INCREASING THE PREDICTIVE POTENTIAL OF MACHINE LEARNING MODELS FOR ENHANCING CYBERSECURITY

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Mostofa Kamrul Ahsan

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Program:
Computer Science

November 2020

Fargo, North Dakota

North Dakota State University
Graduate School

Title

INCREASING THE PREDICTIVE POTENTIAL OF MACHINE LEARNING MODELS FOR ENHANCING CYBERSECURITY

By

Mostofa Kamrul Ahsan

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Kendall E. Nygard
Chair

Dr. Pratap Kotala

Dr. Oksana Myronovych

Dr. Mohiuddin Quadir

Approved:

February 17, 2021
Date

Dr. Simone Ludwig
Department Chair

ABSTRACT

Networks have an increasing influence on our modern life, making Cybersecurity an important field of research. Cybersecurity techniques mainly focus on antivirus software, firewalls and intrusion detection systems (IDSs), etc. These techniques protect networks from both internal and external attacks. This research is composed of three different essays. It highlights and improves the applications of machine learning techniques in the Cybersecurity domain. Since the feature size and observations of the cyber incident data are increasing with the growth of internet usage, conventional defense strategies against cyberattacks are getting invalid most of the time.

On the other hand, the applications of machine learning tasks are getting better consistently to prevent cyber risks in a timely manner. For the last decade, machine learning and Cybersecurity has converged to enhance the risk elimination. Since the cyber domain knowledge and adopting machine learning techniques does not align on the same page in case of deployment of data-driven intelligent systems, there are inconsistencies where it is needed to bridge the gap. We have studied the most recent research works in this field and documented the most common issues regarding the implementation of machine learning algorithms in Cybersecurity. According to these findings, we have conducted research and experiments to improve the quality of service and security strength by discovering new approaches.

ACKNOWLEDGEMENTS

At first, I would like to thank The Almighty for providing me with courage, passion, patience, strength, and knowledge to accomplish this research. This 5+ years of journey has been full of challenges, dedication, emotions, and motivation. I am truly indebted to those who guided me in the right direction, mostly who critiqued my plan and provided timely pep-talks when I was tired of this never-ending journey. My deepest thanks to my thesis advisor Dr. Kendall E. Nygard, for his continuous support, direction, and guidance from the beginning to end of this research. His suggestions and recommendations throughout my entire graduate studies increased my critical thinking and experienced me of different domains. During my journey for Ph.D. in Computer Science, I have completed my Masters in Business Administration (MBA), Graduate degree Certification in Statistics, and Graduate degree Certification in Business Analytics. Besides my research in cybersecurity and machine learning techniques, I have gathered real-world experience as an Intern in the Department of Transportation Support Center, which provided a lot of knowledge to find critical problems in an existing trend. Dr. Nygard continuously supported me and pointed me in the right direction during the whole period of my Doctoral journey. Thanks to my dissertation committee member Dr. Pratap Kotala for his critical remarks on different findings of my research. Thanks to my other committee member Dr. Oksana Myronovych, for her continued support and suggestions for my research ideas and especially for remarkable questions. I want to extend my thanks to Dr. Mohiuddin Quadir for serving on my committee. His insights helped me get together the big picture and real-world application of my research. Working in different academic projects and research labs helped me get an overview of diverse domains that reflect my research. I want to thank Dr. Anne Denton for her support and guidance in introducing machine learning research. I would also like to thank the NDSU

Department of Computer Science for teaching assistantship (2014-2015) and provide me travel grants to present my research findings to numerous conferences.

Further, I would like to extend my thanks to my parents and immediate relatives to prosper me with their blessings and inspiration. I want to extend my gratitude to my classmates, who have mentored me on several occasions. My special thank goes to my lovely wife Asma A Aksa who is currently preparing herself for USMLE, for providing me the strength to continue my studies. My mother, who I respect and love a lot, played a significant role in finishing my graduate degrees. Her advice and direction made me who I am today. I want to conclude by extending my thanks to my loving son, “Izyaan” to let me finish my research on time.

DEDICATION

This doctoral disquisition is dedicated to the memory of my grandfather, the late Doctor Mohammad Islam Mandal. He died before I came to this world. His stories and personality inspired me a lot.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. GENERAL INTRODUCTION

The internet has become a basic requirement of modern life. It aids people in many ways, such as education, business, and entertainment, etc. As there are many risks associated with network attacks under the internet environment, various systems are designed to block the cyber-based attacks. The importance of Cybersecurity is rising in which machine learning is becoming increasingly significant. Several machine learning techniques and statistical methods are incorporated with artificial intelligence that is proven effective in this domain to prevent cyber-attacks. It is obvious by multiple reasons that machine learning in Cybersecurity is far more than merely applying well-established methods to datasets of cyber entities. Cybersecurity domain involves machine learning challenges that require efficient methodological and theoretical handling. Detecting security incident patterns or insights from cybersecurity data and building a corresponding data-driven model to prevent an attack, is the key to make a security system intelligent. Understanding and analyzing the actual phenomena with data, various scientific methods, processes, and systems are combined with machine learning techniques to build a robust and effective predictive model. In this research, we focus and briefly discuss on machine learning improvement for cybersecurity data, the sources where data is being gathered from, and the analytics behind the latest data-driven patterns for providing effective security [1]–[5]. The major concern of applying machine learning in Cybersecurity is to make the computational process more accessible and intelligent as compared to traditional ones in the domain of Cybersecurity [4]. Primarily we are going to discuss and summarize several associated research issues and future directions. Furthermore, we documented experiments on the major improvement scopes of machine learning technique and their applications for the purpose of cybersecurity modeling [1], [4]. Overall, our goal is not only to discuss well-established machine

learning algorithms in Cybersecurity and relevant models but also to discover major improvements using machine learning techniques on data-driven intelligent decision making for protecting the systems from cyber-attacks.

1.1. Motivation

Cybersecurity refers to the safeguard of computers or other devices and systems from damage of software, theft of information and other intellectual properties [6], [7]. Since at most every business, government, and people are data driven now a days, cybersecurity is getting important accordingly. We store a huge amount of data on our personal systems and use to internet to stay connected. This data can be used for public and private access, view and usage. Due to the continuous internet connection, this data source tends to exchange information very frequently over the network and hence exposed to several cyber risks [8]. People with malicious intentions misuse this data. From the last decade, cybersecurity has become one of the topmost issues for every internet user who frequently uses smartphones or computers to stay connected. Cyber-attacks may lead to various issues such as:

- Information extortion, identity theft, blackmailing
- Injecting viruses into multiple systems by inducting malware
- Spamming, phishing and spoofing
- Multiple attacks throughout different denial of services
- Intellectual property theft and sabotaging vital information
- Money scams by hacking accounts
- Ransomware
- Password crack and theft
- Vandalism using various websites

- Exploit privacy using web browsers

Cybersecurity is designed to prevent information theft, data breach and attacks like malware, ransomware [9]. It is considered as the only measured action against online fraud and risk management. Since hackers are getting smarter day by day, cybersecurity needs to deploy more intelligent and effective defense to fight against cyber-attacks.

Artificial intelligence advancements and growth in the machine learning applications number has led to develop new methodologies in the domain of cybersecurity space that is more risk free and automated [10], [11]. With the use of these applications, the cybersecurity personnel can easily organize, process, and manage network traffic log data. Cybersecurity produce a lot of historical data points which can make use of artificial intelligence for classification, clustering, filtering, and processing [11].

Machine learning though a very strong concept, cannot set-up and operate on its own. The servers produce raw data that needed to be processed on which decisions must be made. Machine learning analysis are mostly based on historical specific chunk-based data which finds out optimum solution for both present and future [12]. Therefore, the historical data will have to be made available to combine artificial intelligence, machine learning and cybersecurity logic implementation [13].

The algorithms are used to organize the historical cybersecurity data before providing specific instructions on various patterns to scan threats and others malicious contents [8]. The machine learning algorithms are implemented in such a way that the system can easily differentiate between a normal situation and an anomalous traffic which can compromise the security [1]. Machine learning algorithms are needed to build system that are quick to secure data almost instantly since most of the time hackers breach into a system and contaminate the data

before the organizations detect a breach has happened. With the help of artificial intelligence, attacks can be detected at a very early stage further actions would be taken to neutralize the threat [13].

Artificial intelligence makes strong security tools which increase the efficiency of intrusion detection. Highly effective cybersecurity tools are crucial when it comes to tracing negative entities inside the network [14]. Now a days, data tracking techniques are becoming more effective and hence reducing the risk level with increment of operational efficiencies. This is helping the cybersecurity space to fight the treats in multifold way [15]. Even machine learning is helping cybersecurity experts to analyze large-scale high-volume data sources in several ways such as;

- Finding correlation between different data sets by organizing in a particular pattern, making prediction, scanning for possible threats and forecasting future attacks [16].
- Different data wrangling and cleaning techniques, continuous auditing of data safeguard techniques can be implemented to protect the users and other relevant parties [6], [8].
- With the help of machine learning algorithms, cybersecurity professionals can optimize costs and avoid threats by applying rule-based mechanisms to secure data without being burden on the existing resources [17], [18].
- With the help of machine learning techniques, different malware and malicious contents can be easily detected since intelligent security platforms has the built-in mechanism of scanning high volume data on the network and simultaneously recognizing the threats [19], [20].

1.2. Limitations of machine learning in cybersecurity

The benefits listed above are the primary focus of machine learning techniques in cybersecurity, but there is also limitation that are preventing it from becoming mainstream tools used against cyber-attacks. In order to build a robust cybersecurity infrastructure machine learning models are needed to be free of false positives and missed detection [21], [22]. Here are some major limitations of machine learning techniques in cybersecurity:

- Evolving nature of cyber threats: Cybercriminals have virtually unlimited resources and creative mindset to gain economic benefits [23]. So, attackers introduce novel threats every time and machine learning models needed to be re-trained in order to keep up [24].
- Cybercriminals use machine learning: Since most of the cybersecurity infrastructures are adopting artificial intelligent models, bad actors are trying to keep up using data-driven threats to test their malicious contents against them. They use machine learning algorithms to detect the pattern and security loophole [6].
- Deployment of complex models: Deployment of machine learning models often has limitation due to their complexity to the requirements of legacy system. Changing the entire system configuration is hectic and very much expensive [25]. As a result, cybersecurity professionals always prefer an optimized solution which may not be the most effective one [26].

1.3. Impact of this research on intrusion detection

Machine learning implementations in cybersecurity are limited by their dependence on clean and feature engineered training data [27]. The “norm” is essential before we detect any anomalies. Machine Learning (ML) algorithms learn continuously with and without any human supervision using the feedback loop, but there is a fine line to measure the threshold between aid

and prohibition of its use for which it was intended [28]. Some ML algorithms faces some inherent issues which can lead to the reporting of false alarm [29], [30]. According to reddit, they have many threads from system, reported as malicious by using whitelisting programs build on ML models. They are spending tremendous time filtering manually from the falsified results to train their models again [31]. We can easily assume, that these issues will subside as the data becomes high dimensional. This research is primarily focused on subdue the predictive potentials of machine learning algorithms in cybersecurity by reducing the complexity and human monitoring. This research will help to build smarter and light weight cyber defense tool which could be used on different systems with low efficiency. System requirements is not a major concern rather than predictive potential in the domain of cybersecurity. But, by optimizing the complexity of a predictive model along with increasing the performance will change the cyber defense strategy remarkably. This research outcome is tested and implemented on well-established cybersecurity data and has proven effective in each experiment. This research will help people to adopt a cost-efficient cyber defense to safeguard their information.

1.4. Objectives

i. To research and study the applications of machine learning techniques in the cybersecurity domain and document improvement scopes for intelligent models for deployment on cybersecurity incident datasets.

ii. Further investigation on documented problem statements and research for improving machine learning algorithms and their application.

iii. Design and experiment with different data-driven solutions on multisource cybersecurity data using major improvements in machine learning techniques.

1.5. Dissertation outline

Chapter 2 of this dissertation presents the study of different machine learning applications and the most occurred issues aligned with them. At the last end of this study, we have documented four problem statements regarding machine learning model implementations. The following four chapters describe the methodology and improvements of different problem statements of Chapter 1. This research conducted on publicly available standard cybersecurity datasets to compare the outcomes with existing research. Chapter 3 presents the quality issues of the cybersecurity dataset associated with machine learning applications. Chapter 4 describes a novel hybrid algorithm to reduce the false-positive rates of the cybersecurity domain. Chapter 5 addresses the most critical issue of dynamic feature engineering for hybrid model performance.

1.6. References

1. Bicak, A., Liu, X.M. and Murphy, D., 2015. Cybersecurity curriculum development: Introducing specialties in a graduate program. *Information Systems Education Journal*, *13*(3), p.99.

2. Shin, B. and Lowry, P.B., 2020. A review and theoretical explanation of the 'Cyberthreat-Intelligence (CTI) capability'that needs to be fostered in information security practitioners and how this can be accomplished. *Computers & Security*, *92*, p.101761.

3. Masombuka, M., Grobler, M. and Watson, B., 2018, June. Towards an artificial intelligence framework to actively defend cyberspace. In *European Conference on Cyber Warfare and Security* (pp. 589-XIII). Academic Conferences International Limited.

4. Sarker, I.H., Kayes, A.S.M., Badsha, S., Alqahtani, H., Watters, P. and Ng, A., 2020. Cybersecurity data science: an overview from machine learning perspective. *Journal of Big Data*, *7*(1), pp.1-29.

5. Ahsan, M., Gomes, R. and Denton, A., 2018, May. Smote implementation on phishing data to enhance cybersecurity. In *2018 IEEE International Conference on Electro/Information Technology (EIT)* (pp. 0531-0536). IEEE.

6. Framework, P.C., Improving Critical Infrastructure Cybersecurity Executive Order 13636..

7. Vasilescu, C., 2015. CYBERSECURITY AND CYBERWAR. WHAT EVERYONE NEEDS TO KNOW Authors: PW SINGER, ALLAN FRIEDMAN. *Journal of Defense Resources Management (JoDRM)*, *6*(1), pp.137-138.

8. Donaldson, S.E., Siegel, S.G., Williams, C.K. and Aslam, A., 2015. Defining the cybersecurity challenge. In *Enterprise Cybersecurity* (pp. 3-25). Apress, Berkeley, CA.

9. Ten, C.W., Liu, C.C. and Manimaran, G., 2008. Vulnerability assessment of cybersecurity for SCADA systems. *IEEE Transactions on Power Systems*, *23*(4), pp.1836-1846.

10. Taddeo, M., McCutcheon, T. and Floridi, L., 2019. Trusting artificial intelligence in cybersecurity is a double-edged sword. *Nature Machine Intelligence*, *1*(12), pp.557-560.

11. Taddeo, M., 2019. Three ethical challenges of applications of artificial intelligence in cybersecurity. *Minds and Machines*, *29*(2), pp.187-191.

12. Fraley, J.B. and Cannady, J., 2017, March. The promise of machine learning in cybersecurity. In *SoutheastCon 2017* (pp. 1-6). IEEE.

13. Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A. and Marchetti, M., 2018, May. On the effectiveness of machine and deep learning for cyber security. In *2018 10th international conference on cyber Conflict (CyCon)* (pp. 371-390). IEEE.

14. Wu, M., Song, Z. and Moon, Y.B., 2019. Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods. *Journal of intelligent manufacturing*, *30*(3), pp.1111-1123.

15. Ullah, Z., Al-Turjman, F., Mostarda, L. and Gagliardi, R., 2020. Applications of artificial intelligence and machine learning in smart cities. *Computer Communications*, *154*, pp.313-323.

16. Boudt, K., Danielsson, J. and Laurent, S., 2013. Robust forecasting of dynamic conditional correlation GARCH models. *International Journal of Forecasting*, *29*(2), pp.244-257.

17. Leal, C., Meirinhos, G., Loureiro, M. and Marques, C.S., 2017, March. Cybersecurity management, intellectual capital and trust: a new management dilemma. In *ECIC 2017-9th European Conference on Intellectual Capital* (pp. 171-183).

18. Gordon, Lawrence A., and Martin P. Loeb. *Managing cybersecurity resources: a cost-benefit analysis*. Vol. 1. New York: McGraw-Hill, 2006.

19. Tsai, C.F., Hsu, Y.F., Lin, C.Y. and Lin, W.Y., 2009. Intrusion detection by machine learning: A review. *expert systems with applications*, *36*(10), pp.11994-12000.

20. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H. and Wang, C., 2018. Machine learning and deep learning methods for cybersecurity. *Ieee access*, *6*, pp.35365-35381.

21. Berman, D.S., Buczak, A.L., Chavis, J.S. and Corbett, C.L., 2019. A survey of deep learning methods for cyber security. *Information*, *10*(4), p.122.

