatility of our developed techniques in handling various scenarios. Throughout our analysis we have considered Automated Threat Business Logic as Type 1 attack and API Violation as Type 2 attack. If we consider the set of a particular cyber incident count over time $C = \{c_1, c_2, c_3 \dots c_n\}$, this makes the volume of a specific cyberattack for each particular time interval a candidate for time series analysis. Since we worked with seven days' worth of data, we decided on a smaller time window so that our analysis would be suitable for the available data. The time frame can be altered as the number of data points increases. We computed the attack volumes for different values of time windows represented by τ where $\tau = 5, 10, 15, \text{ and } 30$; minute is the measurement unit. We have considered multiple imputation techniques to fill in the missing data, including forward filling, linear interpolation, and polynomial interpolation. Linear interpolation considers a linear relation between data points that eventually may not accurately reflect data changes because of its inherent nature. Additionally, forward filling assumes that the last observed value is constant until a new value is observed, which can lead to the missing values being filled with constant values that do not reflect the true fluctuations of the data. On the contrary, the polynomial

interpolation method seeks to identify the polynomial function that best fits the data. Finally, we chose polynomial interpolation because it can accurately depict the relationship between data points and fill in the missing values, as necessary.

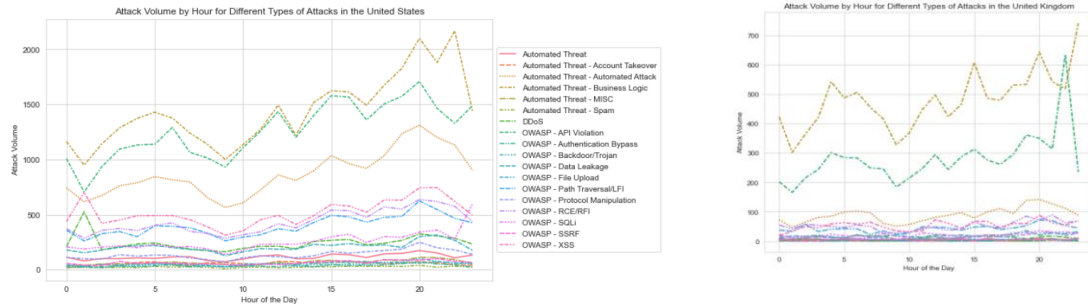


Figure 7: Different Attack volumes for United States and United Kingdom

4.2.3.1 Time Series Data Analysis using Autocorrelation (ρ)

It is attainable to forecast future attack volumes if successive incident counts show correlation. Hence, we have conducted an auto-correlation analysis for several time intervals to determine the viability of using our dataset for time series analysis. As mentioned in the previous section, we began our analysis with a top-level attack in the US before gradually moving on to other scenarios based on our observations. From Figure 8 and Table 2, we observe that the correlation values are consistently above the threshold value up to a lag of 5, indicating that our data can be used for time series forecasting. Additionally, Table 2 demonstrates that the correlation value for the United States decreases with shorter τ value, which prompted us to conduct the experiment for the United Kingdom only with a 15-minute time window. Furthermore, we have added a second time window ($\tau = 30$ minutes) analysis for UK to test whether it improves the accuracy of our predictions.

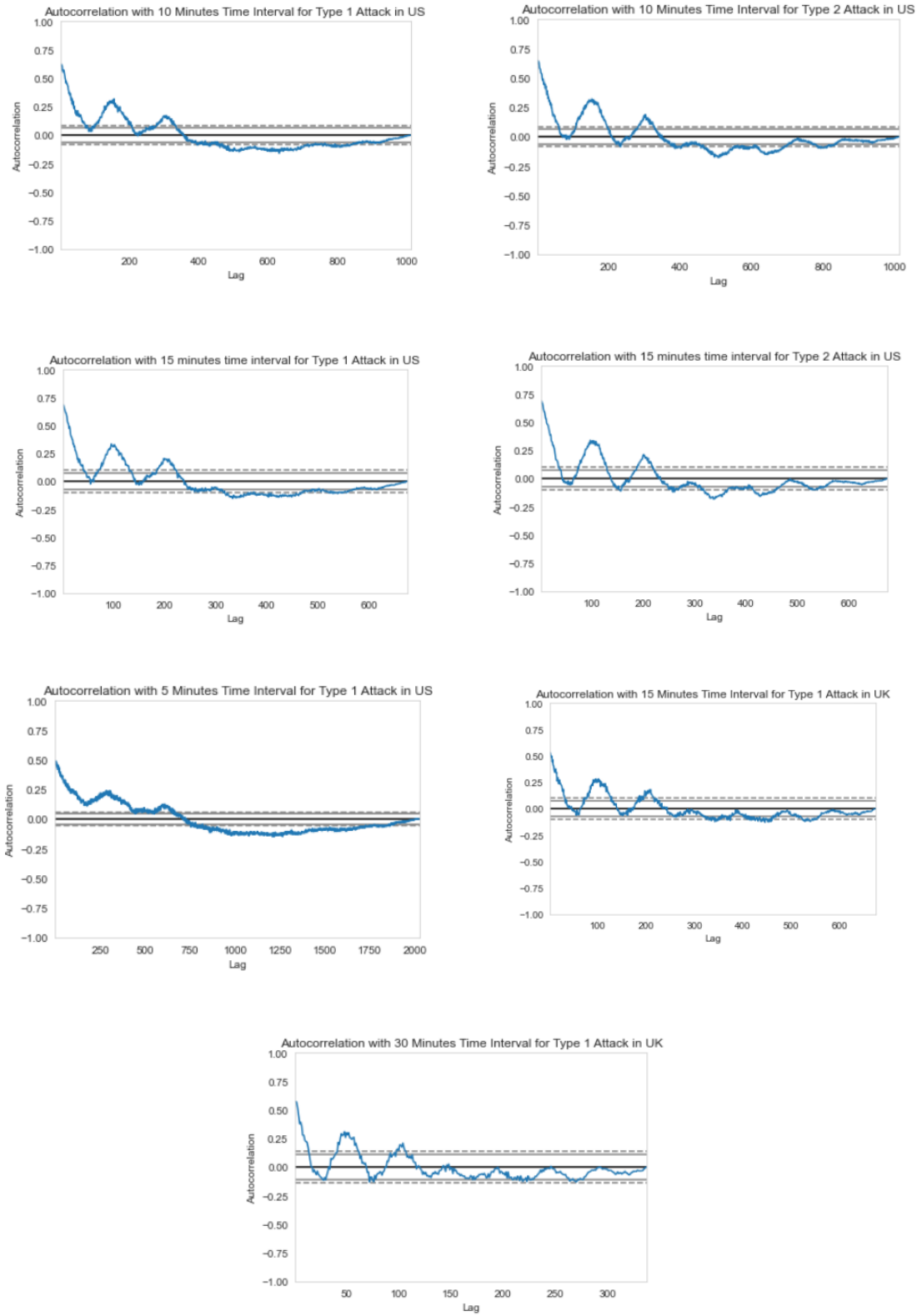


Figure 8: Auto-correlation Map for different lag Values

Table 2: Auto-correlation Values for different lags for Type 1 and Type 2 Attack

Lag	Type 1 Attack - US			Type 2 Attack - US		Type 1 Attack - UK	
	$\tau = 10$	$\tau = 15$	$\tau = 5$	$\tau = 10$	$\tau = 15$	$\tau = 15$	$\tau = 30$
1	0.624	0.687	0.486	0.651	0.691	0.532	0.574
2	0.626	0.669	0.481	0.645	0.669	0.511	0.577
3	0.606	0.654	0.493	0.625	0.671	0.488	0.541
4	0.609	0.643	0.470	0.635	0.642	0.450	0.473
5	0.576	0.626	0.476	0.601	0.618	0.478	0.397

After a detailed analysis of the overall autocorrelation values, we developed our techniques to predict the next period's attack volume based on the past four observed attack volumes as features, and therefore, the prediction function can be defined with the equation below:

$$c_i = f(c_{i-4}, c_{i-3}, c_{i-2}, c_{i-1}) \dots \dots \dots (4.1)$$

4.3 Model Development

4.3.1 Dataset Partitioning and Model Training Protocol

We have partitioned our dataset in a stratiform manner, allocating the first 80% of the data for training and reserving the last 20% for testing. This division prevented any data leakage that would have harmed the model's performance evaluation. Since the sequence of a pattern is crucial for time series analysis, the sequential order of the data in our case was rigidly maintained.

During training, we adopted a sequential approach that leveraged the temporal nature of the data. Specifically, we transformed the attack volume observations into sequences, where each sequence encompassed four consecutive observations, and these sequences served as the fundamental input features for our models. The models were created to forecast the attack volume for the upcoming period based on the previous four observations to aid in learning. Since this formulation naturally reflects temporal dependencies, the

models can recognize patterns and trends across time. Our models were able to learn correlations between the past attack volumes within the training data in this manner. For the prediction process, the models were provided with a sequence of four recent attack volume observations to generate a prediction for the attack volume in the subsequent period. In order to implement model training successfully, our training approach also included determining the optimized hyperparameter values, which required grid search and random search to discover the best values. We chose to use grid search and random search because we needed a methodical, understandable, and resource-conserving way to optimize the hyperparameters. Following are some of the major reasons why we chose grid search and random search over other optimization techniques -

- **Simplicity:** Grid search and random search are simple and well-known approaches for adjusting hyperparameters. Their ease of use in both execution and interpretation makes them appropriate for an extensive implementation like ours, which aims to compare various models and methodologies.
- **Efficiency of Resources:** Given the challenges of training several ML and DL models on a worldwide scale to anticipate the intensity of cyber-attacks, grid search and random search provided a balance between optimization quality and computing demands. More complex techniques, such as Bayesian optimization, may need more computational resources, which may be prohibitively expensive given the scale of our investigation.
- **Execution Time:** Although certain sophisticated optimization techniques may result in faster convergence to the ideal hyperparameters, they may be computationally demanding. The execution time requirements for grid search and random search in our experiments were acceptable.