22. Kwon, D., Kim, H., Kim, J., Suh, S.C., Kim, I. and Kim, K.J., 2019. A survey of deep learning-based network anomaly detection. *Cluster Computing*, *22*(1), pp.949-961.

23. Ganesan, R., Jajodia, S., Shah, A. and Cam, H., 2016. Dynamic scheduling of cybersecurity analysts for minimizing risk using reinforcement learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(1), pp.1-21.

24. Chung, K., Kamhoua, C.A., Kwiat, K.A., Kalbarczyk, Z.T. and Iyer, R.K., 2016, January. Game theory with learning for cyber security monitoring. In *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)* (pp. 1-8). IEEE.

25. Tisdale, S.M., 2015. Cybersecurity: Challenges from a Systems, Complexity, Knowledge Management and Business Intelligence Perspective. *Issues in Information Systems*, *16*(3).

26. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

27. Abraham, S. and Nair, S., 2014. Cyber security analytics: a stochastic model for security quantification using absorbing markov chains. *Journal of Communications*, *9*(12), pp.899-907.

28. Jia, Y., Qi, Y., Shang, H., Jiang, R. and Li, A., 2018. A practical approach to constructing a knowledge graph for cybersecurity. *Engineering*, *4*(1), pp.53-60.

29. Agarwal, C., Nguyen, A. and Schonfeld, D., 2019, September. Improving robustness to adversarial examples by encouraging discriminative features. In *2019 IEEE International Conference on Image Processing (ICIP)* (pp. 3801-3505). IEEE.

30. Sokolova, M. and Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. *Information processing & management*, *45*(4), pp.427-437.

31. Gaffney, D. and Matias, J.N., 2018. Caveat emptor, computational social science: Large-scale missing data in a widely-published Reddit corpus. *PloS one*, *13*(7), p.e0200162.

2. A COMPREHENSIVE STUDY OF MACHINE LEARNING TECHNIQUES IN CYBERSECURITY DOMAIN

2.1. Abstract

The importance of cybersecurity is rising in which machine learning is becoming increasingly significant. Several machine learning techniques and statistical methods are incorporating with artificial intelligence that are proven effective in this domain to prevent cyber-attacks. It is evident by multiple reasons that machine learning in cybersecurity is far more than merely applying well-established methods to datasets of cyber entities. Cybersecurity domain involves machine learning challenges that require efficient methodological and theoretical handling. Detecting security incident patterns or insights from cybersecurity data and building a corresponding data-driven model to prevent the attack, is the key to make a security system intelligent. Understanding and analyzing the actual phenomena with data, various scientific methods, processes, and systems are combined with machine learning techniques to build a robust and effective predictive model. In this research, we focus and briefly discuss on machine learning improvement for cybersecurity data, where the data has been gathered from relevant cybersecurity sources, and the analytics complement the latest data-driven patterns for providing more effective security solutions. The primary concern of applying machine learning in cybersecurity is to make the computing process more actionable and intelligent than traditional ones in the domain of cybersecurity. Primarily we are going to discuss and summarize several associated research issues and future directions. Furthermore, we summarize the major improvement scopes of machine learning technique and their applications for cybersecurity modeling. Overall, our goal is to discuss well-established machine learning algorithms in

cybersecurity and relevant models and focus the applicability on data-driven intelligent decision-making for protecting the systems from cyber-attacks.

2.2. Introduction

With the rapid increment of information technology in the past two decades, various types of security incidents such as unauthorized access [2], denial of service (DoS) [2], malware attack [3], zero-day attack [4], data breach [5], social engineering or phishing [6], etc. have increased at an exponential rate in last decade. From the record, in 2010, there were less than 50 million unique malware executables were documented by the security community. In the year 2012, this reported number is doubled to around 100 million. From the record in 2019, there are more than 900 million malicious executables discovered by the security community, and this number is continued to grow, according to the statistics of the AV-TEST institute in Germany [7]. Cybercrime and different network intrusions can cause devastating financial losses and affect organizations as well as individuals. It's estimated that an average data breach costs 3.9 million USD for the United States and 8.19 million USD worldwide [8], and the annual cost to the global economy from cybercrime is 400 billion USD [9]. According to the security community [10], the number of records breached each year to nearly triple over the next 5 years. Hence, it's essential that organizations need to adopt and implement a strong cybersecurity approach to prevent further loss. According to the latest articles by socio-economic researchers [11], the national security of a country relies on government, the business, and individual citizens having access to applications and tools which have the highest security, and the capability of detecting and eliminating such cyber-threats on time. Therefore, to intelligently identify various cyber incidents either previously seen or unseen, and effectively protect the relevant systems from such cyber-attacks, is the major concern needed to be addressed urgently.

Cybersecurity is an application or combined set of technologies and processes designed to protect programs, networks, computers, and data from damage, attack, or unauthorized access [12]. Cybersecurity is responsible of a variety of contexts, from business to mobile computing, and can be diversed into multiple common categories. These categories are - network security that mainly focuses on prevention of a computer network from cyber attackers or intruders; application security , which takes into account by keeping the devices and the software free of risks or cyber-threats; information security that primarily considers security and the privacy of relevant data; operational security that considers the processes of handling and protecting data assets. Conventional cybersecurity systems are composed of network security systems and computer security systems consisting a firewall, antivirus software, or an intrusion detection system. In current decade, cybersecurity is undergoing massive changes in technology and its operations in the context of computing, and data science is driving the shift, where machine learning, a core part of "Artificial Intelligence" can play a vital role to discover the hidden pattern from data. Machine learning have significantly changed the cybersecurity breakthrough landscape and data science is leading a new scientific paradigm [13, 14]. The efficiency of these related technologies is increasing daily, which is shown in Fig. 2.1, based the last five years collected data from Google Trends [15]. The figure is the representation of timestamp information in terms of a particular date represented in the x-axis and y-axis is representing the corresponding popularity in the range of 0 (minimum) to 100 (maximum). From the Fig. 2.1, the popularity indicates values of these areas are less than 30 in 2014, and they exceed 70 in 2018, i.e., more than double in terms of increased popularity. In this study, we focus on machine learning in Cybersecurity which is vastly related to these areas in terms of security intelligent decision making and data processing techniques to deploy in real-world

applications. Overall, this study is security data-focused, applies machine learning methods to quantify cyber risks, and ultimately seeks to optimize cybersecurity operations. Also, the purpose of this study is for those academia and industry researchers who want to study and develop data-driven smart cybersecurity models based on machine learning techniques. Therefore, major emphasis is placed on this study as a thorough description of different types of machine learning techniques, and their relations and usage in the context of cybersecurity. This study does not describe all of the different techniques used in cybersecurity in detail; but it provides an overview of machine learning modeling on cybersecurity based on artificial intelligence, particularly from smart and robust cybersecurity perspective.

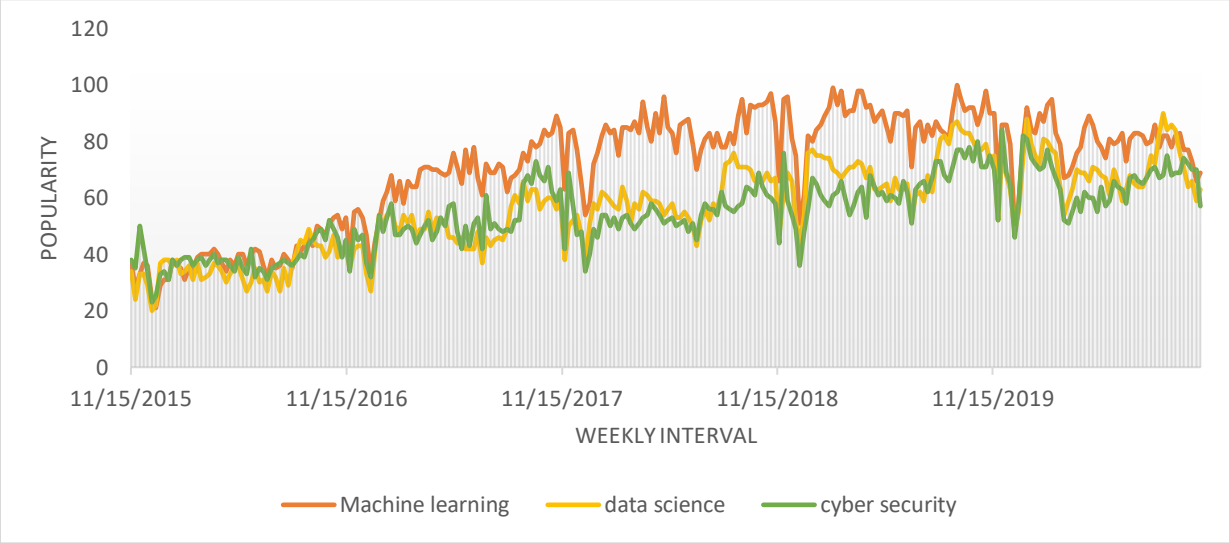

Fig 2.1: Google Trend for Machine learning vs Data Science vs Cybersecurity for last five years.

Analyzing cybersecurity data and developing the right tools to processes them successfully prevent cybersecurity attacks that goes beyond a set of simple functional requirements and knowledge about risks, threats or vulnerabilities. To effectively extract the insights or the patterns of security incidents, several machine learning techniques, such as feature engineering, clustering, classification, and finding association, or neural network focused deep learning techniques can be applied, which are briefly discussed in “Machine learning techniques

in cybersecurity" section. These learning techniques are capable of finding the anomalies or malicious behavior and data-driven patterns of associated security incidents to prevent cyberattacks by taking intelligent decisions.

The ultimate goal of machine learning in cybersecurity is data-driven intelligent decision making from security data for smart cybersecurity solutions. Machine Learning is a partial but most important change from traditional well-known security solutions such as user authentication and access control, firewalls, cryptography systems etc. that might not be effective according to today's need in cyber industry [16–19]. The major issue is that these are mostly fixed manually by a domain experts and security analysts, where ad-hoc data management is performed [20, 21]. However, as a number of cybersecurity incidents in different formats continuously appear over time, such conventional solutions have encountered limitations in mitigating these types of cyber risks. Therefore, numerous new advanced attacks are created and spread very quickly throughout the network. Hence, several researchers use various data analysis and learning techniques to build cybersecurity models which are summarized in "Machine learning techniques in cybersecurity" section, based on the effective discovery of security insights and latest security patterns that could be more useful. To address this cyber problem, we need to develop more flexible and efficient security systems that can respond to attacks and to update security policies to eliminate them intelligently on time. To reach this goal, it is inherently required to analyze a large amount of relevant cybersecurity data generated from multiple sources such as network and system sources, and to identify different insights or better security policies with minimal human intervention in an automated manner.

The discussions of this study are listed as below.

- We first document a brief discussion on the existing concept of cybersecurity defense strategies and relevant methods to understand its applications of data-driven decision making in the domain of cybersecurity which are handled intelligently. To satisfy this purpose, we have reviewed and discussed briefly on different machine learning algorithms used in cybersecurity, and document various cybersecurity datasets highlighting their importance and applicability in different data-driven cyber defense.
- Later on we discuss and summarize a number of related research issues and future directions in the area of machine learning techniques in cybersecurity, that could help both the academia and industry researchers to further research and development in relevant domain.
- Finally, we documented and filtered the most common issues of applying machine learning algorithms on cybersecurity datasets and study the scope of improvements to build a robust system.

The remainder of this study is organized as follows. “Background” section summarizes motivation of our study and gives an overview of the related technologies of cybersecurity. “Cyberattacks and security risks” section defines and discusses briefly about cybersecurity data types including various categories of cyber incidents data. In “Machine learning techniques in cybersecurity” section, we briefly discuss different categories of machine learning techniques including their relations with various cybersecurity tasks and summarize a number of most effective machine learning algorithms for cybersecurity models in this domain. “Cybersecurity research issues and improvements scopes” section briefly discusses and highlights various research issues and future directions in the area of cybersecurity. In “Discussion” section, we

highlight multiple key points regarding our findings. Finally, “Conclusion” section concludes this paper.

2.3. Background

Over the last couple of decades, the Information and Communication Technology (ICT) infrastructure has evolved greatly, which is ubiquitous and immensely integrated with our modern society. Therefore, protecting ICT systems and applications from cyber-attacks has been urged by the security policymakers now a days [22]. The act of protecting ICT structure from various cyber-threats or attacks has been named as cybersecurity [9]. Different aspects are associated with cybersecurity, such as measures to protect ICT, the raw data and information it contains and their processing and transmission; association of virtual and physical elements of the systems; the level of protection resulting from the application of those measures; and eventually the associated field of professional endeavor [23]. According to researcher Craigen , cybersecurity consists of different tools, guidelines, and practices which is employed to protect software programs, computer networks, and data from attack, unauthorized access or damage [24]. Researcher Aftergood et al. [12], defined that, cybersecurity uses different processes and technologies which are useful to protect networks, programs computers, and data from attacks, alteration and unauthorized access, or destruction. In a nutshell, cybersecurity concerns with the understanding of diverse cyber-attacks and deploying corresponding defense strategies that protect several properties listed as below [24, 25, 26].

- Confidentiality is a property that is used to prevent the information disclosure to unauthorized entities, individuals, or systems.
- Integrity is a property that is used to prevent any unauthorized destruction or modification of information.

- Availability is a property that is used to ensure timely and reliable access of information assets and systems to an authorized entity.

2.4. Cyberattacks and security risks

There are three major security factors which are typically considered as risks. Those factors are attacks, i.e., who is attacking, vulnerabilities in the system, i.e., the weaknesses or security pocket they are attacking, and impacts, i.e., what the attack does [9]. A security breach is an act that threatens the confidentiality, integrity, or availability of information assets and systems. Different types of cybersecurity incidents that may result in security risks on an organization's systems and networks or an individual [2]. These cyber-attack types are briefly described in Table 2.1:

Table 2.1: Different Cyber-attack types and their brief description.

Attack Types	Description
Malware	It is a malicious software that is designed to cause damage to a personal system, client, server, or computer network. Malware includes spyware, ransomware, viruses, and worms. Malware breaches a network by creating a vulnerable situation like, user clicking a dangerous link or email attachment and hence installs a risky software. Typically, malware affects the network as: • Network key components are blocked (Ransomware) • Installs additional harmful software for spying with malware itself. • Gain Access to personal data and transmit information. • Disrupts certain components and make the system inoperable to users. Ransomware blocks access to victim's data and threatens the client to destroy it unless ransom is pain. Trojan horse is the most dangerous malware which appears to be useful and routine software and mostly designed to steal financial information. Drive-day attack is a common method for distributing malware. They don't require any actions of users to be activated. The users just need to visit a benign like website and their personal system are infected silently and become an IFRAME that redirect's victim's browser into a site controlled by the attacker.
Phishing	Phishing is a practice of sending fraudulent communications or social engineering which mostly spread through emails. The goal is to steal victim's data such as credit card numbers and login credentials. This attack is often used to obtain a foothold in government or corporate networks as a part of significant plot as an advanced persistent threat (apt). Spear phishing is targeted to particular individual or organizations, government, military intelligence to acquire trade secrets, financial gain or information. Whale phishing is mostly aimed for high profile employees such as CFO or CEO to gain vital information on company's sensitive data.
Man-in-middle-attack	Man-in-the-middle (MITM) also known as eavesdropping occurs when the intruders successfully include themselves inside of two-party transaction or communication. Most common entry for MITM attackers are: • Unsecure public WiFi where intruders insert themselves between visitor's device and the network. • If attacker's malware successfully breaches into victim's system, they can install much software to gain victims secure information.
Denial-of-service-attack	Denial-of-service (DDoS) shuts down a network or service with a huge traffic to exhaust resources and bandwidth resulting the system cannot fulfill legitimate requests. DDoS are often designed to target web servers of high-profile organizations such as trading platform, media, banking and government.
SQL Injection	SQL Injection (SQLI) is aimed to employ malicious code to manipulate backend database access information that was not intended for display. Intruders could carry out a SQL injection simply by submitting malicious code into vulnerable website search box.
Zero-day Exploit attack	Zero-day exploit attack is considered as the term that used to describe the threat of an unknown security vulnerability for which the patch has not been released yet or the application developers are unaware about. To detect this threat the developers requires constant awareness.
DNS-Tunneling	DNS Tunneling uses the DNS protocol to communicate non-DNS traffic over port 53 by sending HTTP and other protocol traffic over DNS. Since using DNS Tunneling is a common and legitimate process, hence using it for malicious reasons are very often. Attackers can use to disguise outbound traffic as DNS, concealing data that is shared through an internet connection.