4.3.2 Baseline Model Development: Statistical Model

Since ARIMA is the model that is most frequently utilized for modelling stationary time series data, we have chosen to use it as our baseline model. As discussed in the background

chapter, to implement the ARIMA model, the Autoregression value (p), the Integrated value (d), and the Moving Average value (q) need to be determined.

4.3.2.1 Selection of Optimal Order Value for ARIMA

For each experimental scenario, we have conducted Augmented Dickey-Fuller (ADF) testing to identify our time series data characteristics. The ADF test is frequently used to determine the stationarity of a time series dataset. Stationarity is an essential property of a time series data, indicating that the mean and standard deviation do not change over time. The ADF test generates an ADF statistic, which are used to assess the null hypothesis. The null hypothesis of the ADF test is that the time series is non-stationary because it has a unit root, and alternately, the time series may be stationary.

Table 3 illustrates the ADF test values for our experiments. We have compared the ADF statistic to the critical values at different significant levels (1%, 5%, and 10%). If the ADF statistic is less than the critical value, it suggests rejecting the null hypothesis in favor of stationarity. In most cases, we observe that the ADF statistic value is lower than the critical values, indicating that we can reject the null hypothesis and conclude that the time series is likely stationary. The experiment clearly demonstrates that no additional value is required to make the data stationary, as such d value is 0.

Table 3: ADF Test Values

	Type 1 Attack - US			Type 2 Attack - US		Type 1 Attack - UK	
	$\tau = 10$	$\tau = 15$	$\tau = 5$	$\tau = 10$	$\tau = 15$	$\tau = 15$	$\tau = 30$
ADF Statistic	-3.416	-3.459	-3.407	-3.691	-3.415	-4.709	-4.779
Critical Value:1%	-3.437	-3.440	-3.434	-3.437	-3.440	-3.440	-3.450
Critical Value:5%	-2.864	-2.866	-2.863	-2.864	-2.866	-2.866	-2.870
Critical Value:10%	-2.568	-2.569	-2.568	-2.568	-2.569	-2.569	-2.571

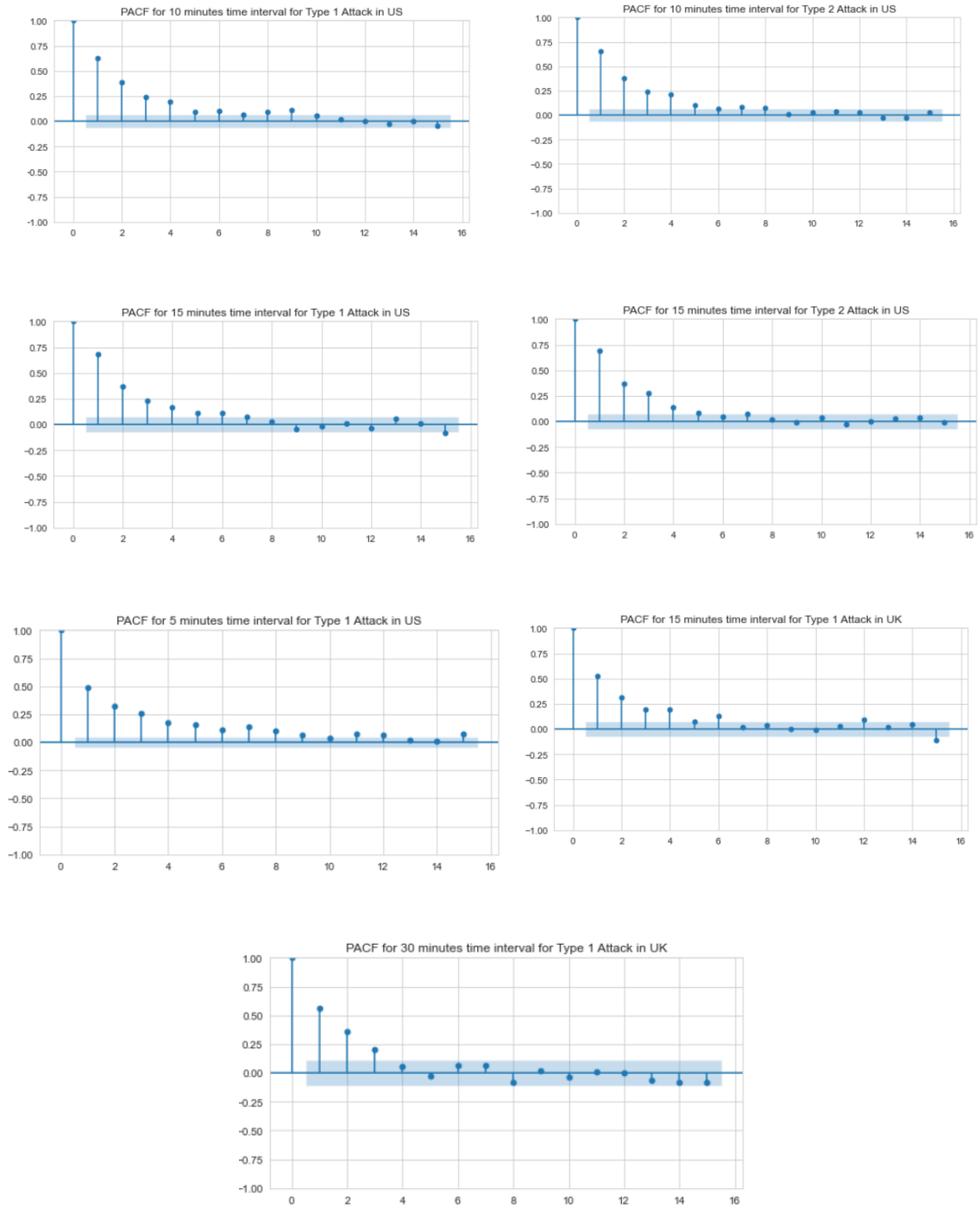


Figure 9: PACF Plot

As illustrated previously, in an ARIMA model, p is the lag value at which the PACF (Partial Autocorrelation Function) plot drops off to zero for the first time [70]. The PACF plot is a plot of the partial correlation coefficients between the series and lags of itself, and it aids

in identifying the order of the AR (Autoregressive) model. The PACF plots in Figure 9 illustrate the direct connection between an observation and its lag. We observed the lag value at which the PACF plot drops to zero for the first time to identify the model's order for each specific scenario.

We have determined the q value based on Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). AIC and BIC are statistical measures used in model selection and statistical modeling to evaluate the fit of various models. The final optimal order values for each experiment are listed in Table 4.

Table 4: Optimal Order Values for ARIMA

	Type 1 Attack - US			Type 2 Attack - US		Type 1 Attack - UK	
Parameters	$\tau = 10$	$\tau = 15$	$\tau = 5$	$\tau = 10$	$\tau = 15$	$\tau = 15$	$\tau = 30$
p	11	8	10	9	8	7	4
d	0	0	0	0	0	0	0
q	4	4	4	4	4	4	4

Our developed ARIMA model formula is derived in the following equation.

$$c_i = \mu + \varphi_1 * c_{i-1} + \varphi_2 * c_{i-2} + \varphi_3 * c_{i-3} + \varphi_4 * c_{i-4} + \theta_1 * \varepsilon_{i-1} + \theta_2 * \varepsilon_{i-2} + \theta_3 * \varepsilon_{i-3} + \theta_4 * \varepsilon_{i-4} + \varepsilon_i. \quad (4.2)$$

In this equation, c_i denotes the predicted attack count for the next time window, μ is the intercept or constant term. Different φ variables are the coefficients corresponding to the autoregressive terms that represent the weights assigned to the past attack volumes. Multiple θ variables are the coefficients corresponding to the moving average, which represent the weights assigned to the lagged forecast errors denoted by ε_{i-1} , ε_{i-2} , ε_{i-3} and ε_{i-4} . ε_i is the white noise or error term. Using the training data, the specific values of μ , φ and θ are estimated during the model fitting process. These values are obtained by minimizing error or maximizing the probability of the observed data.

4.3.4 Deep Learning Model Development

In addition to machine learning, deep learning models play a significant role in time series analysis; hence we incorporated the two most used deep learning models. To optimize the developed architecture of deep learning models, including the hyperparameter values, we have implemented a number of modifications and performance measurements. The optimized designed RNN and DNN architectures will be discussed more in the sections below.

4.3.4.1 Recurrent Neural Network Model Development

In order to process time series data, we designed RNN models with an input layer of shape (Sequence Length,1) where Sequence Length=4. The implemented models use a recurrent layer known as Simple RNN, which is capable of handling sequential data by maintaining an internal state. Data sequences from the input layer are used to perform a basic recurrent transformation. The models include single or multiple dense layers after the Simple RNN layer to perform non-linear transformations and capture complex patterns in the data. Finally, the output layer consists of a single dense layer with one unit that predicts the next value in the time series. We have compiled the models implementing the Adam optimizer with a learning rate of 0.01 and the mean squared error loss function. Table 9 through Table 13 illustrate the developed RNN Architectures.