2.5. Defense strategies

Defense strategies are needed to preserve data or information, information systems, and Networks to prevent from cyber-attacks or intrusions. More precisely, they are responsible for the prevention of data breaches or security incidents monitoring and reacting to threat, which can be defined as any kind of unauthorized activity that causes damage to a network and personal systems [37]. An intrusion detection system (IDS) is described as "a software, device or application that monitors a systems or computer network for malicious activity or policy violations" [39]. The well-established security solutions such as user authentication, access control, anti-virus, firewalls, cryptography systems, and data encryption however might not be effective according to today's need in the cyber industry [16–19]. Besides that, IDS resolves the issues by analyzing security data from several key points in a network or system [38, 40]. Moreover, IDS can be used to detect both internal and external attacks. Intrusion detection systems are of various categories according to the usage scope. As an example, a host-based intrusion detection system (HIDS), and network intrusion detection system (NIDS) are the well-known types based on the scope of single computers to large networks. In a HIDS, the system monitors data, files, secured information on an individual system, while it monitors and analyzes network connections for suspicious traffic in a NIDS. Similarly, based on theories, the signature-based IDS, and anomaly-based IDS are the most well-established variants [37]. There is a brief overview of defense strategies against cyber-attacks are described in Fig 2.2.

2.5.1. Signature-based IDS

A signature can be a defined string, pattern, or rule that corresponds to a previously occurred attack. A known pattern is defined as the detection of corresponding similar threats according to signature-based Intrusion Detection System. An example of a Signature-based IDS

can be sequences used by mostly different types of malware or known patterns or a byte sequence in a network traffic. Anti-virus software is used to detect these attacks, by identifying such types of patterns or sequences as a signature while performing the similar operation. For this reason, Signature-based IDS is also known as knowledge-based or misuse detection [41]. This technique can be an efficient to process a high volume of network traffic, however, is strictly limited to the rule based or supervised detection only. Thus, detecting new attacks or unseen attacks using historical knowledge is one of the biggest challenges faced by this signature-based system.

2.5.2. Anomaly-based IDS

The concept of anomaly-based detection is introduced to overcome the issues of signature-based IDS discussed above. In an anomaly-based intrusion detection system, the user behavior and the traffic of the network is first examined to identify dynamic patterns, to automatically develop a data-driven model, to profile the normal behavior, and thus it detects anomalies in the case of any deviation [41]. Thus, anomaly-based IDS can be described as a dynamic approach, which follows both supervised and unsupervised detection. The major advantage of anomaly-based IDS is the ability to detect zero-day attacks and completely unknown threats [42]. However, the identified anomaly or suspicious behavior sometimes led to false alarm as an indicator of intrusions which arises an issue. Sometimes it may detect several factors such as policy changes or offering a new service as an intrusion.

Besides that, a hybrid detection approach [43, 44] which considers the anomaly-based and the signature-based techniques discussed above can be used to identify intrusions. In a hybrid system, the signature-based detection system is used to identify known types of intrusions and anomaly detection system is used for unknown attacks [45]. In addition to these approaches,

stateful protocol analysis can be useful to detect intrusions that identifies deviations of protocol state which is similar to the anomaly-based method, however it uses predetermined standard profiles according to accepted definitions of benign activity [41]. Among these approaches, a self-aware automatic response system would be more effective as it does not need a human interface between the detection and response systems.

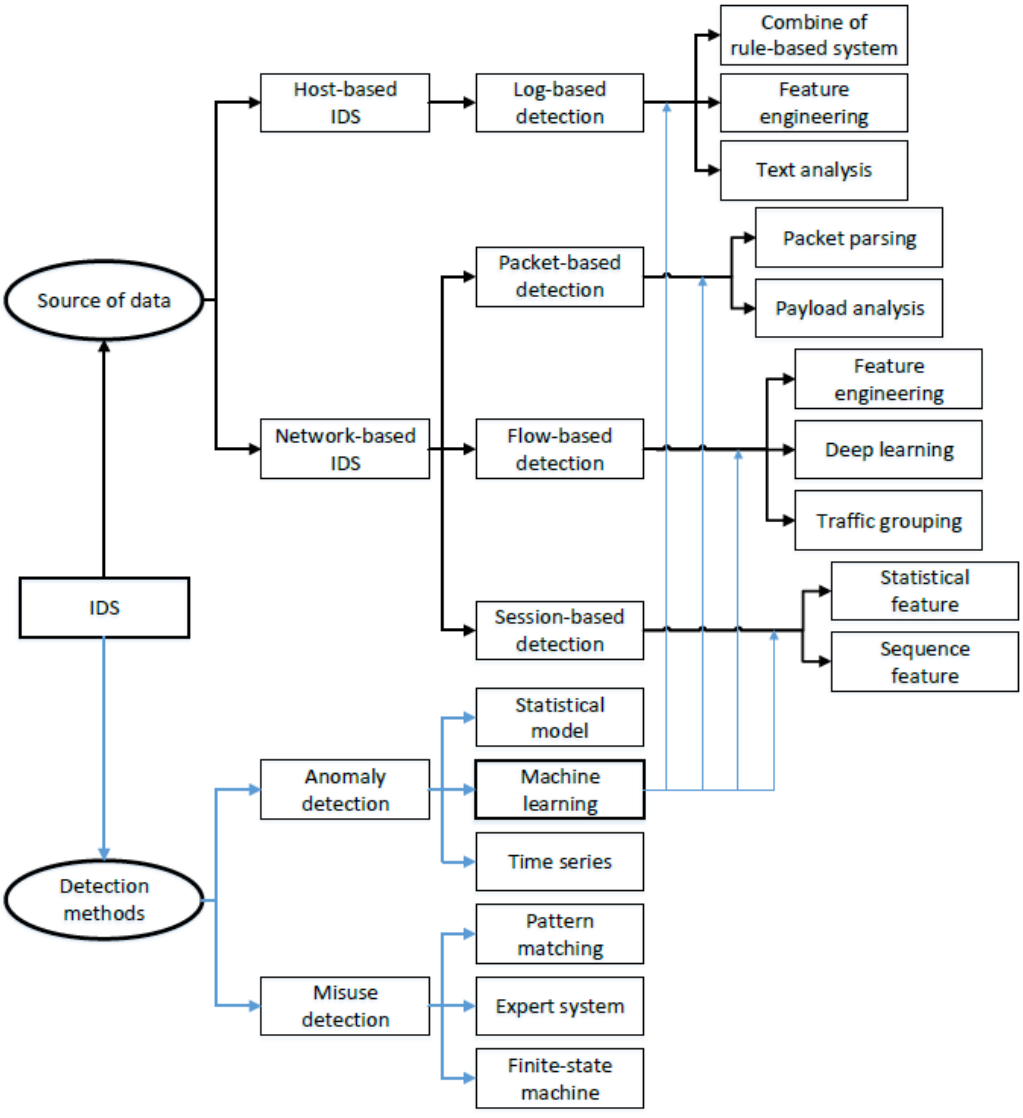

Fig 2.2: Flow chart of defense strategies in cybersecurity.

2.6. Cybersecurity data

Machine learning in cybersecurity is largely driven by the availability of cybersecurity data [48]. Datasets typically represent a collection of records that consist of information as several attributes or features and related facts, in which machine learning techniques in cybersecurity is based on. Therefore, it's important to understand the nature of cybersecurity data containing various types of cyber incidents and relevant features. The reason behind this is, raw security data collected from similar cyber sources can be used to analyze the different patterns of security incidents or malicious behavior, to build a data-driven security model to achieve our goal. Different datasets exist in the area of cybersecurity including Network intrusion analysis, malware analysis, Phishing detection, fraud, anomaly, or spam analysis that are used for various purposes. In Table 2.2, we summarize different types of datasets including their various features and incidents which are accessible on the internet, and emphasize their usage based on machine learning techniques in different cyber applications. Effectively analyzing and processing of these network and standard features, building target machine learning-based security model according to the defensive requirements, and eventually, data-driven decision making, could play a role to provide intelligent cybersecurity services.

Table 2.2: Different types of available cyber-attack datasets and their description.

Dataset	Description
IMPACT	Mostly known as the Protected Repository for the Defense of Infrastructures Against Cyber Threats (PREDICT), which is a community that produces security-relevant network operation data and research in the domain of information security and computer networks. This repository provides developers and researchers with regularly updated network operations data that is relevant to cyber defense technology development. The Dataset Catalog is publicly accessible, and anyone can browse dataset details without using any credentials. Current users can log in to website to request datasets.
SNAP	Not specific to security, but there are several relevant graph data sets.
KYOTO	Traffic Data from Kyoto University's Honeypots.
KDD'99 Cup	Most widely used network data set containing 41 features for evaluating anomaly detection methods, where threats are categorized into four major target labels, such as remote-to-local (R2L), denial of service (DoS), probing, and user-to-remote (U2R) [50]. KDD'99 Cup dataset is widely used to evaluate ML-based threat detection model.
NSL-KDD	This is a refined version of KDD'99 cup dataset in which redundant records are eliminated. Hence ML classification-based security model utilizing NSL-KDD dataset will not be biased towards more frequent records [51]
DARPA	It is a very authenticated Intrusion Detection System (IDS) dataset which includes LLDOS 1.0 and LLDOS 2.0.2 threat scenario data. Data traffic and threats containing in DARPA dataset are collected by MIT Lincoln Laboratory for evaluating Network Intrusion Detection Systems (NIDS)[44, 49]
UNSW-NB15	This dataset has 49 features and nine different types of threat types including DoS which was gathered from the University of New South Wales (UNSW) cybersecurity Lab in 2015 [59]. UNSW-NB15 can be used for evaluation of ML-based anomaly detection systems in cyber applications.
ADFA IDS	This is an intrusion dataset with different versions named ADFA-LD and ADFA-WD that is issued by the Australian Defense Academy (ADFA) [63]. This dataset is designed to evaluate host-based IDS.
MAWI	A collection of cybersecurity dataset governed by Japanese network research institutions and academic institutions that is widely used to detect and evaluate DDoS intrusions using machine learning techniques [62]
CERT	This dataset includes users' activity logs that was created for the purpose of validation of insider-threat detection systems [64, 65]. This can be used to monitor and analyze machine learning based user behavioral activities
Bot-IoT	It is a dataset that incorporates legitimate and simulated Internet of Things (IoT) network traffic, along with different attacks for network forensic analytics in the area of Internet of Things [80]. Bot-IoT is mostly used to evaluate the reliability using different statistical and machine learning techniques for forensics purposes
DGA	The Alexa Top Sites dataset is primarily used as a reliable source of benign domain names [69]. The malicious domain names are collected from OSINT [70] and DGArchive [71]. DGA dataset is mostly used for experiments in ML-based automatic DGA botnet detection or domains classification [72]
CTU-13	This is a labeled malware dataset including background traffic, botnet, and normal user activities which was captured at CTU University, Czech Republic [58]. CTU-13 is primarily used for data-driven malware analysis using machine learning techniques and to evaluate the standard malware detection system.
CAIDA	The CAIDA'07 and CAIDA'08 datasets contain DDoS attack traffic and normal standard traffic history [52, 53]. So, CAIDA DDoS dataset is mostly used to evaluate machine learning based DDoS attack detection model and identifying Internet Denial-of-Service activity.
CIC-DDoS2019	This is a dataset containing historical DDoS attacks that was collected by the Canadian Institute for Cybersecurity [61]. CIC-DDoS is effectively used for network traffic behavioral analytics to identify DDoS attacks using ML techniques
ISCX'12	This dataset contains 19 features and 19.11% of the network traffic belongs to DDoS attacks. ISCX'12 was documented at the Canadian Institute for Cybersecurity [56, 57] and wellknown for the usage of evaluation of the effectiveness of machine learning based network intrusion detection modeling.
Malware	It is a collection of multiple malware based datasets such as Genome project [73], VirusTotal [75], Virus Share [74], Comodo [76], Contagio [77], Microsoft [79], and DREBIN [78] which contain malicious files. These datasets is widely used for data-driven malware analysis using machine learning techniques and to evaluate existing malware detection system.
EMAIL	Email based datasets are difficult to collect because of privacy concerns. This dataset is a collection of some common corpora of emails including EnronSpam [66], LingSpam [68], and SpamAssassin [67].
DREBIN	To foster and improve the research on Android malware and to document a comparison of different detection approaches, researchers have make the datasets from project Drebin publicy available. This dataset contains 5,560 applications from 179 different malware categories. The samples have been collected in between the period of August 2010 to October 2012 and were made publicly available to cybersecurity practitioners by the MobileSandbox project.
CDX_2009_Network_USMA	The National Security Agency (NSA) permitted both the recording and release of the CDX_2009_Network_USMA datasets. In order to provide users of this dataset highlights to correlate IP addresses found in the PCAP files with the IP addresses to hosts on the internal USMA network. NSA has included a pdf file of the planning document that is used just prior to the execution of CDX 2009 (NOTE: USMA utilized network address translation). This was a planning document and have several data inconsistencies. Many changes may have occurred to the USMA network which might not be annotated on this document.

2.7. Machine learning techniques in cybersecurity

Machine learning (ML) is typically described as a branch of "Artificial Intelligence", that is closely related to data mining, computational statistics and analytics, data science, particularly focusing on making the systems to learn from historical data [82, 83]. Therefore, machine learning models typically comprise of a set of rules, methods, or complex functions and equations which can be used to find interesting data patterns, or to recognize sequence or to predict behavior [84], that could play an important role in the area of cybersecurity. In this section, we discuss different methods that can be used to solve machine learning techniques and how they are related to cybersecurity. Figure 2.3 depicts a summarized view of the most frequently used machine learning techniques for cybersecurity.

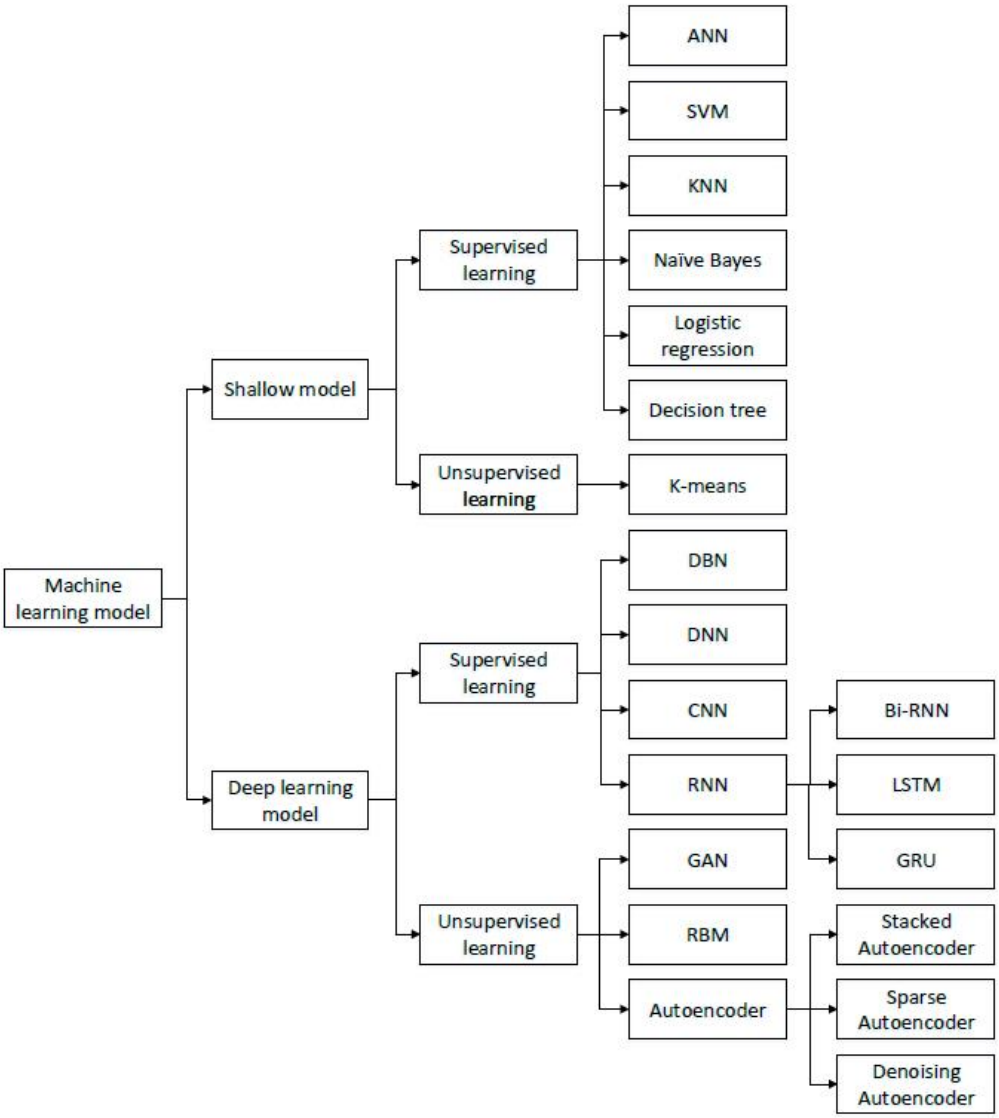

Fig 2.3: Taxonomy of machine learning techniques used in cybersecurity.

2.7.1. Supervised learning

Supervised learning relies on useful information on historical labeled data. Supervised learning is performed when targets are predefined to reach from a certain set of inputs, i.e., task-driven approach. classification and regression methods are the most popular supervised learning techniques [129]. These techniques are well established to classify or predict the target variable for a particular security threat. For instance, for the prediction of denial-of-service attack (yes,

no) or for the identification different labels of network threats such as scanning and spoofing, classification techniques can be used in the cybersecurity. Navies Bayes [131], support vector machines [135], Decision Tree [132, 133], K-nearest neighbors [134], adaptive boosting [136], and logistic regression [137] are some of the well-known classification techniques in shallow model. Besides that, to predict the continuous target variable or numeric value, e.g., predicting total phishing attacks in a certain period of time or predicting the network packet features, regression techniques are useful. Regression analysis can also be used to detect the root causes of cybercrime and other types of risk analysis [138]. Linear regression [82], support vector regression [135], random forest regressor are some of the popular regression techniques. The major difference between classification and regression is that the output variable in the regression is numerical or continuous, but the predicted output for classification task is categorical or discrete. An ensemble learning is an extension of supervised learning which mix different shallow models, e.g., XGBoost, Random Forest learning [139] that generates multiple decision trees to solve a particular security task.

2.7.2. Unsupervised learning

In unsupervised learning problems, the major task is to find patterns, structures, or useful information in unlabeled data, i.e., data-driven approach [140]. In the area of cybersecurity, risks such as malware stays hidden in some ways, include changing their behavior dynamically to avoid being detected. Clustering techniques, another type of unsupervised learning, can help to uncover the hidden patterns and insights from the datasets, to detect indicators of such sophisticated attacks. For instance, in identifying anomalies, policy violations, detecting, and eliminating noisy instances in data, clustering techniques can be useful. K-means [141], K-medoids [142] are well-established partitioning clustering algorithms, and single linkage [143] or

complete linkage [144] are the popular hierarchical clustering algorithms used in various application domains. Moreover, Principal component analysis (PCA), linear discriminant Analysis (LDA), Pearson correlation, or non-negative matrix factorization are the weel established dimensionality reduction techniques to solve such issues [82]. Association rule mining is another example, where machine learning based policy rules can learn to prevent cyber incidents. In an expert system, the rules and logics are usually manually documented and deployed by a knowledge engineer collaborating with a domain expert [37, 140, 146]. Association rule learning in contrast discovers the rules or relationships among a set of available security features or variables in a given dataset [147]. To quantify the strength of relationships, different statistical analysis is used [138]. Various association rule mining algorithms have been proposed in the area of machine learning and data mining literature, such as tree-based [152], logic-based [148], frequent pattern based [149–151], etc. Moreover, Apriori [149], Apriori-TID and Apriori-Hybrid [149], AIS [147], FP-Tree [152], and RARM [154], and Eclat [155] are the well-established association rule learning algorithms that are capable of solving such issues by producing a set of policy rules of cybersecurity.

2.7.3. Shallow models

The traditional machine learning models are often known as Shallow Models for Intrusion Detection System (IDS) primarily include the support vector machine (SVM), K-nearest neighbor (KNN), naïve Bayes, logistic regression (LR), decision tree, artificial neural network (ANN), clustering, and combined and hybrid methods. Some of these methods have been studied for several decades, and their methodology is well-established. They focus not only on the effective intrusion detection but also on labeling, e.g., detection efficiency and data management.

2.7.4. Deep learning models

Deep learning is a segment of machine learning in the area of artificial intelligence, which is a computationally complex model that is inspired by the biological neural networks in the human brain [82]. Deep learning models consist of multiples of deep networks. The main difference between deep learning and shallow machine learning is its performance on the amount of security data increases. The number of studies of deep learning-based Intrusion Detection System has increased rapidly from 2015 to the present. Deep learning models directly learn feature representations from the original data, such as images and texts, without requiring extensive manual feature engineering. Thus, deep learning methods can execute with less data processing and more effective manner. For large datasets, deep learning methods have a significant advantage over classical machine learning models. Some of the widely used deep learning techniques in cybersecurity includes, deep brief networks (DBNs), deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) as supervised learning models, while autoencoders, restricted Boltzmann machines (RBMs), and generative adversarial networks (GANs) as unsupervised learning models. Table 2.3 contains brief description of machine learning techniques used in cybersecurity now a days.

Table 2.3: A summary of machine learning techniques in the domain of cybersecurity.

Machine Learning Techniques	Purpose	References
SVM	• For classification of various attacks such as DoS, Probe, U2R, and R2L • Feature selection, intrusion detection and classification • Evaluating host-based anomaly detection systems	Kotpalliwar et al. [85] Pervez et al. [86], Yan et al. [87], Li et al. [88], Raman et al. [89] Xie et al. [91]
KNN	• Network intrusion detection system • To reduce the false alarm rate	Shapoorifard et al. [94], Vishwakarma et al. [95] Meng et al. [96]
SVM and KNN	To build intrusion detection system	Dada et al. [97]
K-means and KNN	To build intrusion detection system	Sharifi et al. [98,177]
Naïve Bayes	To develop an intrusion detection system for multi-class classification.	Koc et al. [100,173]
Decision Tree	• To identify the malicious code's behavior information by running malicious code on the virtual machine and analyze the behavior information for intrusion detection. • Significant feature selection and to develop an effective network intrusion detection system.	Moon et al. [101,174] Ingre et al. [102], Malik et al. [103,175], Relan et al. [104], Rai et al. [105], Sarker et al. [106], Puthran et al.[107,176]
Decision Tree and KNN	Anomaly intrusion detection system.	Balogun et al. [108]
Genetic Algorithm and Decision Tree	To address the issue of small disjunct in the decision tree-based intrusion detection system.	Azad et al. [109,180]
Decision Tree and ANN	To measure the performance and vulnerability of intrusion detection system and beta test of ethical hacking.	Jo et al. [110]
Random Forests	To build network intrusion detection systems.	Zhang et al. [111,181]
Association Rule	To build network intrusion detection systems.	Tajbakhsh et al. [112,178]
Semi-supervised Adaboost	For network anomaly detection.	Yuan et al. [115,179]
Genetic Algorithm	To prevent cyberterrorism through dynamic and evolving intrusion detection System (IDS).	Hansen et al. [118], Aslahi et al. [119]
Deep Learning, RNN, LSTM	To develop an anomaly intrusion detection system and attack classifier.	Alrawashdeh et al. [120], Yin et al.[121], Kim et al. [122], Almiani et al.[123,173]
Deep Learning Convolutional	Malware traffic classification system.	Kolosnjaji et al. [124], Wang et al.[125]
Deep Reinforcement Learning	Malicious activities and intrusion detection system.	Alauthman et al. [126], Blanco et al.[127, 171], Lopez et al. [128]

2.8. Cybersecurity research issues and improvements scopes

In our study we have documented several research issues and challenges in the area of machine learning in cybersecurity to extract insight from relevant data towards data-driven intelligent decision making for cybersecurity solutions. In the following, we listed the most common challenges ranging from data collection to decision making.

2.8.1. Cybersecurity datasets availability

Source datasets are the primary component to work with machine learning in cybersecurity. Most of the existing and publicly available datasets are old and might not be sufficient in terms of understanding the undocumented behavioral patterns of different cyber-attacks. Although the existing data can be transformed into a knowledge level after performing several primary processing tasks, there are still a lack of understanding of the nature of recent attacks and their patterns of occurrence. Therefore, further processing or machine learning techniques may provide a low accuracy rate for making the final decisions. Establishing a large number of recent cybersecurity datasets for a particular problem domain like attack prediction or intrusion detection is crucial, which could be one of the primary challenges to perform machine learning techniques in cybersecurity.

2.8.2. Quality problems in cybersecurity datasets

The cybersecurity datasets might be imbalanced, noisy, incomplete, insignificant, or may contain inconsistent instances related to a particular security breach. Such problems in a data set may degrade the quality of the learning process and affect the performance of the machine learning-based models [161,170]. To build a data-driven intelligent decision for cybersecurity solutions, such problems in data is needed to deal effectively before building the cyber models using machine learning techniques. Therefore, understanding such problems in cybersecurity

data and effectively handling such issues using existing algorithms or newly proposed algorithm for a specific problem domain like malware analysis or intrusion detection and prevention is needed, which would be another research issue for machine learning in cybersecurity domain.

2.8.3. Hybrid learning

Most popular and well-established techniques in the cybersecurity domain contain signature-based intrusion detection methods [41,181]. However, missing features or significant feature reduction or insufficient profiling can cause these techniques to skip unknown attacks or incidents. To address this issue, anomaly-based detection techniques or hybrid technique, which is a combination of both anomaly-based and signature-based can be used to overcome such drawbacks. A hybrid learning technique combining multiple machine learning techniques or a combination of deep learning, statistical analysis and machine-learning methods can be used to extract the target insight for a particular problem domain like intrusion detection, malware analysis, phishing detection, etc. and make the intelligent decision for corresponding cybersecurity solutions.

2.8.4. Feature engineering in cybersecurity

The efficiency and performance of a machine learning-based security model has always been challenged due to the high volume of network traffic data with a large number of insignificant traffic features. The large dimensionality of data has been handled using several techniques such as principal component analysis (PCA) [167,169], singular value decomposition (SVD) [168,172], Linear Discriminant Analysis (LDA) etc. Often in addition to low-level features in the datasets, the contextual relationships between suspicious activities might be relevant. Such contextual data can be processed through an ontology or taxonomy for further analysis. Hence, how to effectively select the optimal features or extracting the significant

features considering the machine-readable features as well as the contextual features, for efficient cybersecurity solutions would be another research issue for machine learning techniques in cybersecurity.

2.9. Conclusion

Driven by the growing significance of cybersecurity and machine learning technologies, in this study, we have discussed how machine learning techniques are applied to data-driven intelligent decision making in cybersecurity systems and services. In this study, we also have discussed how it can impact security data, both in terms of extracting insight of security incidents and analyzing dataset. We aimed to work on machine learning improvements and research issues in cybersecurity domain discussing the state-of-the-art documented security incidents dataset and corresponding security services. Our discussion also included how machine learning techniques can impact in the domain of cybersecurity and examine the security challenges that remain to further research areas. In terms of existing research, a lot of focus has been provided on traditional security solutions, with less available work in machine learning algorithm-based security systems. The paper follows an IDS taxonomy that takes data sources as the main thread to present the numerous machine learning algorithms used in this field. According to this taxonomy, we then analyze and discuss IDSs applied to various data sources, i.e., logs, packets, flow, and sessions. IDSs are targeted to detect attacks, therefore it is important to select proper data source according to attack characteristics. Logs contain detailed semantic information, which are suitable for detecting SQL injection, U2R, and R2L attacks and hence they can be used for further analysis using machine learning techniques. And packets provide communication contents, which are useful to detect U2L and R2L attacks. We have further detected and discussed various key issues in security analysis to showcase the interest of future

research ideas in the domain of machine learning with cybersecurity. We are primarily focused on extracting insights from security data, to set a research design with specific attention to concepts for data-driven intelligent security solutions using machine learning.

2.10. References

1. Li, S., Da Xu, L. and Zhao, S., 2015. The internet of things: a survey. *Information Systems Frontiers*, *17*(2), pp.243-259.

2. Sun, N., Zhang, J., Rimba, P., Gao, S., Zhang, L.Y. and Xiang, Y., 2018. Data-driven cybersecurity incident prediction: A survey. *IEEE communications surveys & tutorials*, *21*(2), pp.1744-1772.

3. McIntosh, T., Jang-Jaccard, J., Watters, P. and Susnjak, T., 2019, December. The inadequacy of entropy-based ransomware detection. In *International Conference on Neural Information Processing* (pp. 181-189). Springer, Cham.

4. Alazab, M., Venkatraman, S., Watters, P. and Alazab, M., 2010. Zero-day malware detection based on supervised learning algorithms of API call signatures.

5. Shaw, A., 2009. Data breach: from notification to prevention using PCI DSS. *Colum. JL & Soc. Probs.*, *43*, p.517.

6. Gupta, B.B., Tewari, A., Jain, A.K. and Agrawal, D.P., 2017. Fighting against phishing attacks: state of the art and future challenges. *Neural Computing and Applications*, *28*(12), pp.3629-3654.

7. Geer, D., Jardine, E. and Leverett, E., 2020. On market concentration and cybersecurity risk. *Journal of Cyber Policy*, *5*(1), pp.9-29.

8. Buecker, A., Borrett, M., Lorenz, C. and Powers, C., 2010. Introducing the ibm security framework and ibm security blueprint to realize business-driven security. *International Technical Support Organization*.

9. Fischer, Eric A. "Cybersecurity issues and challenges: In brief." (2014).

10. Chernenko, E., Demidov, O. and Lukyanov, F., 2018. Increasing international cooperation in cybersecurity and adapting cyber norms. *Council on Foreign Relations*.

11. Papastergiou, S., Mouratidis, H. and Kalogeraki, E.M., 2019, May. Cyber security incident handling, warning and response system for the european critical information infrastructures (cybersane). In *International Conference on Engineering Applications of Neural Networks* (pp. 476-487). Springer, Cham.

12. O'Connell, M.E., 2012. Cyber security without cyber war. *Journal of Conflict and Security Law*, *17*(2), pp.187-209.

13. Tolle, K.M., Tansley, D.S.W. and Hey, A.J., 2011. The fourth paradigm: data-intensive scientific discovery [point of view]. *Proceedings of the IEEE*, *99*(8), pp.1334-1337.

14. Cukier, K., 2010. *Data, data everywhere: A special report on managing information*. Economist Newspaper.

15. Choi, H. and Varian, H., 2012. Predicting the present with Google Trends. *Economic record*, *88*, pp.2-9.

16. Anwar, S., Mohamad Zain, J., Zolkipli, M.F., Inayat, Z., Khan, S., Anthony, B. and Chang, V., 2017. From intrusion detection to an intrusion response system: fundamentals, requirements, and future directions. *Algorithms*, *10*(2), p.39.

17. Mohammadi, S., Mirvaziri, H., Ghazizadeh-Ahsaee, M. and Karimipour, H., 2019. Cyber intrusion detection by combined feature selection algorithm. *Journal of information security and applications*, *44*, pp.80-88.

18. Tapiador, J.E., Orfila, A., Ribagorda, A. and Ramos, B., 2013. Key-recovery attacks on KIDS, a keyed anomaly detection system. *IEEE Transactions on Dependable and Secure Computing*, *12*(3), pp.312-325.

19. Tapiador, J.E., Orfila, A., Ribagorda, A. and Ramos, B., 2013. Key-recovery attacks on KIDS, a keyed anomaly detection system. *IEEE Transactions on Dependable and Secure Computing*, *12*(3), pp.312-325. 20. Foroughi F, Luksch P. Data science methodology for cybersecurity projects. arXiv preprint arXiv :1803.04219 , 2018.

21. Saxe, J. and Sanders, H., 2018. *Malware Data Science: Attack Detection and Attribution*. No Starch Press.

22. Rainie, L., Anders, J. and Connolly, J., 2014. Cyber attacks likely to increase. *Digital Life in, 2025*.

23. Fischer, E.A., 2005, February. Creating a national framework for cybersecurity: An analysis of issues and options. LIBRARY OF CONGRESS WASHINGTON DC CONGRESSIONAL RESEARCH SERVICE.

24. Craigen, D., Diakun-Thibault, N. and Purse, R., 2014. Defining cybersecurity. *Technology Innovation Management Review*, *4*(10).

25. National Research Council, 2007. *Toward a safer and more secure cyberspace*. National Academies Press.

26. Jang-Jaccard, J. and Nepal, S., 2014. A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, *80*(5), pp.973-993.

27. Mukkamala, S., Sung, A. and Abraham, A., 2005. Cyber security challenges: Designing efficient intrusion detection systems and antivirus tools. *Vemuri, V. Rao, Enhancing Computer Security with Smart Technology.(Auerbach, 2006)*, pp.125-163.

28. Bilge, L. and Dumitraş, T., 2012, October. Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on Computer and communications security* (pp. 833-844).

29. Davi, L., Dmitrienko, A., Sadeghi, A.R. and Winandy, M., 2010, October. Privilege escalation attacks on android. In *international conference on Information security* (pp. 346-360). Springer, Berlin, Heidelberg.