Table 9: Architecture of RNN for Type 1 Attack in US ($\tau = 10$)

Layer Type	Output Shape	Number of Units	Activation	Regularization
Input Layer	(Sequence Length,1)	-	-	
Simple RNN	(50,)	50	ReLU	-
Dense	(25,)	25	-	-
Dense	(25,)	25	-	-
Dense	(15,)	15	-	-
Dense	(1,)	1	-	-

Table 10: Architecture of RNN for Type 2 Attack in US ($\tau = 10$)

Layer Type	Output Shape	Number of Units	Activation	Regularization
Input Layer	(Sequence Length,1)	-	-	
Simple RNN	(50,)	50	ReLU	L2 Regularization (0.0001)
Dense	(1,)	1	-	-

Table 11: Architecture of RNN for Type 1 and Type 2 Attack in US ($\tau = 15$)

Layer Type	Output Shape	Number of Units	Activation	Regularization
Input Layer	(Sequence Length,1)	-	-	
Simple RNN	(50,)	50	ReLU	-
Dense	(25,)	25	-	-
Dense	(1,)	1	-	-

Table 12: Architecture of RNN for Type 1 Attack in US ($\tau = 5$) and UK ($\tau = 15$)

Layer Type	Output Shape	Number of Units	Activation	Regularization
Input Layer	(Sequence Length,1)	-	-	
Simple RNN	(50,)	50	ReLU	L2 Regularization (0.001)
Dense	(25,)	25	-	-
Dense	(15,)	15	-	-
Dense	(1,)	1	-	-

Table 13: Architecture of RNN for Type 1 Attack in UK ($\tau = 30$)

Layer Type	Output Shape	Number of Units	Activation	Regularization
Input Layer	(Sequence Length,1)	-	-	
Simple RNN	(50,)	50	ReLU	L2 Regularization (0.001)
Dense	(25,)	25	-	-
Dense	(1,)	1	-	-

4.3.4.2 Deep Neural Network Model Development

Similar to RNN, to process time series data, we developed DNN models with an input layer of shape (Sequence Length,1). A convolutional layer is included in the developed models to perform convolutional operations by sliding the filters across the input, capturing local patterns and features. A flatten layer has been added to reshape the output feature maps from the convolutional layer into a 1-dimensional array. The flatten layer combines all the features into a single vector that serves as input for subsequent dense layers. Correspondingly, multiple dense layers are included based on the performance of the models. The output layer consists of a single unit that produces the final output of the model, representing the forecasted value for the time series.

Table 14: Architecture of DNN for Type 1 Attack in US ($\tau = 10,15$)

Layer Type	Filter	Kernel Size	Number of Units	Activation	Regularization
Conv1D	64	2	-	ReLU	-
Flatten	-	-	-	-	-
Dense	-	-	50	ReLU	-
Dense	-	-	25	ReLU	-
Dense	-	-	15	ReLU	-
Dense	-	-	1	-	-

Table 15: Architecture of DNN for Type 2 Attack in US ($\tau = 10$) and Type 1 Attack in UK ($\tau = 15$)

Layer Type	Filter	Kernel Size	Number of Units	Activation	Regularization
Conv1D	64	2	-	ReLU	-
Flatten	-	-	-	-	-
Dense	-	-	50	ReLU	L2 Regularization (0.01)
Dense	-	-	25	ReLU	-
Dense	-	-	15	ReLU	-
Dense	-	-	1	-	-

Table 16: Architecture of DNN for Type 2 Attack in US ($\tau = 15$)

Layer Type	Filter	Kernel Size	Number of Units	Activation	Regularization
Conv1D	64	2	-	ReLU	-
Flatten	-	-	-	-	-
Dense	-	-	50	ReLU	L2 Regularization (0.01)
Dense	-	-	25	ReLU	-
Dense	-	-	10	ReLU	-
Dense	-	-	1	-	-

Table 17: Architecture of DNN for Type 1 Attack in UK ($\tau = 30$)

Layer Type	Filter	Kernel Size	Number of Units	Activation	Regularization
Conv1D	64	2	-	ReLU	-
Flatten	-	-	-	-	-
Dense	-	-	50	ReLU	L2 Regularization (0.01)
Dense	-	-	25	ReLU	-
Dense	-	-	15	ReLU	-
Dense	-	-	10	ReLU	-
Dense	-	-	1	-	-

4.4 Evaluation Criteria

A benchmark, standard, or factor against which the performance of a machine learning model is measured is referred to as evaluation criteria. The following describes the criteria to assess our implemented models.

4.4.1 Mean Absolute Percentage Error

A common way of measuring forecasting performance is with the mean absolute percentage error (MAPE). Due to its very intuitive interpretation in terms of relative error, mean absolute percentage error is frequently used as a loss function for regression problems and model evaluation. It usually expresses accuracy as a ratio defined by the formula given below:

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \dots \dots \dots (4.3) [71]$$

The actual value is denoted by A_t , while the forecasted value is denoted by F_t in this equation. The difference between them is then divided by the value A_t itself. The absolute value of this ratio is calculated by adding it up for every point in time that has been projected, then dividing that total by the number of points (n) that have been fitted.

4.4.2 Root Mean Square Error

Another standard way to evaluate a prediction accuracy is by determining the RMSE, which measures the mean square deviation. The method for calculating the RMSE involves first determining the residual value of the prediction from each data point, then calculating the mean value by summing the squares of residuals, and finally computing the square root of the mean value. The equation for RMSE can be defined as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y(i) - p(i))^2}{N}} \dots \dots \dots (4.4) [72]$$

In the above equation, $y(i)$ is the i -th value, where $p(i)$ is the corresponding prediction, and N is the number of data points.

4.5 Power BI Analysis for Cyber-attack Data Trend

Microsoft Power BI platform is a cloud-based data analysis tool that can be utilized as a standalone application or as part of the institutional Power Platform to analyze and report on data. Its Report Server can be used to manage data working as standalone software. The Power BI embedded "REST API" is also useful in allowing other applications to integrate with the platform services. The platform offers three different versions of its Power BI, which include Power BI Mobile, Power BI Desktop, and Power BI Service. We have developed a power BI platform to enable the analysis of recent cyber-attack data to gain insights into cyber-attack trends and to comprehend the interdependent factors of cyber-attack.

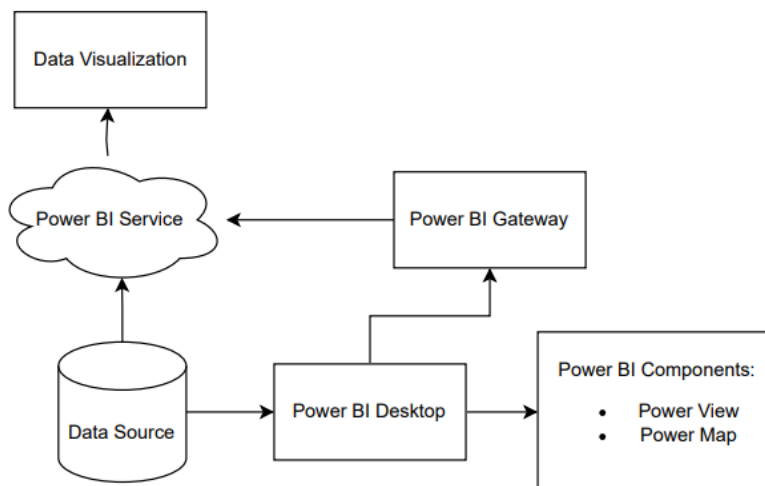


Figure 10: High-level View of Cyber-attack Trend Analysis with Power BI

Figure 10 illustrates a high-level view of our global cyber-attack trend analysis using Power BI. We have started with Power BI Desktop, which is a development tool for Power Query, Power Pivot, and Power View. With Power BI Desktop, we have Power BI components within the same solution, and by implementing the components, we have developed visualization for cyber-attack trend analysis. We have used Power View and Power Map, two of the useful Power BI components, to develop our dashboard. With Power View, we have created interactive charts and other visual effects from our data by connecting them to our data source. Power Map has been applied to create a visual representation of our data by plotting and mapping it in a 3D format which allows visualizing different data types based on geographical locations, such as source and destination country for cyber-attack. The Power BI gateway has ensured our secure data transfer between on-premises and cloud-based services, providing a reliable and efficient solution. Finally, to access and share the Power BI reports from anywhere with an internet connection, we have connected the Power BI desktop to the Power BI service, which is also known as Power BI online.

Figure 11 shows the overview of our dashboard for global cyber-attack trend analysis which contains the key features we created to better analyze the data we gathered.



Figure 11: Dashboard for Global Cyber-attack Trend Analysis

The key features of the developed Power BI system are listed below -

1. **Statistical Analysis of Cyber-attack:** To get the statistics on the current data, we have plotted some graphs using the “Build visual” feature of Power BI. With the help of the interactive dashboard (Figure 11), the “Top 3” and “Top 10” page navigation options take us to the pages showing the top 3 and top 10 cyber-attack statistics, as shown in Figure 12.

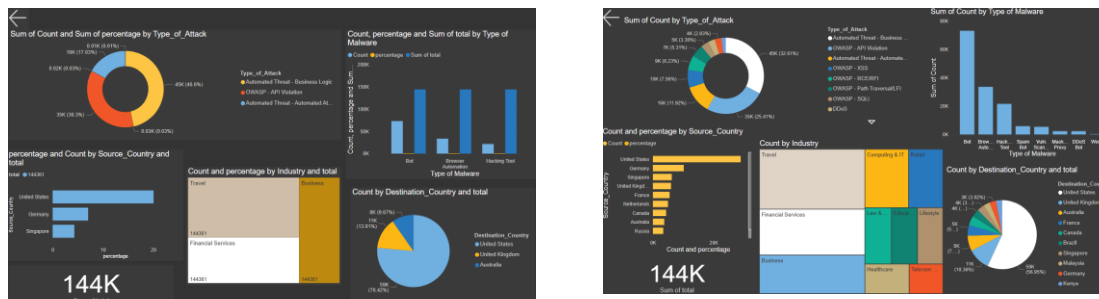


Figure 12: Top 3 and Top 10 Analysis on Current Cyber-attack Data

2. **Map Visualization of Cyber-attack:** We have used the "Build Visual Map" feature of Power BI to show geographic-based cyber-attack patterns based on source and destination country. This feature will allow users to select a specific country that will filter out the corresponding threatened destination or source countries.

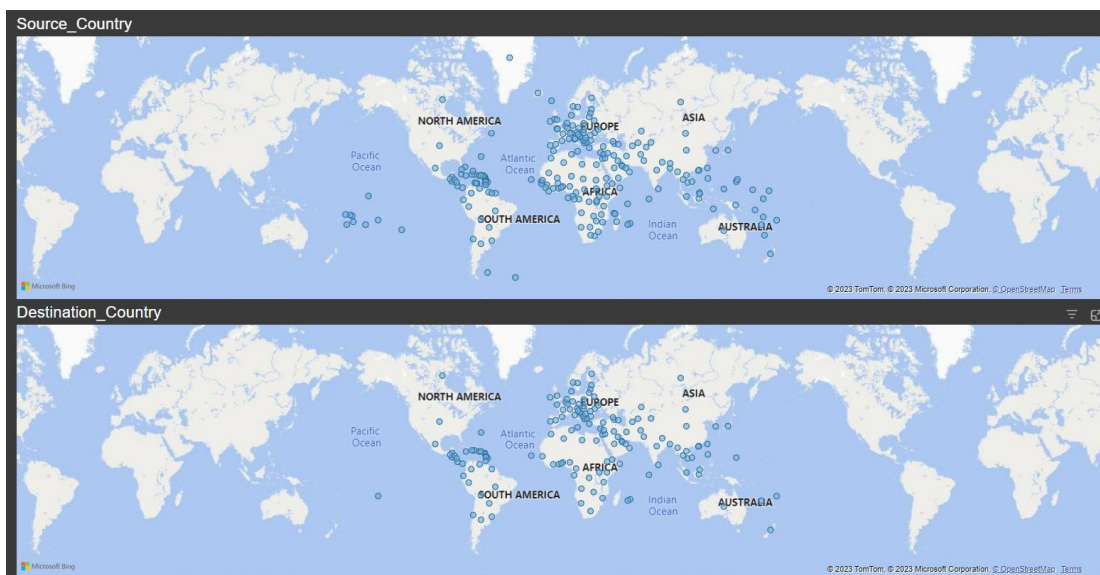


Figure 13: Map Visualization for Countries of Origin Launching Attacks on Target Countries

3. Filtering Mechanism for Corresponding Cyber-attack: To visualize corresponding cyber-attack information for a specific selection of data, we have implemented a feature of Power BI where a filtering process is triggered upon selecting particular data point that displays the corresponding table with unique data while excluding non-relevant information. Furthermore, we have also applied a time frame filter by which we can analyze the cyber-attack trends in specific periods. Figure 14 shows a page of our developed system containing cyber-attack information before any specific entity selection and Figure 15 shows the same page after selecting “Bangladesh” as the destination country with time frame for first two days of data collection. Notably, after selecting “Bangladesh” as the destination country, we can view five Industries, four attack categories, four types of malwares, and corresponding source countries within the selected time frame.

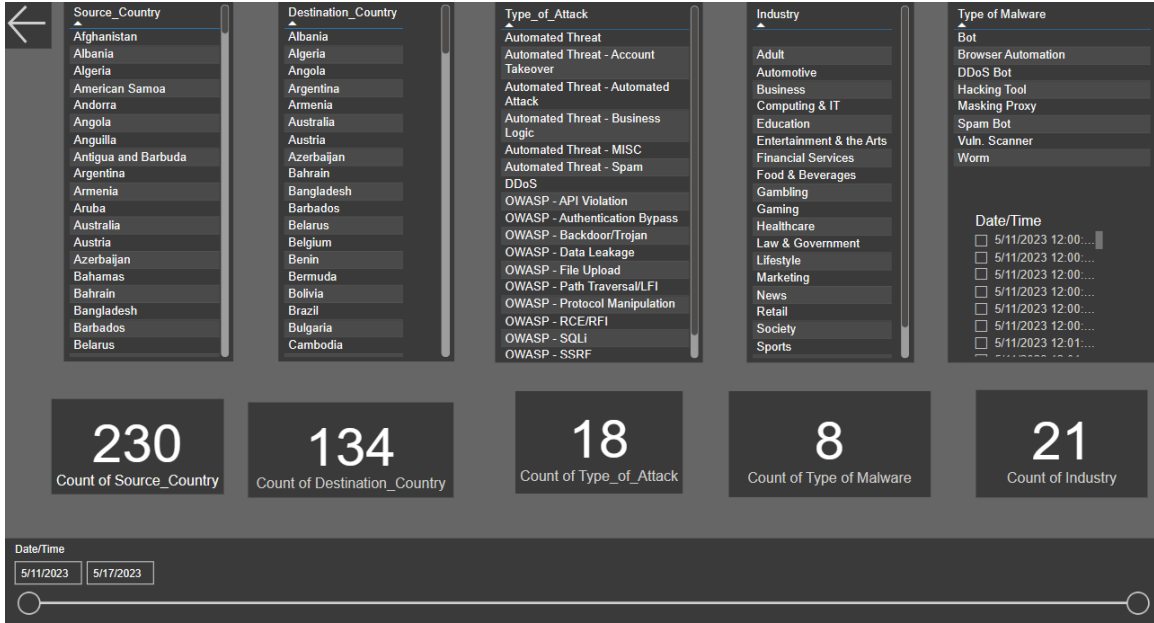


Figure 14: Page with Filtering Mechanism for Corresponding Cyber-attack Information before any selection

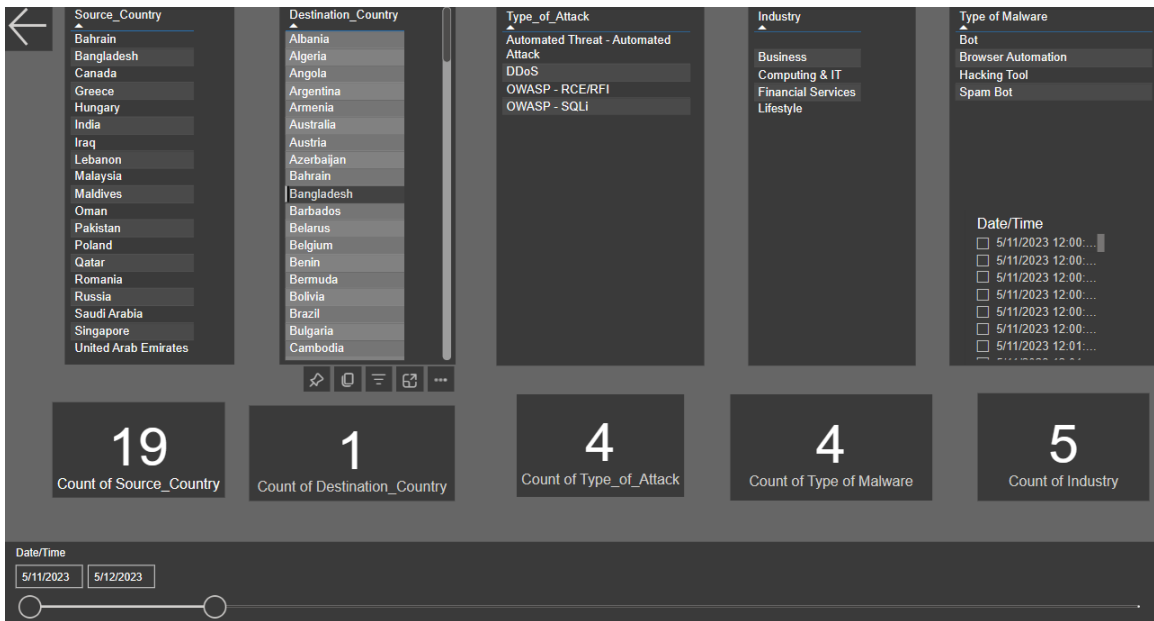


Figure 15: Page with Filtering Mechanism for Corresponding Cyber-attack Information after Destination Country and Time Range Selection

- 4. Enabling Data Updates with On-Premises Data Gateway:** To update data in a cloud service without having to access the Power BI Desktop, we have used an on-premises data gateway. This solution allowed us to synchronize data between the cloud and local sources, and it can be particularly beneficial in scenarios where accessing the desktop is not feasible. With the use of an on-premises gateway, we can easily update our data in the cloud solution without having to perform a manual intervention. The gateway solution has eliminated the need to perform various tasks manually and improved the efficiency of our data management operations.