30. Jovičić, B. and Simić, D., 2006. Common web application attack types and security using asp. net. *Computer Science and Information Systems*, *3*(2), pp.83-96.

31. Warkentin, M. and Willison, R., 2009. Behavioral and policy issues in information systems security: the insider threat. *European Journal of Information Systems*, *18*(2), pp.101-105.

32. Kügler, D., 2003, January. "Man in the Middle" Attacks on Bluetooth. In *International Conference on Financial Cryptography* (pp. 149-161). Springer, Berlin, Heidelberg.

33. Virvilis, N. and Gritzalis, D., 2013, September. The big four-what we did wrong in advanced persistent threat detection?. In *2013 international conference on availability, reliability and security* (pp. 248-254). IEEE.

34. Boyd, S.W. and Keromytis, A.D., 2004, June. SQLrand: Preventing SQL injection attacks. In *International Conference on Applied Cryptography and Network Security* (pp. 292-302). Springer, Berlin, Heidelberg.

35. Sigler, K., 2018. Crypto-jacking: how cyber-criminals are exploiting the crypto-currency boom. *Computer Fraud & Security*, *2018*(9), pp.12-14.

36. Jartelius, M., 2020. The 2020 Data Breach Investigations Report–a CSO's perspective. *Network Security*, *2020*(7), pp.9-12.

37. Khraisat, A., Gondal, I., Vamplew, P. and Kamruzzaman, J., 2019. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1), pp.1-22.

38. Johnson, L., 2013. *Computer incident response and forensics team management: Conducting a successful incident response*. Newnes.

39. Brahmi, I., Brahmi, H. and Yahia, S.B., 2015, May. A multi-agents intrusion detection system using ontology and clustering techniques. In *IFIP International Conference on Computer Science and its Applications* (pp. 381-393). Springer, Cham.

40. Qu, X., Yang, L., Guo, K., Ma, L., Sun, M., Ke, M. and Li, M., 2019. A survey on the development of self-organizing maps for unsupervised intrusion detection. *Mobile networks and applications*, pp.1-22.

41. Liao, H.J., Lin, C.H.R., Lin, Y.C. and Tung, K.Y., 2013. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, *36*(1), pp.16-24.

42. Alazab, A., Hobbs, M., Abawajy, J. and Alazab, M., 2012, October. Using feature selection for intrusion detection system. In *2012 international symposium on communications and information technologies (ISCIT)* (pp. 296-301). IEEE.

43. Viegas, E., Santin, A.O., Franca, A., Jasinski, R., Pedroni, V.A. and Oliveira, L.S., 2016. Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems. *IEEE Transactions on Computers*, *66*(1), pp.163-177.

44. Xin Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H. and Wang, C., 2018. Machine learning and deep learning methods for cybersecurity. *Ieee access*, *6*, pp.35365-35381.

45. Dutt, I., Borah, S., Maitra, I.K., Bhowmik, K., Maity, A. and Das, S., 2018. Real-time hybrid intrusion detection system using machine learning techniques. In *Advances in Communication, Devices and Networking* (pp. 885-894). Springer, Singapore.

46. Ragsdale, D.J., Carver, C.A., Humphries, J.W. and Pooch, U.W., 2000, October. Adaptation techniques for intrusion detection and intrusion response systems. In *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0* (Vol. 4, pp. 2344-2349). IEEE.

47. Cao, L., 2017. Data science: challenges and directions. *Communications of the ACM*, *60*(8), pp.59-68.

48. Rizk, A. and Elragal, A., 2020. Data science: developing theoretical contributions in information systems via text analytics. *Journal of Big Data*, *7*(1), pp.1-26.

49. Lippmann, R.P., Fried, D.J., Graf, I., Haines, J.W., Kendall, K.R., McClung, D., Weber, D., Webster, S.E., Wyschogrod, D., Cunningham, R.K. and Zissman, M.A., 2000, January. Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00* (Vol. 2, pp. 12-26). IEEE.

50. Drewek-Ossowicka, A., Pietrołaj, M. and Rumiński, J., 2020. A survey of neural networks usage for intrusion detection systems. *Journal of Ambient Intelligence and Humanized Computing*, pp.1-18.

51. Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A.A., 2009, July. A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1-6). IEEE.

52. Hoque, N., Kashyap, H. and Bhattacharyya, D.K., 2017. Real-time DDoS attack detection using FPGA. *Computer Communications*, *110*, pp.48-58.

53. Brownlee, N., 2012, March. One-way traffic monitoring with iatmon. In *International Conference on Passive and Active Network Measurement* (pp. 179-188). Springer, Berlin, Heidelberg.

54. Elhadad, M.K., Li, K.F. and Gebali, F., 2019, August. Fake news detection on social media: a systematic survey. In *2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)* (pp. 1-8). IEEE.

55. Spitzner, L., 2003. The honeynet project: Trapping the hackers. *IEEE Security & Privacy*, *1*(2), pp.15-23.

56. Panigrahi, R. and Borah, S., 2020. A Statistical Analysis of Lazy Classifiers Using Canadian Institute of Cybersecurity Datasets. In *Advances in Data Science and Management* (pp. 215-222). Springer, Singapore.

57. Shiravi, A., Shiravi, H., Tavallaee, M. and Ghorbani, A.A., 2012. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *computers & security*, *31*(3), pp.357-374.

58. Garcia, S. and Uhlir, V., 2019. The CTU-13 dataset. a labeled dataset with botnet, normal and background traffic.

59. Moustafa, N. and Slay, J., 2015, November. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 military communications and information systems conference (MilCIS)* (pp. 1-6). IEEE.

60. Sharafaldin, I., Lashkari, A.H. and Ghorbani, A.A., 2018, January. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSp* (pp. 108-116).

61. Yulianto, A., Sukarno, P. and Suwastika, N.A., 2019, March. Improving adaboost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset. In *Journal of Physics: Conference Series* (Vol. 1192, No. 1, p. 012018). IOP Publishing.

62. Jing, X., Yan, Z., Jiang, X. and Pedrycz, W., 2019. Network traffic fusion and analysis against DDoS flooding attacks with a novel reversible sketch. *Information Fusion*, *51*, pp.100-113.

63. Xie, M., Hu, J., Yu, X. and Chang, E., 2015, November. Evaluating host-based anomaly detection systems: Application of the frequency-based algorithms to ADFA-LD. In *International Conference on Network and System Security* (pp. 542-549). Springer, Cham.

64. Lindauer, B., Glasser, J., Rosen, M., Wallnau, K.C. and ExactData, L., 2014. Generating Test Data for Insider Threat Detectors. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, *5*(2), pp.80-94.

65. Glasser, J. and Lindauer, B., 2013, May. Bridging the gap: A pragmatic approach to generating insider threat data. In *2013 IEEE Security and Privacy Workshops* (pp. 98-104). IEEE.

66. Kumar, C., 2016. Apache Web Server Hardening & Security Guide. *Retrieved February*, *2*, p.2017.

67. Xia, T., 2020. A constant time complexity spam detection algorithm for boosting throughput on rule-based filtering systems. *IEEE Access*, *8*, pp.82653-82661.

68. Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Paliouras, G. and Spyropoulos, C.D., 2000. An evaluation of naive bayesian anti-spam filtering. *arXiv preprint cs/0006013*.

69. Hoy, M.B., 2018. Alexa, Siri, Cortana, and more: an introduction to voice assistants. *Medical reference services quarterly*, *37*(1), pp.81-88.

70. Treglia, J. and Delia, M., 2017, June. Cyber Security Inoculation. In *NYS Cyber Security Conference, Empire State Plaza Convention Center, Albany, NY, June* (pp. 3-4).

71. Almashhadani, A.O., Kaiiali, M., Carlin, D. and Sezer, S., 2020. MaldomDetector: A system for detecting algorithmically generated domain names with machine learning. *Computers & Security*, *93*, p.101787.

72. Zago, M., Pérez, M.G. and Pérez, G.M., 2020. Umudga: A dataset for profiling algorithmically generated domain names in botnet detection. *Data in brief*, *30*, p.105400.

73. Zhou, Y. and Jiang, X., 2012, May. Dissecting android malware: Characterization and evolution. In *2012 IEEE symposium on security and privacy* (pp. 95-109). IEEE.

74. Popov, I., 2017, April. Malware detection using machine learning based on word2vec embeddings of machine code instructions. In *2017 Siberian symposium on data science and engineering (SSDSE)* (pp. 1-4). IEEE.

75. Peng, P., Yang, L., Song, L. and Wang, G., 2019, October. Opening the blackbox of virustotal: Analyzing online phishing scan engines. In *Proceedings of the Internet Measurement Conference* (pp. 478-485).

76. Lee, Y. and Larsen, K.R., 2009. Threat or coping appraisal: determinants of SMB executives' decision to adopt anti-malware software. *European Journal of Information Systems*, *18*(2), pp.177-187.

77. Maiorca, D., Ariu, D., Corona, I., Aresu, M. and Giacinto, G., 2015. Stealth attacks: An extended insight into the obfuscation effects on android malware. *Computers & Security*, *51*, pp.16-31.

78. Kumar, R., Xiaosong, Z., Khan, R.U., Kumar, J. and Ahad, I., 2018, March. Effective and explainable detection of android malware based on machine learning algorithms. In *Proceedings of the 2018 International Conference on Computing and Artificial Intelligence* (pp. 35-40).

79. Ronen, R., Radu, M., Feuerstein, C., Yom-Tov, E. and Ahmadi, M., 2018. Microsoft malware classification challenge. *arXiv preprint arXiv:1802.10135*.

80. Koroniotis, N., Moustafa, N., Sitnikova, E. and Turnbull, B., 2019. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, *100*, pp.779-796.

81. McIntosh, T.R., Jang-Jaccard, J. and Watters, P.A., 2018, December. Large scale behavioral analysis of ransomware attacks. In *International Conference on Neural Information Processing* (pp. 217-229). Springer, Cham.

82. Han, J., Kamber, M. and Pei, J., 2011. Data mining concepts and techniques third edition. *The Morgan Kaufmann Series in Data Management Systems*, *5*(4), pp.83-124.

83. Witten, I.H., Frank, E., Hall, M.A. and Pal, C.J., 2005. Practical machine learning tools and techniques. *Morgan Kaufmann*, p.578.

84. Dua, S. and Du, X., 2016. *Data mining and machine learning in cybersecurity*. CRC press.

85. Kotpalliwar, M.V. and Wajgi, R., 2015, April. Classification of Attacks Using Support Vector Machine (SVM) on KDDCUP'99 IDS Database. In *2015 Fifth International Conference on Communication Systems and Network Technologies* (pp. 987-990). IEEE.

86. Pervez, M.S. and Farid, D.M., 2014, December. Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. In *The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014)* (pp. 1-6). IEEE.

87. Yan, M. and Liu, Z., 2010, October. A new method of transductive SVM-based network intrusion detection. In *International Conference on Computer and Computing Technologies in Agriculture* (pp. 87-95). Springer, Berlin, Heidelberg.

88. Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X. and Dai, K., 2012. An efficient intrusion detection system based on support vector machines and gradually feature removal method. *Expert systems with applications*, *39*(1), pp.424-430.

89. Raman, M.G., Somu, N., Jagarapu, S., Manghnani, T., Selvam, T., Krithivasan, K. and Sriram, V.S., 2019. An efficient intrusion detection technique based on support vector machine and improved binary gravitational search algorithm. *Artificial Intelligence Review*, pp.1-32.

90. Kokila, R.T., Selvi, S.T. and Govindarajan, K., 2014, December. DDoS detection and analysis in SDN-based environment using support vector machine classifier. In *2014 Sixth International Conference on Advanced Computing (ICoAC)* (pp. 205-210). IEEE.

91. Xie, M., Hu, J. and Slay, J., 2014, August. Evaluating host-based anomaly detection systems: Application of the one-class SVM algorithm to ADFA-LD. In *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)* (pp. 978-982). IEEE.

92. Saxena, H. and Richariya, V., 2014. Intrusion detection in KDD99 dataset using SVM-PSO and feature reduction with information gain. *International Journal of Computer Applications*, *98*(6).

93. Chandrasekhar, A.M. and Raghuveer, K., 2014, April. Confederation of fcm clustering, ann and svm techniques to implement hybrid nids using corrected kdd cup 99 dataset. In *2014 International Conference on Communication and Signal Processing* (pp. 672-676). IEEE.

94. Shapoorifard, H. and Shamsinejad, P., 2017. Intrusion detection using a novel hybrid method incorporating an improved KNN. *Int. J. Comput. Appl*, *173*(1), pp.5-9.

95. Vishwakarma, S., Sharma, V. and Tiwari, A., 2017. An intrusion detection system using KNN-ACO algorithm. *Int J Comput Appl*, *171*(10), pp.18-23.

96. Meng, W., Li, W. and Kwok, L.F., 2015. Design of intelligent KNN-based alarm filter using knowledge-based alert verification in intrusion detection. *Security and Communication Networks*, *8*(18), pp.3883-3895.

97. Dada, E.G., 2017. A hybridized SVM-kNN-pdAPSO approach to intrusion detection system. In *Proc. Fac. Seminar Ser* (pp. 14-21).

98. Sharifi, A.M., Amirgholipour, S.K. and Pourebrahimi, A., 2015. Intrusion detection based on joint of K-means and KNN. *Journal of Convergence Information Technology*, *10*(5), p.42.

99. Lin, W.C., Ke, S.W. and Tsai, C.F., 2015. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-based systems*, *78*, pp.13-21.

100. Koc, L., Mazzuchi, T.A. and Sarkani, S., 2012. A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. *Expert Systems with Applications*, *39*(18), pp.13492-13500.

101. Moon, D., Im, H., Kim, I. and Park, J.H., 2017. DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks. *The Journal of supercomputing*, *73*(7), pp.2881-2895.

102. Ingre, B., Yadav, A. and Soni, A.K., 2017, March. Decision tree based intrusion detection system for NSL-KDD dataset. In *International conference on information and communication technology for intelligent systems* (pp. 207-218). Springer, Cham.

103. Malik, A.J. and Khan, F.A., 2018. A hybrid technique using binary particle swarm optimization and decision tree pruning for network intrusion detection. *Cluster Computing*, *21*(1), pp.667-680.

104. Relan, N.G. and Patil, D.R., 2015, January. Implementation of network intrusion detection system using variant of decision tree algorithm. In *2015 International Conference on Nascent Technologies in the Engineering Field (ICNTE)* (pp. 1-5). IEEE.

105. Rai, K., Devi, M.S. and Guleria, A., 2016. Decision tree based algorithm for intrusion detection. *International Journal of Advanced Networking and Applications*, *7*(4), p.2828.

106. Sarker, I.H., Abushark, Y.B., Alsolami, F. and Khan, A.I., 2020. Intrudtree: a machine learning based cyber security intrusion detection model. *Symmetry*, *12*(5), p.754.

107. Puthran, S. and Shah, K., 2016, September. Intrusion detection using improved decision tree algorithm with binary and quad split. In *International symposium on security in computing and communication* (pp. 427-438). Springer, Singapore.

108. Balogun, A.O. and Jimoh, R.G., 2015. Anomaly intrusion detection using an hybrid of decision tree and K-nearest neighbor.

109. Azad, C. and Jha, V.K., 2015. Genetic algorithm to solve the problem of small disjunct in the decision tree based intrusion detection system. *International Journal of Computer Network and Information Security*, *7*(8), pp.56-71.

110. Jo, S., Sung, H. and Ahn, B., 2015. A comparative study on the performance of intrusion detection using decision tree and artificial neural network models. *Journal of the Korea Society of Digital Industry and Information Management*, *11*(4), pp.33-45.

111. Zhang, J., Zulkernine, M. and Haque, A., 2008. Random-forests-based network intrusion detection systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(5), pp.649-659.

112. Tajbakhsh, A., Rahmati, M. and Mirzaei, A., 2009. Intrusion detection using fuzzy association rules. *Applied Soft Computing*, *9*(2), pp.462-469.

113. Mitchell, R. and Chen, R., 2014. Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems. *IEEE Transactions on Dependable and Secure Computing*, *12*(1), pp.16-30.

114. Alazab, M., Venkataraman, S. and Watters, P., 2010, July. Towards understanding malware behaviour by the extraction of API calls. In *2010 second cybercrime and trustworthy computing workshop* (pp. 52-59). IEEE.

115. Yuan, Y., Kaklamanos, G. and Hogrefe, D., 2016, November. A novel semi-supervised adaboost technique for network anomaly detection. In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (pp. 111-114).

116. Ariu, D., Tronci, R. and Giacinto, G., 2011. HMMPayl: An intrusion detection system based on Hidden Markov Models. *computers & security*, *30*(4), pp.221-241.