Finally, we can conclude that our developed Power Bi platform can be utilised in real time; more specifically, it can be deployed in addition to our predictive modeling system that can help an organisation understand which cyber-attacks and industries to prioritise in certain cases by observing the attack statistics. After learning about high-priority cyber attacks in the present period, organisations can utilise the specific predictive model developed to forecast the severity of that particular attack in the future period to take preventive steps.

4.6 Development Environment

For model development and execution, we used our Windows 11 pro server- 64-bit operating system, x64-based processor 12th Gen Intel(R) Core (TM) i9-12900K 3.20 GHz, RAM- 128 GB. For Data collection, we utilized windows 10 PC 64-bit operating system, x64-based processor - Intel(R) Core (TM) i5-7500T CPU @ 2.70GHz 2.71 GHz. We have used pandas, NumPy, scikit-learn, matplotlib, seaborn libraries for data analysis, preprocessing and model implementation. Moreover, Power BI has been used for cyber-attack trend analysis.

Chapter 5

5 Results and Discussion

This chapter will provide a brief overview of our experiments and results throughout the development of the system for global cyber-attack pattern analysis, including comparative and explanatory analysis of overall findings for different techniques and scenarios.

5.1 Performance Analysis of Developed Models

As described in Chapter 4, we began our experiment by developing a baseline model and eventually progressed to incorporating machine learning and deep learning models for different time frames, top two attacked countries, and attack types. We will primarily describe our performance analysis based on the MAPE value, which focuses on the error percentage difference. We have also provided an RMSE score to highlight the squared difference between error numbers. However, in some circumstances, the MAPE decreases while the RMSE grows, and vice versa. This can occur when the model improves in terms of capturing the percentage difference between anticipated and actual values while introducing larger absolute errors in certain cases.

Table 18 through Table 24 indicate that, on average, machine learning models achieve a significant improvement in MAPE value of 15% or above compared to baseline model, indicating substantial progress. Therefore, deep learning models outperform their machine learning counterparts. After observing a significant improvement with an increase in the time window (τ) from 10 to 15 minutes, we implemented $\tau=5$ minutes to determine if performance varies with the time frame. The overall performance degrades for $\tau=5$ minutes, as shown in Table 22; however, the pattern of improvement from the baseline model to machine learning and deep learning models remains consistent. Afterward, observing improved experimental outcomes with $\tau = 15$ minutes for the United States, we conducted our experiment for attack volume prediction in the United Kingdom using the same time frame. The initial performance of our model with the specified τ value was found to be suboptimal. In light of this, we decided to conduct further experimentation by

increasing the τ value to 30. The subsequent evaluation revealed a significant improvement in performance, as demonstrated in Table 24.

Table 18: Evaluation Scores for Type 1 Attack in US ($\tau = 10$)

	Model	MAPE	RMSE
Statistical Model (Baseline Model)	ARIMA (11,0,4)	39.96%	5.18
Machine Learning Model	XGB	24.41%	3.62
	RF	24.56%	3.63
	SVR	24.01%	3.75
	KNR	24.66%	3.58
Deep Learning Model	RNN	23.19%	3.70
	DNN	23.63%	3.69

Table 19: Evaluation Scores for Type 2 Attack in US ($\tau = 10$)

	Model	MAPE	RMSE
Statistical Model (Baseline Model)	ARIMA	41.35%	4.99
Machine Learning Model	XGB	26.00%	3.72
	RF	26.01%	3.79
	SVR	26.08%	3.67
	KNR	26.22%	3.82
Deep Learning Model	RNN	24.99%	3.83
	DNN	24.74%	3.68

Table 20: Evaluation Scores for Type 1 Attack in US ($\tau = 15$)

	Model	MAPE	RMSE
Statistical Model (Baseline Model)	ARIMA	35.85%	7.09
Machine Learning Model	XGB	17.77%	4.10
	RF	17.54%	4.15
	SVR	19.00%	4.34
	KNR	17.91%	4.18
Deep Learning Model	RNN	17.11%	4.11
	DNN	16.77%	4.11

Table 21: Evaluation Scores for Type 2 Attack in US ($\tau = 15$)

	Model	MAPE	RMSE
Statistical Model (Baseline Model)	ARIMA	33.84%	6.36
Machine Learning Model	XGB	21.29%	4.74
	RF	21.08%	4.73
	SVR	21.25%	4.72
	KNR	21.10%	4.88
Deep Learning Model	RNN	20.69%	4.88
	DNN	19.93%	4.56

Table 22: Evaluation Scores for Type 1 Attack in US ($\tau = 5$)

	Model	MAPE	RMSE
Statistical Model (Baseline Model)	ARIMA	54.73%	3.11
Machine Learning Model	XGB	43.04%	2.74
	RF	41.17%	2.64
	SVR	43.23%	2.34
	KNR	42.18%	2.70
Deep Learning Model	RNN	38.27%	2.54
	DNN	40.13%	2.59

Table 23: Evaluation Scores for Type 1 Attack in UK ($\tau = 15$)

	Model	MAPE	RMSE
Statistical Model (Baseline Model)	ARIMA	62.47%	3.55
Machine Learning Model	XGB	38.46%	2.53
	RF	37.57%	2.59
	SVR	34.74%	2.44
	KNR	37.71%	2.56
Deep Learning Model	RNN	33.81%	2.42
	DNN	31.98%	2.41

Table 24: Evaluation Scores for Type 1 Attack in UK ($\tau = 30$)

	Model	MAPE	RMSE
Statistical Model (Baseline Model)	ARIMA	44.95%	5.94
Machine Learning Model	XGB	26.29%	4.39
	RF	25.37%	4.21
	SVR	24.67%	3.89
	KNR	24.86%	4.32
Deep Learning Model	RNN	22.51%	3.83
	DNN	20.99%	4.16

Table 25 and Table 26 display the actual and predicted values for various experimental conditions. To clearly present the comparison analysis, we have only included the plots for the top performing ML and DL models in Table 26. Compared to the baseline model, which is a nearly horizontal line, our developed models follow the trends in the volume of cyber incidents quite well but are unable to precisely predict the number of attacks. Comparing the baseline models to our models demonstrates that the approach to predicting the intensity of a cyber incident based on the collected dataset has merit and requires further research and exploration. In addition, in order to ensure that our deep learning model does not overfit on the training data, we experimented with various architectures by analyzing the learning curve, as shown in Table 27, and selected the most appropriate one. The learning curves show that the validation loss over each epoch is less than or equal to the training loss which signifies that our model is not overfitting.

Table 25: Actual Value vs Predicted Value Plots for ARIMA Model

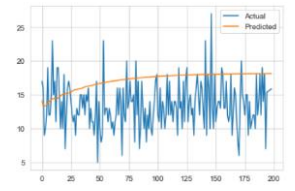
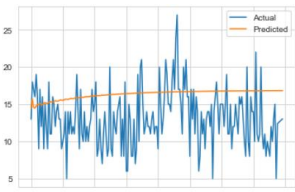
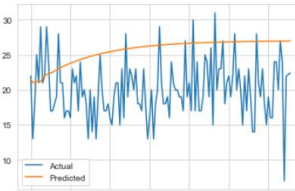
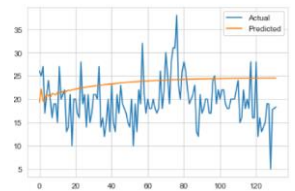
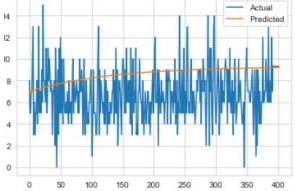
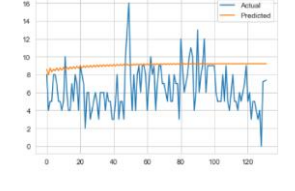
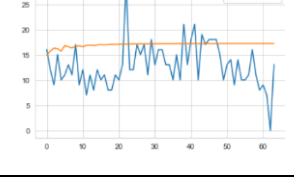
	ARIMA Model (Baseline Model)
$\tau = 10$ US Type 1 Attack	
$\tau = 10$ US Type 2 Attack	
$\tau = 15$ US Type 1 Attack	
$\tau = 15$ US Type 2 Attack	
$\tau = 5$ US Type 1 Attack	
$\tau = 15$ UK Type 1 Attack	
$\tau = 30$ UK Type 1 Attack	

Table 26: Actual Value vs Predicted Value Plots for Best Performing ML and DL Models

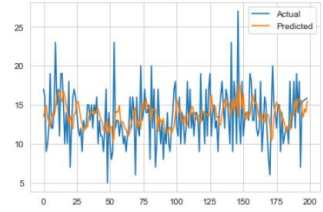
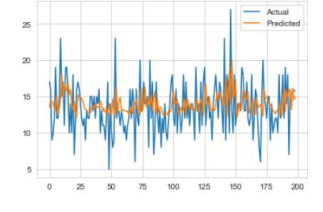