117. Arnes, A., Valeur, F., Vigna, G. and Kemmerer, R.A., 2006. Using hidden Markov models to evaluate the risks of intrusions: system architecture and model validation. *Lecture notes in computer science*, pp.145-164.

118. Hansen, J.V., Lowry, P.B., Meservy, R.D. and McDonald, D.M., 2007. Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection. *Decision Support Systems*, *43*(4), pp.1362-1374.

119. Aslahi-Shahri, B.M., Rahmani, R., Chizari, M., Maralani, A., Eslami, M., Golkar, M.J. and Ebrahimi, A., 2016. A hybrid method consisting of GA and SVM for intrusion detection system. *Neural computing and applications*, *27*(6), pp.1669-1676.

120. Alrawashdeh, K. and Purdy, C., 2016, December. Toward an online anomaly intrusion detection system based on deep learning. In *2016 15th IEEE international conference on machine learning and applications (ICMLA)* (pp. 195-200). IEEE.

121. Yin, C., Zhu, Y., Fei, J. and He, X., 2017. A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, *5*, pp.21954-21961.

122. Kim, J., Kim, J., Thu, H.L.T. and Kim, H., 2016, February. Long short term memory recurrent neural network classifier for intrusion detection. In *2016 International Conference on Platform Technology and Service (PlatCon)* (pp. 1-5). IEEE.

123. Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S. and Razaque, A., 2020. Deep recurrent neural network for IoT intrusion detection system. *Simulation Modelling Practice and Theory*, *101*, p.102031.

124. Kolosnjaji B, Zarras A, Webster G, Eckert C. Deep learning for classification of malware system call sequences. In: Australasian joint conference on artificial intelligence. New York: Springer; 2016. p. 137–49.

125. Wang W, Zhu M, Zeng X, Ye X, Sheng Y. Malware traffic classification using convolutional neural network for representation learning. In: 2017 international conference on information networking (ICOIN). IEEE; 2017. p. 712–17.

126. Alauthman M, Aslam N, Al-kasassbeh M, Khan S, Al-Qerem A, Choo K-KR. An efficient reinforcement learningbased botnet detection approach. J Netw Comput Appl. 2020;150:102479.

127. Blanco R, Cilla JJ, Briongos S, Malagón P, Moya JM. Applying cost-sensitive classifiers with reinforcement learning to ids. In: International conference on intelligent data engineering and automated learning. New York: Springer; 2018. p. 531–38.

128. Lopez-Martin M, Carro B, Sanchez-Esguevillas A. Application of deep reinforcement learning to intrusion detection for supervised problems. Exp Syst Appl. 2020;141:112963.

129. Ferreira, J., Carvalho, E., Ferreira, B.V., de Souza, C., Suhara, Y., Pentland, A. and Pessin, G., 2017. Driver behavior profiling: An investigation with different smartphone sensors and machine learning. PLoS one, 12(4), p.e0174959.

130. Holte RC. Very simple classification rules perform well on most commonly used datasets. Mach Learn. 1993;11(1):63–90.

131. John GH, Langley P. Estimating continuous distributions in bayesian classifiers. In: Proceedings of the eleventh conference on uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.; 1995. p. 338–45.

132. Salzberg, S.L., 1994. C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993.

133. Hwang, G.J., Chu, H.C., Shih, J.L., Huang, S.H. and Tsai, C.C., 2010. A decision-tree-oriented guidance mechanism for conducting nature science observation activities in a context-aware ubiquitous learning environment. Journal of Educational Technology & Society, 13(2), pp.53-64.

134. Aha, D.W., Kibler, D. and Albert, M.K., 1991. Instance-based learning algorithms. *Machine learning*, *6*(1), pp.37-66.

135. Keerthi SS, Shevade SK, Bhattacharyya C, Murthy KRK. Improvements to platt's smo algorithm for svm classifier design. Neural Comput. 2001;13(3):637–49.

136. Freund, Y. and Schapire, R.E., 1996, July. Experiments with a new boosting algorithm. In *icml* (Vol. 96, pp. 148-156).

137. Le Cessie, S. and Van Houwelingen, J.C., 1992. Ridge estimators in logistic regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *41*(1), pp.191-201.

138. Watters, P.A., McCombie, S., Layton, R. and Pieprzyk, J., 2012. Characterising and predicting cyber attacks using the Cyber Attacker Model Profile (CAMP). *Journal of Money Laundering Control*.

139. Breiman, L., 2001. Random forests. Machine learning, 45(1), pp.5-32.

140. Luo, C., Goncalves, J., Velloso, E. and Kostakos, V., 2020. A survey of context simulation for testing mobile context-aware applications. ACM Computing Surveys (CSUR), 53(1), pp.1-39.

141. MacQueen, J., 1967, June. Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).

142. Ricci, F., Rokach, L. and Shapira, B., 2011. Introduction to recommender systems handbook. In Recommender systems handbook (pp. 1-35). Springer, Boston, MA.

143. Sneath, P.H., 1957. The application of computers to taxonomy. Microbiology, 17(1), pp.201-226.

144. Sorensen, T.A., 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. Biol. Skar., 5, pp.1-34.

145. Sarker, I.H., Colman, A., Kabir, M.A. and Han, J., 2018. Individualized time-series segmentation for mining mobile phone user behavior. The Computer Journal, 61(3), pp.349-368.

146. Kim, G., Lee, S. and Kim, S., 2014. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Systems with Applications, 41(4), pp.1690-1700.

147. Agrawal, R., Imieliński, T. and Swami, A., 1993, June. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD international conference on Management of data (pp. 207-216).

148. Flach, P.A. and Lachiche, N., 2001. Confirmation-guided discovery of first-order rules with Tertius. Machine learning, 42(1), pp.61-95.

149. Agrawal, R. and Srikant, R., 1994, September. Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).

150. Houtsma, M. and Swami, A., 1995, March. Set-oriented mining for association rules in relational databases. In Proceedings of the eleventh international conference on data engineering (pp. 25-33). IEEE.

151. Liu, B., Hsu, W. and Ma, Y., 1998, August. Integrating classification and association rule mining. In Kdd (Vol. 98, pp. 80-86).

152. Han, J., Pei, J. and Yin, Y., 2000. Mining frequent patterns without candidate generation. ACM sigmod record, 29(2), pp.1-12.

153. Versichele, M., De Groote, L., Bouuaert, M.C., Neutens, T., Moerman, I. and Van de Weghe, N., 2014. Pattern mining in tourist attraction visits through association rule learning on Bluetooth tracking data: A case study of Ghent, Belgium. Tourism Management, 44, pp.67-81.

154. Das, A., Ng, W.K. and Woon, Y.K., 2001, October. Rapid association rule mining. In Proceedings of the tenth international conference on Information and knowledge management (pp. 474-481).

155. Zaki, M.J., 2000. Scalable algorithms for association mining. IEEE transactions on knowledge and data engineering, 12(3), pp.372-390.

156. Coelho, I.M., Coelho, V.N., Luz, E.J.D.S., Ochi, L.S., Guimarães, F.G. and Rios, E., 2017. A GPU deep learning metaheuristic based model for time series forecasting. Applied Energy, 201, pp.412-418.

157. Van Efferen, L. and Ali-Eldin, A.M., 2017, May. A multi-layer perceptron approach for flow-based anomaly detection. In 2017 International Symposium on Networks, Computers and Communications (ISNCC) (pp. 1-6). IEEE.

158. Liu, H., Lang, B., Liu, M. and Yan, H., 2019. CNN and RNN based payload classification methods for attack detection. Knowledge-Based Systems, 163, pp.332-341.

159. Berman, D.S., Buczak, A.L., Chavis, J.S. and Corbett, C.L., 2019. A survey of deep learning methods for cyber security. Information, 10(4), p.122.

160. Bellman, R., 1957. A Markovian decision process. Journal of mathematics and mechanics, 6(5), pp.679-684.

161. Kaelbling, L.P., Littman, M.L. and Moore, A.W., 1996. Reinforcement learning: A survey. Journal of artificial intelligence research, 4, pp.237-285.

162. Mair, C., Kadoda, G., Lefley, M., Phalp, K., Schofield, C., Shepperd, M. and Webster, S., 2000. An investigation of machine learning based prediction systems. Journal of systems and software, 53(1), pp.23-29.

163. Kayes, A.S.M., Han, J. and Colman, A., 2013, October. An ontology-based approach to context-aware access control for software services. In International Conference on Web Information Systems Engineering (pp. 410-420). Springer, Berlin, Heidelberg.

164. Kayes, A.S.M., Rahayu, W. and Dillon, T., 2018, May. An ontology-based approach to dynamic contextual role for pervasive access control. In 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA) (pp. 601-608). IEEE.

165. Colombo, P. and Ferrari, E., 2019. Access control technologies for Big Data management systems: literature review and future trends. Cybersecurity, 2(1), pp.1-13.

166. Aleroud, A. and Karabatis, G., 2017. Contextual information fusion for intrusion detection: a survey and taxonomy. Knowledge and Information Systems, 52(3), pp.563-619.

167. Olejnik, K., Dacosta, I., Machado, J.S., Huguenin, K., Khan, M.E. and Hubaux, J.P., 2017, May. Smarper: Context-aware and automatic runtime-permissions for mobile devices. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 1058-1076). IEEE.

168. Wall, M.E., Rechtsteiner, A. and Rocha, L.M., 2003. Singular value decomposition and principal component analysis. In A practical approach to microarray data analysis (pp. 91-109). Springer, Boston, MA.

169. Qiao, L.B., Zhang, B.F., Lai, Z.Q. and Su, J.S., 2012, May. Mining of attack models in ids alerts from network backbone by a two-stage clustering method. In 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & Phd Forum (pp. 1263-1269). IEEE.

170. Saeed, A. and Kolberg, M., 2018. Towards optimizing WLANs power saving: Novel context-aware network traffic classification based on a machine learning approach. IEEE Access, 7, pp.3122-3135.

171. Ullah, F. and Babar, M.A., 2019. Architectural tactics for big data cybersecurity analytics systems: a review. Journal of Systems and Software, 151, pp.81-118.

172. Zhao, S., Leftwich, K., Owens, M., Magrone, F., Schonemann, J., Anderson, B. and Medhi, D., 2014, May. I-can-mama: Integrated campus network monitoring and management. In 2014 IEEE Network Operations and Management Symposium (NOMS) (pp. 1-7). IEEE.

173. Abomhara, M. and Køien, G.M., 2015. Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks. Journal of Cyber Security and Mobility, pp.65-88.

174. Helali, R.G.M., 2010. Data mining based network intrusion detection system: A survey. In Novel Algorithms and Techniques in Telecommunications and Networking (pp. 501-505). Springer, Dordrecht.

175. Ryoo, J., Rizvi, S., Aiken, W. and Kissell, J., 2013. Cloud security auditing: challenges and emerging approaches. IEEE Security & Privacy, 12(6), pp.68-74.

176. Densham, B., 2015. Three cyber-security strategies to mitigate the impact of a data breach. Network Security, 2015(1), pp.5-8.

177. Salah, K., Rehman, M.H.U., Nizamuddin, N. and Al-Fuqaha, A., 2019. Blockchain for AI: Review and open research challenges. IEEE Access, 7, pp.10127-10149.

178. Gandomi, A. and Haider, M., 2015. Beyond the hype: Big data concepts, methods, and analytics. International journal of information management, 35(2), pp.137-144.

179. Berman, E., Felter, J.H. and Shapiro, J.N., 2020. Small wars, big data: the information revolution in modern conflict. Princeton University Press.

180. Hariri, R.H., Fredericks, E.M. and Bowers, K.M., 2019. Uncertainty in big data analytics: survey, opportunities, and challenges. Journal of Big Data, 6(1), pp.1-16.

181. Tsai, C.W., Lai, C.F., Chao, H.C. and Vasilakos, A.V., 2015. Big data analytics: a survey. Journal of Big data, 2(1), pp.1-32.

3. SMOTE IMPLEMENTATION ON PHISHING DATA TO ENHANCE CYBERSECURITY[1]

3.1. Abstract

Phishing is a form of cybersecurity threat where the criminal tries to gain access to user's personal information by infecting their system using malware and viruses. Appearing to come from legitimate sources, it is very easy to fall into the phishing scam. As each phishing data contains features that are consistently different from another, using a predefined set of rules makes a system useless. Data mining techniques can be applied to collected network traffic and build models to predict future attacks. However, since most of the data packets are legitimate, the model tends to produce a bias towards positive results in this imbalanced dataset. In this study, we investigate how prediction accuracy varies in a balanced dataset against an imbalanced one. SMOTE is applied to balance the dataset. XGBoost, Random Forest, and Support Vector Machines have been applied to the phishing dataset. Results show much higher accuracy rates with SMOTE applications. The highest jump inaccuracy has been recorded in XGBoost from 89.87% to 97.17%, showing that SMOTE is an effective tool in phishing data monitoring.

3.2. Introduction

The consistent growth of the internet and information technology solutions has made our society, economy, and financial structures vastly dependent on it. This significant growth of online social interactions and trading has led to an increased amount of cyber-attacks, often with disastrous outcomes [1]. Even though, the cautiousness and security has increased, but, cyber threats are getting more advanced with mixing of once particular sorts of more harmful shapes

[1] The material in this chapter is co-authored by Mostofa Ahsan, Rahul Gomes and Anne Denton. Mostofa Ahsan had primary responsibility for all the experiments and development of conclusions. Rahul Gomes was responsible for finding suitable data source. Anne Denton served as proofreader. Mostofa Ahsan also drafted and revised all version of this chapter [39].

[2]. Phishing is the primary choice of weapon to attain malicious intents in cyberspace. Phishing is considered as a form of cyber threat that is described as the art of mimicking websites of an honest enterprise aiming and to acquire confidential information such as username, passwords and financial data by deceiving users [3]. Still, now, no such solution has been made that can prevent every phishing attack, though a lot of innovative and effective defense mechanisms have been proposed. Since the historical data of phishing with distinct features are now available, one of the promising approach which can be employed in preventing phishing attacks is using machine learning techniques which will classify malignant e-mails [4]. Various data-mining algorithms like Support Vector Machine (SVM), Random Forest (RF), Associative Classification and intelligent model like Artificial Neural Networks are proven effective for fault-tolerant model against phishing attack [5][6][3][4][7]. However, researchers are facing an issue which is the scarcity of actual phishing website data compared to benign website data in training datasets. This problem leads to imbalanced and biased learning of classification which is one of the major causes of degrading the accuracy of machine learning model predictions [8].

In this paper we propose a solution to this issue by applying the Synthetic Minority Over-sampling Technique (SMOTE) on the website phishing dataset to deal with the class imbalance [9]. SMOTE is proven effective in achieving higher accuracy in land cover, credit card fraud detection, bioinformatics and other domains [10][11][12]–[15][16][17]. Three machine learning techniques such as XGBoost (XGB), SVM, RF has been evaluated. Results have been compared with SMOTE and without SMOTE application on them. This paper is organized as follows: In section 2 we have discussed about the previous work. Section 3 describes SMOTE and its application on imbalanced dataset. Section 4 discuss the dataset and its features. Section 5 talks

about the algorithms used. Section 6 explains the experiments and results. And finally, in section 7 has been discussed the conclusion and future work.

3.3. Related work

The number of phishing attacks are increasing over time. So this problem needs a smart solution to cope with its evolving nature [5]. Different countries and societies have applied various approaches like legal actions, educating users, and increasing awareness [18][19]. However, these non-technical actions cannot stop intrusions in cyberspace. With the increased amounts of phishing attacks, quite a lot of anti-phishing solutions are offered like browser extensions, plug-ins, and filtering tools, which are not efficient in making an accurate decision and hence raises the false-positive [4][6]. The familiar blacklist method compares with a predefined phishing URL. However, it cannot deal with newly launched threats [5]. Some Fuzzy rule-based approaches have been used to classify the phishing data as legitimate, suspicious, phishy, and deal with a wide range of features but cannot explain the fine line that separates the labels [20]. Carnegie Mellon Anti-phishing and Network Analysis Tool (CANTINA) is a content-based technique that uses Term-frequency-inverse-Document-Frequency (TF-IDF). It assigns weights and makes the assessment of the document's word, then counts its frequency [21][22]. This method shows some limitations while tested on legitimate websites consisting of images and hidden text [5].

Nowadays, as the prevalence of historical phishing data is increasing, machine learning algorithms have proven effective for the detection of phishing attacks [5][6][3][4][7]. Support Vector Machine (SVM) is a commonly used method that helps to deal with classification problems [23]. To detect unusual activities like spam and phishing, a method was proposed using SVM based on two variables. The first variable denotes domain name and the second one poses

the page category which describes structural features that are tough to duplicate [24]. In some extended works, researchers collected data with more features and instances and compared some familiar machine-learning methods, including SVM, Decision tree and Naïve Bayes on datasets from the various domain, including phishing dataset [25][26]. SVM has a high dimensional space with a better functional margin, which helped to produce better accuracy. Later, "Phishing Identification by Learning on Features of email Received" (PILFER) was introduced by the authors, which is a revised version of the Random Forest (RF) algorithm. They tested PILFER on a dataset that contained 860 real phishing e-mails and 695 legitimate e-mails. Since the dataset was balanced, it has good accuracy for identifying fake e-mails[26]. In 2006 Pan et al. [24] used six-featured SVM on a dataset which contained 297 phishing and 100 legitimate website data. Their overall error rate was at 16% because of the imbalanced dataset. Using the attributes from the heuristic features of CANTINA, Daisuke et al. [27] compared the efficiency of nine machine learning techniques with a balanced dataset of 1500 each for phish and legitimate, which showed an average error rate of 14.67%. In [7], the authors replaced some features from CANTINA provided dataset and used six learning methods with a balanced dataset of 100 each, which showed an improved accuracy of 92.5% using Neural Network and 91% using AdaBoost. But, in real time scenario, in most of the cases, the phishing data is not balanced, which leads to biased learning for training dataset.

3.4. SMOTE

Imbalanced datasets are rapidly emerging in many practical applications and arising a challenge for researchers in every domain. The majority of the established classification approaches are developed based on the assumption that the underlying training data will be balanced [28]. When the training dataset is highly imbalanced, we can distribute them into two

classes such as majority and minority. But this leads to a severely biased decision boundary to the minority class and suffers a poor performance according to the receiver operator characteristic curve (ROC). To prevent this issue, many classification algorithms have been improved, such as under-sampling the majority classes, oversampling the minority classes, various cost-sensitive learning and feature selection techniques.

SMOTE is a solution for class imbalance and it oversamples the minority class without replicating them [9]. It has proven very effective in dealing cloud cover Landsat data using RF, NB and Decision tree by increasing the overall accuracy by 7%, 4% and 11%, respectively with using SMOTE over without using SMOTE[11]. In [15] for fraud detection, authors used both random undersampling and SMOTE to oversample data. First the minority class data outliers were cleaned using k-Reverse Nearest Neighbor (kRNN) and then it was oversampled using SMOTE. In the same time, majority class was randomly undersampled and then combined with the minority class to make samples for the training dataset. This technique resulted from a higher accuracy of 81.5%. A variant of SMOTE blended with cost algorithm is applied prior to SVM method and then tested on ten different imbalanced datasets from UCI (University of California Irvine Machine Learning Repository). SMOTE approach outperformed the accuracy of all current work [12]. A combination of genetic algorithm and SMOTE is used for solving the class imbalance problem in [29], which gave the best accuracy with SVM.

Phishing website datasets are highly skewed as the prevalence of actual phishing data is rare in them. Using different under-sampling methods and removing the extreme outliers can produce a biased result. So, this is a perfect domain to apply SMOTE which can improve the predictions of familiar machine learning methods.

3.5. Datasets

In this paper we have used the dataset from UCI Machine Learning Repository [5]. This multi-variate dataset contains nine attributes that can distinguish phishing websites from legitimate ones. Server From Handler (SFH): When the user information is submitted, the webpage sends the data to the server for processing. Usually, webpage loading and information handling is done by the same domain. Phishers change this situation by making the server form handling empty or rerouting the information somewhere different than the legitimate domain.

Pop Up Window: Normally, legitimate sites do not ask users to submit their accreditation through pop up window.

SSL (Secure Sockets Layer) Final State: When the user is browsing a legitimate website, it is reflected by the presence of HTTPS protocol for every time-sensitive information. But the phishers use the false HTTPS protocol to trick the users. Verifying the HTTPS protocol, which is now offered by some responsible issuer, is recommended.

Request URL (Uniform Resource Locator): Generally, a webpage is formed with text, images, and videos. Usually, these objects are loaded from the same server where the webpage is stored. If the contents do not load from the same server, then we can flag it as phishy.

URL of Anchor: It is very much like the URL feature except, the links on the page directs to a domain that is not the domain typed in the address bar.

Web Traffic: Genuine websites have more traffic than phishing ones because they are visited frequently. As the fake websites usually have a short span of life, they don’t have so much traffic and have a low rank.

URL Length: Phishers cover up the suspicious portion of URL to divert the user’s submitted information or transfer page to fake domains. Technically, there is no standard length

that makes a fine line between phishing and legitimate ones. In [30] the authors suggested that URL length greater than 54 characters may be phishy.

Age of Domain: Webpages that have a long span of existence, like 1 year, can be considered legit.

Having IP: When an IP address is showed on the domain name of the URL, it means that someone is intentionally trying to access personal information. This kind of trick may contain a link that will start with an IP address that was previously familiar in company websites. About 20% of data that contain this type of IP address can be classified as Phishing websites.

Features of the dataset are categorized as Legitimate, Suspicious and Phishy which have been replaced by 1, 0 and -1 respectively. Table 3.1 highlights each of the previously mentioned attributes in the dataset combined with the logic used to classify them into Legitimate, Suspicious and Phishy.

Table 3.1: Feature description of dataset.

Feature	Logic	Result
SFH	Blank ∨ empty	-1
	Diverts to different Domain	0
	Else	1
Pop-up Window	Right Click Disabled	-1
	Right Click showing Alert	0
	Else	1
SSL Final State	https ∧ trusted issuer ∧ age ≥ 2years	1
	https ∧ issuer is not trusted	0
	Else	-1
Request URL	URL ≤ 22%	1
	URL ≥ 22% ∧ <61%	0
	Else	-1
URL anchor	URL anchor % < 31%	1
	URL anchor % ≥ 31% ∧ ≤67%	0
	Else	-1
Web Traffic	Web Traffic < 150000	1
	Web Traffic >150000	0
	Else	-1
URL Length	URL length < 54	1
	URL length ≥ 54 ∧ ≤ 75	0
	Else	-1
Age of Domain	Age ≤ 6 Months	1
	Else	-1
Having IP	IP address exists in URL	-1
	Else	1

3.6. Algorithms used

For any machine learning technique, it is essential to choose both method and parameter to achieve high level performance from the predictive learning model [31]. Machine learning libraries of today, allows the user to run simulations with varying conditions to determine which parameters produce a desirable solution. In this paper, we have selected machine learning algorithms SVM, RF and XGBoost for the classification of phishing data.

3.6.1. Support vector machines

SVM is a supervised learning model which is very effective in classification, regression problem and tasks like outlier detections. It constructs a hyperplane or set of infinite-dimensional space that makes it convenient to measure the functional margin. Because of the infinite dimensional space, it produces a large margin which tends to lower to error of the classifier [25]. Beside linear classification, SVM can also perform nonlinear classification.

To perform nonlinear classification, SVM uses kernel trick, which implicitly maps the inputs to high dimensional feature spaces and improves the efficiency in a consistent manner. Two nonlinear kernel functions used in this paper are Polynomial and Gaussian radial basis functions. Since SVM can map data to a higher dimension to accurately determine the hyperplane, this method has a strategic advantage over probabilistic models such as Naïve Bayes that do not increase the dimensionality. SVM usually performs one-against-one calculation for a binary classification problem. For multi-label analysis, a one-against- all approach is used [32].

3.6.2. Random forests

Random forest is an ensemble learning method that is very effective in solving both classification and regression analysis. It operates by creating multiple decision trees and aggregating the results obtained from them to assign a class label. Usually, the records are assigned a class label if they have maximum frequency [33][34]. Later, the bagging idea is introduced to random forest for constructing a collection of decision trees for variance reduction [35]. Bootstrapping method also leads to a better model prediction because it decreases the variance without the increment of bias. Usually, hundred to several thousand trees may be used in random forest depending on the size of the training dataset. But a lesser number of tree can be

used if a cross-validation technique is applied. After applying cross-validation, the training and test error tend to touch down when some number of trees have been fit.

In a way, random forest improves the decision tree's demerit of overfitting to the training data [36]. The random forest also addresses the problem of oversimplification that decision trees suffer from. Since decision trees try to produce a global solution, it prunes the tree, thereby trading accuracy. However, random forests construct the entire tree, thereby generating much more accurate results.

3.6.3. XGBoost

XGBoost is an updated implementation of gradient boosting machines that is designed to be very flexible, efficient, and portable. It solves many data science problems fast and accurately with parallel tree boosting. The algorithm uses multiple parameters, which tends to tune these parameters is essential to achieve higher accuracy. In this paper, we have tuned two parameters named as around and nfold. A number of trees are set by nround and nfold denotes the number of folds for cross-validation [37].

Unlike bagging algorithms, which work by creating decision trees on bootstrap aggregation, boosting method performs multiple iterations to try and maximize accuracy. This method can produce better results than random forest on most occasions. However, since multiple iterations is computationally intensive, some researchers prefer using bagging methods instead of boosting when few points of accuracy can be traded for reduced computation complexity.

3.7. Experiments and results

A 65-35 split ratio was used for training and testing data for stage 1 i.e. without SMOTE application. The original dataset contained 702, 103 and 548 counts corresponding to -1,0 and 1

class labels, respectively. In stage 2, SMOTE was applied on the training dataset only. This ensured that the testing dataset had original values. After applying SMOTE, the total count of all values in the training dataset was 421 each. The ratio was maintained at 65-35 after SMOTE application as well.

Accuracy estimation was done using several runs on the phishing dataset for four different algorithms. The objective of using multiple algorithms was to evaluate how they behave when used with SMOTE and without it. For XGBoost, ten rounds with two-fold cross-validation, fifty rounds with five- fold cross-validation and one hundred rounds with ten-fold cross-validations were implemented. Accuracy obtained after these models were run on testing data is shown in figure 1. Best accuracy was observed for one-hundred rounds with ten-fold cross-validation. The confusion matrix for this model on the testing data is shown in table 3.2. This confirms the relationship between the increasing number of iterations with the accuracy of XGBoost [37]. Accuracy without SMOTE was 89.87%. After SMOTE application, the accuracy surged to 97.17% showing that XGBoost is highly impacted by SMOTE. Figure 3.1 describes the increment of accuracy using SMOTE with XGBoost.

Table 3.2: Confusion matrix of XGBoost

		Reference					
Algorithm	Pred	1	2	3	Total	Sensitivity	Specificity
XGBoost	1	411	4	6	421	.974	.974
	2	0	55	1	56	.917	.999
	3	11	1	323	335	.979	.975
	Total	422	60	330	812		

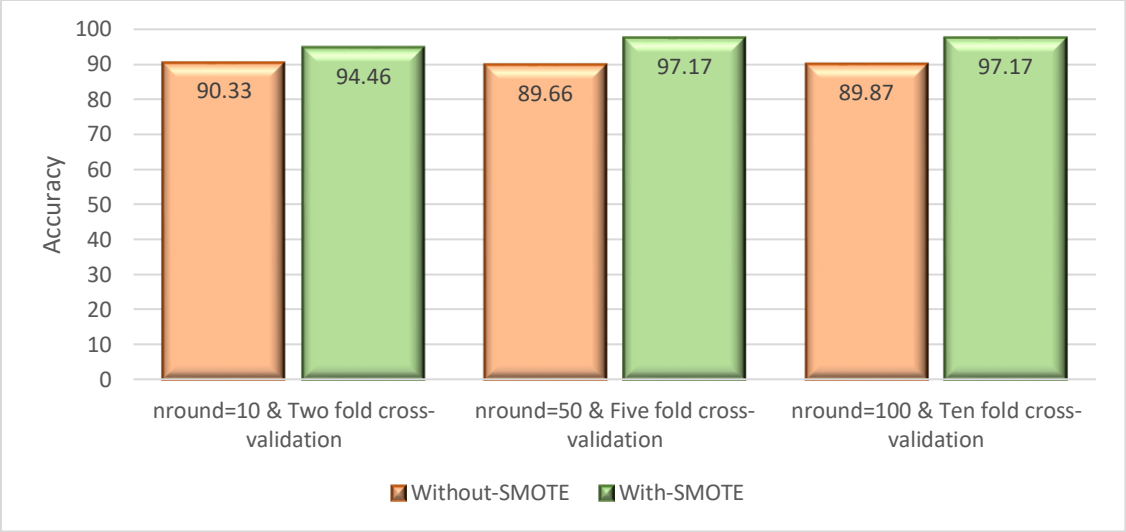

Fig 3.1: Accuracy estimates of XGBoost

The first stage in random forest implementation is the determination of the best mtry values. Mtry is a splitting variable that denotes the number of predictors sampled to split at every node. Usually, the mtry default value for classification is √P, where P is the number of predictors. Bagging is obtained as a special case of a random forest when the mtry=P. Accuracy does not change dramatically due to different mtry applied. Sometimes mtry=1 gives better performance on a dataset rather than a larger mtry [38]. In this paper, for the random forest, an mtry of six was selected as it gave the highest overall accuracy compared to other mtry values. Three-fold, five-fold and six-fold cross-validations were used with the mtry value. Accuracy without SMOTE was 87.97% for three-fold and 88.19% for both five-fold and six-fold cross-validations on the testing dataset. After SMOTE implementation, results improved further but produced constant accuracies at 97.17% for all the cross-validation combinations. Table 3.2 shows the accuracy matrix after SMOTE implementation for six-fold cross-validation. The following results demonstrate that bagging algorithms also produce better results when SMOTE is used. Another contributing factor to this increase in accuracy can be owed to the ensemble learning approach that both XGBoost and the random forest takes. Since for random forest the

results obtained is an aggregation of multiple decision trees, SMOTE makes the decision boundaries much more visible and by doing so it increased the Gini index, making splits of high purity. Figure 3.2 shows the performance of random forest for without SMOTE and when SMOTE applied on the dataset.

Table 3.3: Confusion matrix of Random Forest

		Reference					
Algorithms	Prediction	1	2	3	Total	Sensitivity	Specificity
Randomforest	1	409	0	9	418	.972	.977
	2	4	55	1	60	.982	.993
	3	8	1	325	334	.970	.981
	Total	421	56	335	812		

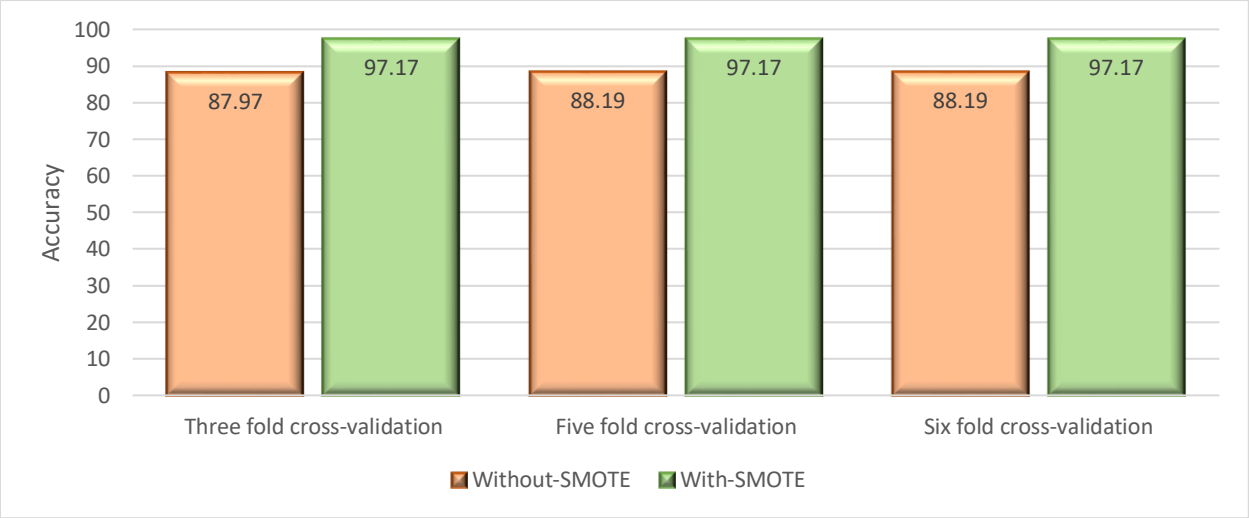

Fig 3.2: Accuracy estimates of Random Forest

3.8. Conclusion

Imbalanced data is one of the most commonly faced issue to hamper the prediction performance of a machine learning algorithm. In the cybersecurity domain this issue persists more often since historically the threats are much lower than the normal traffic. In this paper, we discussed how different machine learning algorithms behave when dealing with the imbalanced dataset and SMOTE. A standard Phishing dataset was used with class labels of -1,0 and 1

corresponding to phishy, suspicious and legitimate, respectively to test a different approach. Experiments conducted showed much lower accuracy when SMOTE is not applied on the dataset. Previous research on this dataset established effective prediction roadmaps using data processing and manipulation. Due to heavily preprocessing of datasets, many records were removed as outlier and inconsistent and this led to less credible models with less prediction performance. Most cybersecurity datasets are imbalanced with legitimate records overshadowing any phishy labels. Since these classes have significant differences in certain attributes, using SMOTE can create clusters amongst the labels, which makes them separable and equally significant. Using SMOTE on cybersecurity data mining is highly advisable due to its oversampling technique to optimize data loss. Less data loss tends to build robust models for reproduction. Using SMOTE on the phishing dataset resulted in an outcome of more than 9.5% increment in the accuracy and established an unprecedented example. This study shows that the application of SMOTE with the Random Forest algorithm results better prediction performance than other prominent machine learning algorithms. Tree-based bagging algorithms are proven effective in cybersecurity as well as other classification problems. This research shows that, the performance of random forest increases for imbalanced dataset after the usage of SMOTE. Further work includes investigating the relationship of different SVM kernels to SMOTE application. Other over-sampling techniques such as random-over sampling will also be done to ensure that all over-sampling methods perform uniformly on phishing datasets.