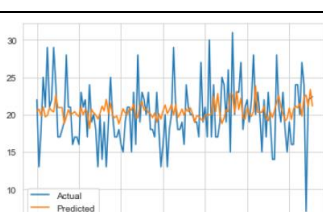
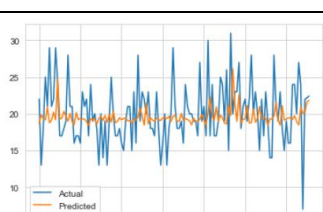
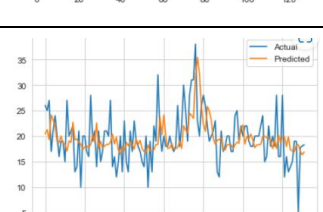
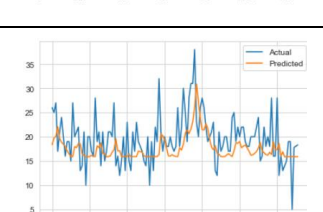
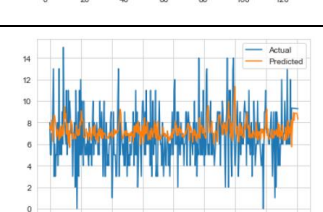
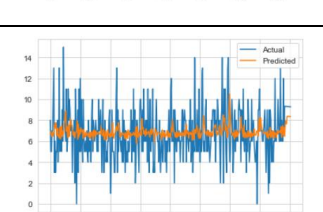
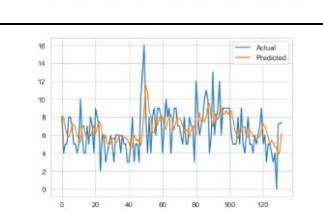
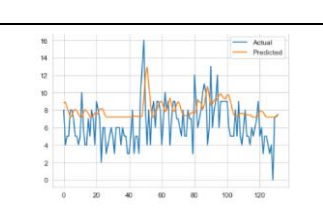
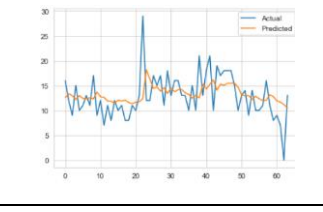
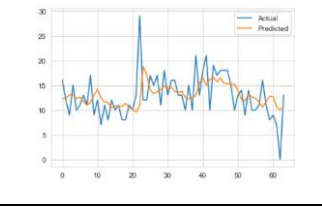
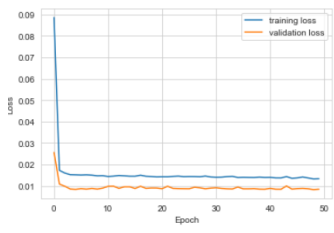
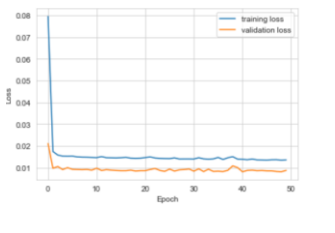
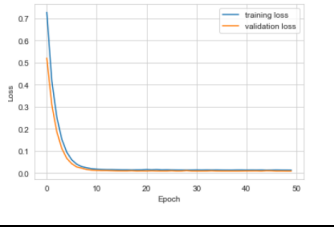
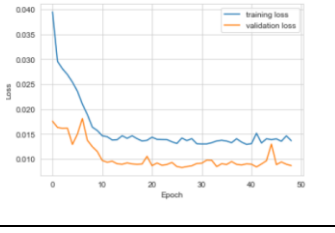
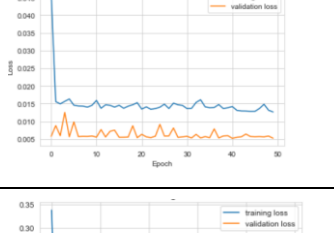
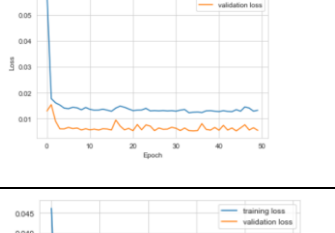
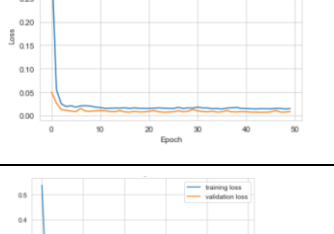
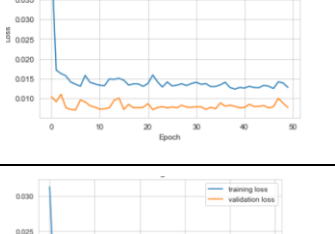
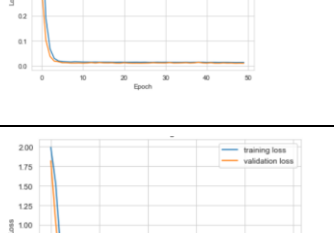
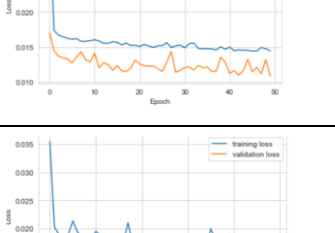
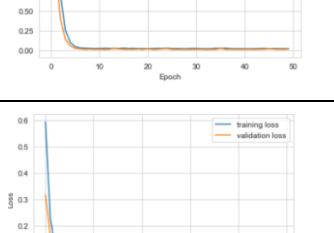
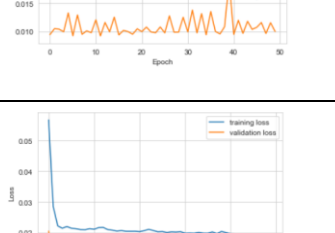
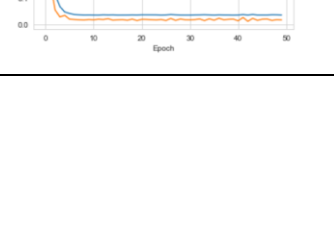
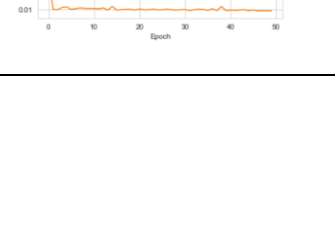
	Best Machine Learning Model	Best Deep learning Model
$\tau = 10$ US Type 1 Attack		
$\tau = 10$ US Type 2 Attack		
$\tau = 15$ US Type 1 Attack		
$\tau = 15$ US Type 2 Attack		
$\tau = 5$ US Type 1 Attack		
$\tau = 15$ UK Type 1 Attack		
$\tau = 30$ UK Type 1 Attack		

Table 27: Learning Curve for Developed DL Models

	DNN Learning Curve	RNN Learning Curve
$\tau = 10$ US Type 1 Attack		
$\tau = 10$ US Type 2 Attack		
$\tau = 15$ US Type 1 Attack		
$\tau = 15$ US Type 2 Attack		
$\tau = 5$ US Type 1 Attack		
$\tau = 15$ UK Type 1 Attack		
$\tau = 30$ UK Type 1 Attack		

5.2 Explanatory Analysis of Overall Results and Findings

We have observed a strong relationship between the autocorrelation (ρ) values and the effectiveness of our time series dataset analysis. In particular, we identified that increasing the autocorrelation value enhanced model performance. In addition, we found that the correlation increases as the value of the time window (τ) increases. This result can be attributed to the fact that a larger time frame reveals the variation of attack volume over time more clearly. This increased clarity enables the model to learn and recognize underlying data patterns more readily. Figure 16 displays the ρ values at lag 4 for various examinations, as well as their corresponding MAPE values. The results demonstrate conclusively that the MAPE values for distinct scenarios decrease as the autocorrelation increases, as this increased correlation consequently improves the model's ability to precisely predict future values. Furthermore, as mentioned before, it has been observed that there exists a positive correlation between the autocorrelation value and the corresponding value of τ .

These results demonstrate the importance of including the time window parameter in our analysis. By selecting an appropriate time window, we can enhance the performance of the model by capitalizing on the increased correlation between various time lags. Overall, our findings indicate that a higher autocorrelation value, in conjunction with a larger time window, strengthens the predictive ability of the model and the precision of our analysis.

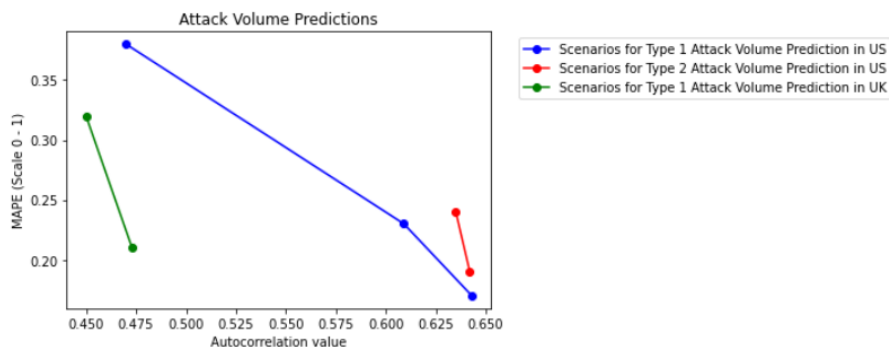


Figure 16: MAPE vs Autocorrelation Plot

Chapter 6

6 Conclusion

Machine learning is becoming increasingly crucial in cyber-attack pattern analysis, therefore, can be used to detect cyber anomalies and attacks and to provide security services in an IoT network [73] [74]. Predictive models can be developed using machine learning to identify potential cyber-attacks before they happen. These models can analyze historical data to predict the likelihood of a particular attack happening.

We aimed to contribute to the cyber security field by analyzing global cyber-attack patterns using AI techniques, including current global cyber-attack data. This thesis began by examining the current state of cyber security analysis and identifying several shortcomings. In contrast to related works, our research collected recent data from an up-to-date website, and our data collection framework can continue to collect the latest updated data, providing a valuable resource for future cyber-attack analysis.

Furthermore, we implemented the Power BI tool to visualize current global cyber-attack trends, making it easier to identify patterns and trends in cyber-attacks. Unlike most current research, which focuses on determining the presence or type of a cyber-attack from the pattern of an incident, our research focuses on predicting the severity of cyber-attacks in a future period based on counts of previous incidents.

We have leveraged various optimization methods, such as random and grid searching, to optimize our hyperparameters to ensure the highest accuracy in deep learning and machine learning models. These experiments have allowed us to identify the ideal parameters and produce a promising outcome.

In conclusion, our research contributes to the field of cyber security analysis by providing an up-to-date and valuable data resource, a cyber-attack trend visualization platform, and novel prediction techniques.

6.1 Limitation and Future Work

We encountered a limitation in our data collection technique, which required us to restart the data collection program almost after every few hours. In the future, we plan to improve our data collection technique to automatically enable continuous data collection for a more extended period for the analysis of cyber-attack patterns.

While our model performed well with the collected dataset, it may not be sufficient to make predictions for other types of datasets. In future work, we plan to integrate our data collection framework with different live data resources to improve the scope of our research. However, this integration is currently limited by website restrictions on web scraping.

We evaluated our model using various evaluation metrics, but we acknowledge that this may not be sufficient to prove its robustness. In future work, we intend to perform further evaluation to establish the robustness of our model.

We have completed our experiment on one-step forecasting, including the top two countries and attack types, and going forward, one of our goals is to integrate multistep forecasting, additionally involving more countries and attack types to offer greater flexibility for cyber security professionals, allowing for more accurate predictions of future cyber attacks and enabling proactive measures to be taken.

We have performed our analysis for smaller time frames; however, in the future, when we have access to a greater quantity of data, we would like to contribute to daily or monthly analysis. This will allow us to provide more comprehensive insights into the trends and patterns of the data.

Finally, to reduce our prediction error percentage, we will experiment with multivariate forecasting because multivariate forecasting leverages information from multiple related variables that can capture complex interdependencies and correlations among variables, leading to more accurate predictions. In the case of multivariate forecasting, we can integrate additional dataset features that we did not include in our predictive modelling methodologies. Additionally, as we have performed our analysis for smaller time frames;

in the future, when we have access to a greater quantity of data, we would like to contribute to daily or monthly analysis. The larger dataset will allow our models to train more effectively by capturing more information from the dataset, and we have established from our experimental results that expanding the time frame has the potential to increase model performance.

Bibliography

1. Apruzzese G, Colajanni M, Ferretti L, Guido A, Marchetti M. On the effectiveness of machine and deep learning for cyber security. In: International Conference on Cyber Conflict, CYCON. NATO CCD COE Publications; 2018. p. 371–89.
2. Kotsias J, Ahmad A, Scheepers R. Adopting and integrating cyber-threat intelligence in a commercial organisation. *European Journal of Information Systems*. 2023;32(1):35–51.
3. Dhawan SM, Gupta BM, Elango B. Global Cyber Security Research Output (1998–2019): A Scientometric Analysis. *Sci Technol Libr (New York, NY)*. 2021;40(2):172–89.
4. Cyber Research Areas of Interest – The Cyber Team [Internet]. 2023 [cited 2023 Feb 14]. Available from: <https://coar.risc.anl.gov/research/>
5. Jovanovic B. Better Safe Than Sorry: Cyber Security Statistics and Trends for 2023 [Internet]. 2023 [cited 2023 Apr 13]. Available from: <https://dataprot.net/statistics/cyber-security-statistics/>
6. James N. 90+ Cyber Crime Statistics 2023: Cost, Industries & Trends [Internet]. 2023 [cited 2023 Apr 13]. Available from: <https://www.getastra.com/blog/security-audit/cyber-crime-statistics/>

7. Brooks C. Cybersecurity Trends & Statistics For 2023; What You Need To Know [Internet]. [cited 2023 Apr 13]. Available from: <https://www.forbes.com/sites/chuckbrooks/2023/03/05/cybersecurity-trends--statistics-for-2023-more-treachery-and-risk-ahead-as-attack-surface-and-hacker-capabilities-grow/?sh=6bb5ea0319db>
8. James N. 160 Cybersecurity Statistics: Updated Report 2023 [Internet]. 2023 [cited 2023 Apr 13]. Available from: <https://www.getastra.com/blog/security-audit/cyber-security-statistics/>
9. Li Y, Liu Q. A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. Energy Reports. 2021 Nov 1;7:8176–86.
10. Pan X. Quantitative Analysis and Prediction of Global Terrorist Attacks Based on Machine Learning. Sci Program. 2021;15:1–5.
11. Kent S. Global Trends [Internet]. 2021 [cited 2023 Aug 23]. Available from: https://www.dni.gov/files/ODNI/documents/assessments/GlobalTrends_2040.pdf
12. Foro Económico Mundial., Marsh & McLennan., SK Group., Zurich Insurance Group. The global risks report 2022. World Economic Forum; 2022.
13. How AI and Machine Learning Are Improving Cybersecurity | SailPoint [Internet]. 2022 [cited 2023 Feb 14]. Available from: <https://www.sailpoint.com/identity-library/how-ai-and-machine-learning-are-improving-cybersecurity/>
14. Abdulkarem HS, Alethawy AD. DDOS ATTACK DETECTION AND MITIGATION AT SDN ENVIROMENT [Internet]. Vol. 4, Iraqi Journal of Information and Communications Technology(IJICT). 2021. Available from: <https://ijict.edu.iq>
15. What is a distributed denial-of-service (DDoS) attack? | Cloudflare [Internet]. [cited 2023 Feb 14]. Available from: <https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/>

16. API attack | Radware [Internet]. [cited 2023 Apr 21]. Available from: <https://www.radware.com/cyberpedia/application-security/api-attack/>
17. What is Remote Code Execution (RCE)? - Check Point Software [Internet]. 2022 [cited 2023 Apr 21]. Available from: <https://www.checkpoint.com/cyber-hub/cyber-security/what-is-remote-code-execution-rce/>
18. Essex D. What is RFI (request for information)? [Internet]. [cited 2023 Apr 21]. Available from: <https://www.techtarget.com/searcherp/definition/RFI-request-for-information>
19. What Is Path Traversal and How Does It Work? [Internet]. [cited 2023 Apr 21]. Available from: <https://www.synopsys.com/glossary/what-is-path-traversal.html>
20. Adejumola R. What Are Local File Intrusion (LFI) Attacks and Should You Be Worried? [Internet]. 2023 [cited 2023 Apr 21]. Available from: <https://www.makeuseof.com/what-are-local-file-intrusion-attacks/>
21. Watson C. OWASP Automated Threat Handbook. OWASP [Internet]. 2015 Oct 26 [cited 2023 Feb 14]; Available from: <https://www.owasp.org/images/3/33/Automated-threat-handbook.pdf>
22. What is a Business Logic Attack | How to Prevent It? | Netacea [Internet]. 2021 [cited 2023 Feb 14]. Available from: <https://netacea.com/glossary/business-logic-attack/>
23. Kirsten. Cross Site Scripting (XSS) | OWASP Foundation [Internet]. [cited 2023 Apr 21]. Available from: <https://owasp.org/www-community/attacks/xss/>
24. What is SQL Injection? Tutorial & Examples | Web Security Academy [Internet]. [cited 2023 Apr 21]. Available from: <https://portswigger.net/web-security/sql-injection>

25. What is a Data Breach | Tips for Data Leak Prevention | Imperva [Internet]. [cited 2023 Apr 21]. Available from: <https://www.imperva.com/learn/data-security/data-breach/>
26. Kothari N, Mahajan R, Millstein T, Govindan R, Musuvathi M. Finding protocol manipulation attacks. In: Proceedings of the ACM SIGCOMM 2011 Conference, SIGCOMM'11. 2011. p. 26–37.
27. Backdoor Trojan - Firewalls.com [Internet]. [cited 2023 Apr 21]. Available from: <https://www.firewalls.com/blog/security-terms/backdoor-trojan/>
28. How to fix file upload vulnerability of web applications | Web pentesting | VAPT Pentesting Services | cyber security whitepapers | Pune Mumbai Hyderabad Delhi Bangalore Ahmedabad Kolkata India Dubai Bahrain Qatar Kuwait Singapore Australia USA UK Germany Croatia Botswana Mauritius [Internet]. [cited 2023 Apr 21]. Available from: <https://www.valencynetworks.com/kb/file-upload-vulnerability-attacks.html>
29. Server-Side Request Forgery (SSRF) [Internet]. [cited 2023 Apr 21]. Available from: <https://www.imperva.com/learn/application-security/server-side-request-forgery-ssrf/>
30. What Is Authentication Bypass Vulnerability? How To Prevent It? - The Sec Master [Internet]. [cited 2023 Apr 21]. Available from: <https://thesecmaster.com/what-is-authentication-bypass-vulnerability-how-to-prevent-it/>
31. Time series analysis in cybersecurity [Internet]. 2023 [cited 2023 Jun 9]. Available from: <https://www.oreilly.com/library/view/hands-on-machine-learning>
32. Autocorrelation in Time Series Data [Internet]. 2019 [cited 2023 Jun 29]. Available from: <https://www.influxdata.com/blog/autocorrelation-in-time-series-data/>

33. Smith T. Autocorrelation: What It Is, How It Works, Tests [Internet]. 2023 [cited 2023 Jun 9]. Available from:
<https://www.investopedia.com/terms/a/autocorrelation.asp>
34. Bodner K, Fortin MJ, Molnár PK. Making predictive modelling ART: accurate, reliable, and transparent. *Ecosphere*. 2020 Jun 1;11(6).
35. What is Predictive Analytics? An enterprise guide [Internet]. [cited 2023 Jun 11]. Available from:
<https://www.techtarget.com/searchenterpriseai/definition/predictive-modeling>
36. Statistical model [Internet]. [cited 2023 Jun 11]. Available from:
https://en.wikipedia.org/wiki/Statistical_model
37. Hayes A. Autoregressive Integrated Moving Average (ARIMA) Prediction Model [Internet]. [cited 2023 Jun 11]. Available from:
<https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp>
38. Banerjee P. A Guide on XGBoost hyperparameters tuning [Internet]. [cited 2023 Jun 11]. Available from: <https://www.kaggle.com/code/prashant111/a-guide-on-xgboost-hyperparameters-tuning>
39. Shaik AB, Srinivasan S. A brief survey on random forest ensembles in classification model. In: *Lecture Notes in Networks and Systems*. Springer; 2019. p. 253–60.
40. Understanding Random Forest. How the Algorithm Works and Why it Is... | by Tony Yiu | Towards Data Science [Internet]. 2019 [cited 2023 Feb 14]. Available from: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>
41. Random Forest Algorithm for Machine Learning | by Madison Schott | Capital One Tech | Medium [Internet]. 2019 [cited 2023 Feb 14]. Available from: <https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb>

42. Introduction to Random Forest in Machine Learning | Engineering Education (EngEd) Program | Section [Internet]. 2020 [cited 2023 Feb 14]. Available from: <https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/>
43. Raj A. Unlocking the True Power of Support Vector Regression [Internet]. [cited 2023 Jun 12]. Available from: <https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression>
44. Support Vector Regression in Machine Learning [Internet]. Great Learning Team. 2022 [cited 2023 Jun 12]. Available from: <https://www.mygreatlearning.com/blog/support-vector-regression/>
45. Sharp T. An Introduction to Support Vector Regression (SVR). [cited 2023 Jun 12]; Available from: <https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr>
46. Understanding Support Vector Machine Regression [Internet]. 2023 [cited 2023 Jun 12]. Available from: <https://www.mathworks.com/help/stats/understanding-support-vector-machine-regression.html>
47. Miller M. The Basics: KNN for classification and regression [Internet]. [cited 2023 Jun 14]. Available from: <https://towardsdatascience.com/the-basics-knn-for-classification-and-regression>
48. K Nearest Neighbors - Regression [Internet]. [cited 2023 Jun 19]. Available from: https://www.saedsayad.com/k_nearest_neighbors_reg.htm
49. Scott Z. Working with RNNs [Internet]. [cited 2023 Jun 14]. Available from: https://www.tensorflow.org/guide/keras/working_with_rnn
50. Dancker J. A Brief Introduction to Recurrent Neural Networks [Internet]. 2022 [cited 2023 Jun 14]. Available from: <https://towardsdatascience.com/a-brief-introduction-to-recurrent-neural-networks>

51. Awan A. Recurrent Neural Network Tutorial (RNN) [Internet]. 2022 [cited 2023 Jun 14]. Available from: <https://www.datacamp.com/tutorial/tutorial-for-recurrent-neural-network>
52. Saeed M. Understanding Simple Recurrent Neural Networks in Keras [Internet]. 2022 [cited 2023 Jun 14]. Available from: <https://machinelearningmastery.com/understanding-simple-recurrent-neural-networks-in-keras/>
53. Chatterjee C. Implementation of RNN, LSTM, and GRU [Internet]. [cited 2023 Jul 14]. Available from: <https://towardsdatascience.com/implementation-of-rnn-lstm-and-gru-a4250bf6c090>
54. Johnson J. What's a Deep Neural Network? Deep Nets Explained – BMC Software | Blogs [Internet]. [cited 2023 Apr 13]. Available from: <https://www.bmc.com/blogs/deep-neural-network/>
55. Arora T, Sharma M, Khatri SK. Detection of Cyber Crime on Social Media using Random Forest Algorithm. In: 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC). IEEE; 2019. p. 47–51.
56. Azizan AH, Mostafa SA, Mustapha A, Mohd Foozy CF, Abd Wahab MH, Mohammed MA, et al. A machine learning approach for improving the performance of network intrusion detection systems. *Annals of Emerging Technologies in Computing*. 2021;5(Special issue 5):201–8.
57. Elmrabbit N, Zhou F, Li F, Zhou H. Evaluation of Machine Learning Algorithms for Anomaly Detection [Internet]. 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security). 2020. Available from: <https://www2.le.ac.uk/offices/itservices/ithelp/services/hpc>
58. Swarna Priya RM, Maddikunta PKR, Parimala M, Koppu S, Gadekallu TR, Chowdhary CL, et al. An effective feature engineering for DNN using hybrid

- PCA-GWO for intrusion detection in IoMT architecture. *Comput Commun.* 2020 Jul 1;160:139–49.
59. Zhang J, Pan L, Han QL, Chen C, Wen S, Xiang Y. Deep Learning Based Attack Detection for Cyber-Physical System Cybersecurity: A Survey. *IEEE/CAA Journal of Automatica Sinica.* 2022 Mar 1;9(3):377–91.
 60. Sengan S, V S, V I, Velayutham P, Ravi L. Detection of false data cyber-attacks for the assessment of security in smart grid using deep learning. *Computers & Electrical Engineering.* 2021 Jul 1;93:107211.
 61. Al-Abassi A, Karimipour H, Dehghantanha A, Parizi RM. An ensemble deep learning-based cyber-attack detection in industrial control system. *IEEE Access.* 2020;8:83965–73.
 62. Bilen A, Özer AB. Cyber-attack method and perpetrator prediction using machine learning algorithms. *PeerJ Comput Sci.* 2021;7:1–21.
 63. Ben Fredj O, Mihoub A, Krichen M, Cheikhrouhou O, Derhab A. CyberSecurity Attack Prediction: A Deep Learning Approach. In: *ACM International Conference Proceeding Series.* Association for Computing Machinery; 2020.
 64. Ansari MS, Bartoš V, Lee B. GRU-based deep learning approach for network intrusion alert prediction. *Future Generation Computer Systems.* 2022 Mar 1;128:235–47.
 65. Sokol P, Staňa R, Gajdoš A, Pekarčík P. Network security situation awareness forecasting based on statistical approach and neural networks. *Log J IGPL.* 2022 Mar 30;
 66. Yin K, Yang Y, Yao C, Yang J. Long-Term Prediction of Network Security Situation Through the Use of the Transformer-Based Model. *IEEE Access.* 2022;10:56145–57.

67. Live Threat Map | Real-time View of Cyber Attacks | Imperva [Internet]. [cited 2023 Mar 13]. Available from: <https://www.imperva.com/cyber-threat-attack-map/>
68. Sri Shakthi Institute of Engineering and Technology, Institute of Electrical and Electronics Engineers. Madras Section, India Electronics & Semiconductor Association, Institute of Electrical and Electronics Engineers. 2019 International Conference on Computer Communication and Informatics : January 23-25, 2019, Coimbatore, India.
69. ChromeDriver - WebDriver for Chrome - Getting started [Internet]. [cited 2023 Feb 25]. Available from: <https://chromedriver.chromium.org/getting-started>
70. Identifying the orders of AR and MA terms in an ARIMA model [Internet]. [cited 2023 May 27]. Available from: <https://people.duke.edu/~rnau/411arim3.htm>
71. Mean absolute percentage error [Internet]. [cited 2023 May 30]. Available from: https://en.wikipedia.org/wiki/Mean_absolute_percentage_error
72. Root-mean-square deviation [Internet]. [cited 2023 May 30]. Available from: https://en.wikipedia.org/wiki/Root-mean-square_deviation
73. Malathi C, Padmaja IN. Identification of cyber attacks using machine learning in smart IoT networks. Mater Today Proc. 2021 Jul;80:2518–23.
74. Sarker IH. CyberLearning: Effectiveness analysis of machine learning security modeling to detect cyber-anomalies and multi-attacks. Internet of Things (Netherlands). 2021 Jun 1;14:1–5.

Curriculum Vitae

Name: Nusrat Kabir Samia

Post-secondary Education and Degrees: M.Sc. Candidate, Computer Science
The University of Western Ontario
2021-2023

B.Sc., Computer Science
Military Institute of Science & Technology
2015-2020

Honours and Awards: Western Graduate Research Scholarship (WGRS)
2021-2022

Related Work Experience Teaching Assistant
The University of Western Ontario
2021-2023