3.9. Reference

1. Jang-Jaccard, J. and Nepal, S., 2014. A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, *80*(5), pp.973-993.

2. Choo, K.K.R., 2011. The cyber threat landscape: Challenges and future research directions. *Computers & security*, *30*(8), pp.719-731.

3. Mohammad, R.M., Thabtah, F. and McCluskey, L., 2014. Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications*, *25*(2), pp.443-458.

4. Mohammad, R.M., Thabtah, F. and McCluskey, L., 2014. Intelligent rule-based phishing websites classification. *IET Information Security*, *8*(3), pp.153-160.

5. Abdelhamid, N., Ayesh, A. and Thabtah, F., 2014. Phishing detection based associative classification data mining. *Expert Systems with Applications*, *41*(13), pp.5948-5959.

6. Chandrasekaran, M., Narayanan, K. and Upadhyaya, S., 2006, June. Phishing email detection based on structural properties. In *NYS cyber security conference* (Vol. 3).

7. Sanglerdsinlapachai, N. and Rungsawang, A., 2010, January. Using domain top-page similarity feature in machine learning-based web phishing detection. In *2010 Third International Conference on Knowledge Discovery and Data Mining* (pp. 187-190). IEEE.

8. Eshmawi, A. and Nair, S., 2014, September. Semi-Synthetic Data for Enhanced SMS Spam Detection: [Using Synthetic Minority Oversampling TEchnique SMOTE]. In *Proceedings of the 6th International Conference on Management of Emergent Digital EcoSystems* (pp. 206-212).

9. Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, pp.321-357.

10. Bunkhumpornpat, C., Sinapiromsaran, K. and Lursinsap, C., 2009, April. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 475-482). Springer, Berlin, Heidelberg.

11. Johnson, B.A. and Iizuka, K., 2016. Integrating OpenStreetMap crowdsourced data and Landsat time-series imagery for rapid land use/land cover (LULC) mapping: Case study of the Laguna de Bay area of the Philippines. *Applied Geography*, *67*, pp.140-149.

12. Akbani, R., Kwek, S. and Japkowicz, N., 2004, September. Applying support vector machines to imbalanced datasets. In *European conference on machine learning* (pp. 39-50). Springer, Berlin, Heidelberg.

13. Perols, J., 2011. Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Auditing: A Journal of Practice & Theory*, *30*(2), pp.19-50.

14. Phua, C., Alahakoon, D. and Lee, V., 2004. Minority report in fraud detection: classification of skewed data. *Acm sigkdd explorations newsletter*, *6*(1), pp.50-59.

15. Padmaja, T.M., Dhulipalla, N., Bapi, R.S. and Krishna, P.R., 2007, December. Unbalanced data classification using extreme outlier elimination and sampling techniques

for fraud detection. In *15th International Conference on Advanced Computing and Communications (ADCOM 2007)* (pp. 511-516). IEEE.

16. Dittman, D.J., Khoshgoftaar, T.M., Wald, R. and Napolitano, A., 2014, May. Comparison of data sampling approaches for imbalanced bioinformatics data. In *The twenty-seventh international FLAIRS conference*.

17. Dang, X.T., Hirose, O., Bui, D.H., Saethang, T., Tran, V.A., Nguyen, L.A.T., Le, T.K.T., Kubo, M., Yamada, Y. and Satou, K., 2013. A novel over-sampling method and its application to cancer classification from gene expression data. *Chem-Bio Informatics Journal*, *13*, pp.19-29.

18. Dodge Jr, R.C., Carver, C. and Ferguson, A.J., 2007. Phishing for user security awareness. *computers & security*, *26*(1), pp.73-80.

19. Bowen, B.M., Devarajan, R. and Stolfo, S., 2011, November. Measuring the human factor of cyber security. In *2011 IEEE International Conference on Technologies for Homeland Security (HST)* (pp. 230-235). IEEE.

20. Aburrous, M., Hossain, M.A., Dahal, K. and Thabtah, F., 2010, April. Predicting phishing websites using classification mining techniques with experimental case studies. In *2010 Seventh International Conference on Information Technology: New Generations* (pp. 176-181). IEEE.

21. Xiang, G., Hong, J., Rose, C.P. and Cranor, L., 2011. Cantina+ a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)*, *14*(2), pp.1-28.

22. Zhang, Y., Hong, J.I. and Cranor, L.F., 2007, May. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on World Wide Web* (pp. 639-648).

23. Song, M. and Brook Wu, Y.F. eds., 2008. *Handbook of research on text and web mining technologies*. IGI global.

24. Pan, Y. and Ding, X., 2006, December. Anomaly based web phishing page detection. In *2006 22nd Annual Computer Security Applications Conference (ACSAC'06)* (pp. 381-392). IEEE.

25. Cortes, C. and Vapnik, V., 1995. Support-vector networks Machine learning (pp. 237–297), Vol. 20. *Boston, MA: Kluwer Academic Publisher*.

26. Fette, I., Sadeh, N. and Tomasic, A., 2007, May. Learning to detect phishing emails. In *Proceedings of the 16th international conference on World Wide Web* (pp. 649-656).

27. Miyamoto, D., Hazeyama, H. and Kadobayashi, Y., 2008, November. An evaluation of machine learning-based methods for detection of phishing sites. In *International*

Conference on Neural Information Processing (pp. 539-546). Springer, Berlin, Heidelberg.

28. Wang, J., Xu, M., Wang, H. and Zhang, J., 2006, November. Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding. In *2006 8th international Conference on Signal Processing* (Vol. 3). IEEE.

29. Deepa, T. and Punithavalli, M., 2011, April. An E-SMOTE technique for feature selection in high-dimensional imbalanced dataset. In *2011 3rd International Conference on Electronics Computer Technology* (Vol. 2, pp. 322-324). IEEE.

30. Mohammad, R.M., Thabtah, F. and McCluskey, L., 2012, December. An assessment of features related to phishing websites using an automated technique. In *2012 International Conference for Internet Technology and Secured Transactions* (pp. 492-497). IEEE.

31. Basnet, R., Mukkamala, S. and Sung, A.H., 2008. Detection of phishing attacks: A machine learning approach. In *Soft computing applications in industry* (pp. 373-383). Springer, Berlin, Heidelberg.

32. Liu, Y. and Zheng, Y.F., 2005, July. One-against-all multi-class SVM classification using reliability measures. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* (Vol. 2, pp. 849-854). IEEE.

33. Ho, T.K., 1995, August. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (Vol. 1, pp. 278-282). IEEE.

34. Ho, T.K., 1998. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, *20*(8), pp.832-844.

35. Amit, Y. and Geman, D., 1997. Shape quantization and recognition with randomized trees. *Neural computation*, *9*(7), pp.1545-1588.

36. Hastie, T., Tibshirani, R. and Friedman, J., 2009. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

37. Chen, T. and Guestrin, C., 2016, August. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

38. Liaw, A. and Wiener, M., 2002. Classification and regression by randomForest. *R news*, *2*(3), pp.18-22.

39. Ahsan, M., Gomes, R. and Denton, A., 2018, May. Smote implementation on phishing data to enhance cybersecurity. In *2018 IEEE International Conference on Electro/Information Technology (EIT)* (pp. 0531-0536). IEEE.

4. CONVOLUTIONAL NEURAL NETWORKS WITH LSTM FOR INTRUSION DETECTION[2]

4.1. Abstract

A variety of attacks are always attempted to network infrastructure. With the increasing development of artificial intelligence algorithms, it has become effective in preventing network intrusion for the last couple of decades. Besides accuracy and precision, false negative is an important performance metric to determine the usability of machine learning models in the cybersecurity domain. Deep learning methods are proven effective to achieve high accuracy with low false negatives to detect network intrusions. A conventional neural network architecture differs according to different domain dataset. Hence, achieving lower false alarm with an increment of accuracy using deep learning algorithm is one of the most critical research area, A novel approach using a hybrid algorithm of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) is introduced in this paper to provide a better network intrusion detection by lowering the false alarm. This bidirectional algorithm showed the highest accuracy of 99.70% on a standard dataset known as NSL KDD. The performance of this algorithm is measured using precision, false positive, F1 score, and recall, which are found promising for deployment on live network infrastructure.

4.2. Introduction

Competing with the recent colossal growth of computer networks and internet usage, network intrusion has also become a crucial issue in the current decade. A question is frequently arising from the security advocates, why should we bother detecting intrusions if we have

[2] The material in this chapter is co-authored by Mostofa Ahsan,and Dr. Kendall E. Nygard. Mostofa Ahsan had primary responsibility for all the experiments and development of conclusions. Dr. Kendall E. Nygard served as proofreader. Mostofa Ahsan also drafted and revised all version of this chapter [41].

already installed firewalls, operating system patches, and encrypted passwords for soundness? The answer to this question is simple: because intrusion still occurs. Just as people forget to lock their personal computers sometimes, they also forget to update the firewall, verify deeply before opening an unwanted email. Even with the 100% most advanced protection applied on a network or personal computer, it is not safe [27, 29].

It is described by Heady et al. [1], "An intrusion is a set of actions that attempt to compromise the confidentiality, integrity, or availability of information resources." The unauthorized malicious users are always trying to breach the vulnerable network architecture by attempting to a break-in, attack by penetration or getting authentication of different types of authentication, etc. There is no dispute of the fact that the growth of security intrusion is increasing so much competitively with the increment of the digital lifestyle. People are dependent on internet service and inherently increasing their usage daily. Since the security breach can affect their lifestyle vastly, it is very crucial to develop precautionary measures to safeguard the interest of users from various categories of attacks it is susceptible to [2, 35].

The system design which is used to detect malicious actions in a network is termed a Network Intrusion Detection System (NIDS) [3,2]. It has two categories, namely, Signature-based Network Intrusion Detection System (SNIDS) and Anomaly Detection based Network Intrusion Detection System (ADNIDS). SNIDS can detect unauthorized access or intrusion by matching patterns on the features it is trained for. And ADNIDS detects any anomaly when there is a deviation in the normal traffic pattern [2]. Since ADNIDS is highly pruned to a false alarm, SNIDS is proven the best approach to Detect Network Intrusions. Different artificial intelligence approaches are the key factor used in SNIDS. Since machine learning techniques are proven efficient to detect patterns from historical data, they have been employed to develop NIDS for

anomaly detection. But, the drawback of machine learning approaches is that based on different features, training, and test dataset, the prediction results differ in various scenarios. Compared to them, deep learning methods are promising as they are robust and efficient for a large number of features [2].

Deep learning is a branch of machine learning which can achieve outstanding performances [4]. Deep learning techniques are most prominent for getting efficient results from big datasets. Deep learning methods can learn automatically from raw data and then output results by operating end-to-end fashion and very practical to develop on an existing system. Previously, many deep learning approaches have been proven better for NIDS. The state of art Neural Networks is more promising than the previous ones because they are agent-based. Since NIDS still suffers from high false alarms and the historical dataset is still uprising, deep learning implementation to fight intrusion is the most efficient solution.

4.3. Related work

The trend of Intrusion Detection (ID) was first proposed by Anderson in 1980. He developed an anomaly-based ID process for network monitoring to detect abnormal activity [5]. After that, a lot of progress made in the field of ID by using different Artificial Intelligence algorithms. They were promising for small networks where the varieties of usage were limited, and the data stream was not so large to process. Since the growth of network usage has significantly increased, and the scenarios are more complex than the early stage, IDs are needed to cope up with the changes. So, machine learning techniques played a major role in classifying intrusions from authenticity. A modified support vector machine (SVM) combined with kernel principal component analysis (KPCA) and Genetic algorithm (GA) showed efficient results in 2014 [6]. GA is vastly used for improved efficiency, which is based on genetic principles and

certain environmental factors. But, in the training process of GA, some fixed rules are implemented from the analyzed data of the algorithm, and the data are tabulated in the form of a large number of rules which could be used for monitoring NIDS [9, 36, 37]. A novel scheme using Principal Component Analysis (PCA) identifying anomalies as outliers were proposed by Shyu et al. [8] in 2003. Since the anomalous data is highly susceptible to outliers, we can train the datasets using random oversampling or Synthetic Minority Oversampling Technique (SMOTE) is effective in classifying malicious phishing emails [18,21]. Clustering is a process of creating a partition of data in a way that each group represents similar characteristics. By finding the same pattern, the data is segregated. Since clustering can learn from the record and audit the data itself, it has a significant benefit for Intrusion Detection [9]. Mini batch K-means clustering produced very good accuracy by using the K-means principal idea of allocating different random groups of distinct memory sizes, which facilitates the easiest process to store [10]. Since it takes different batches, it is a little bit time-consuming, which impedes user usage.

For continuous streaming data classification techniques play a major role in anomaly detection. To enhance user experience and fast network stream, Li et al. proposed a K-Nearest Neighbor (KNN) classification in the wireless sensor network [7, 38, 21]. Various machine-learning algorithms such as decision tree, rule-based induction, Bayesian network, genetic algorithm have a significant impact on enhancing network security. Nowadays, ensemble learning is being used under classification techniques that have optimized false alarms. The classifier of Ensemble Accuracy (AUE) is a modified version of Accuracy Weighted Ensemble (AWE), which uses the concept of updating a classifier according to the distribution [11].

Practically, traditional machine learning models, like the support vector machine (SVM) and k-nearest neighbor (KNN), contain none or only one hidden layer. So, these traditional

machine learning models are also called shallow models [13]. Deep learning methods integrate high-level feature extraction and classification tasks which overcome most of the limitations of shallow learning and further promote the progress of intrusions detection systems [12]. Deep learning methods can automatically extract features and perform classification on the dataset, such as Auto Encoder, deep belief network (DBN), deep neural network, and recurrent neural network (RNN) [14,15]. Previously many deep learning approaches are proven effective for NSL KDD datasets [2, 4, 5, 6, 9, 11, 12, 13, 14, 15, 17, 18, 19, 22]. Stacked autoencoders were used in IEEE 802.11 networks platform to detect intrusion, which had an accuracy of 98.60% [16]. Ma et al. [17] proved a hybrid method that combined spectral clustering and deep neural network for intrusion detection on the NSL KDD dataset, which achieved an accuracy of 72.64%. The gated recurrent unit (GRU) recurrent neural network (RNN) combination used as (GRU-RNN) was presented to detect intrusion over a software-defined network (SDN) had an overall accuracy of 89% [19]. A hybrid of the stacked non-symmetric autoencoder and the random forest was used for NIDS by Shone et al. [20]. Muna et al. [22] used a deep autoencoder for feature extraction and feedforward neural networks for classification for intrusion detection. Restricted Boltzmann machine (RBM) is also proven effective in classifying normal and anomalous network traffic [23].

Above mentioned deep learning approaches are promising and effective, but still there are detection error, such as a low detection rate for unprecedented attacks and high false-positive rate for minority attacks. To overcome these classification issues, this paper is going to use a novel technique that makes a hybrid of Convolutional Neural Network (CNN) and Long Short Term Memory neural network (LSTM) to improve the detection rate of unknown attacks along with low false-positive the rate for minority attacks.

4.4. Dataset

Several researches were committed on the KDD Cup 1999 dataset using machine learning techniques. But this dataset was had various disadvantages, such as redundant records. The training dataset had 78% redundancy, whereas testing had 75% duplicate records. As a result, most of the prediction was biased [9]. Since the availability of the public data set of network intrusion systems is limited, a new version of this dataset, also known as NSL KDD, is used by the researchers. The newer version combines some original data from the previous version, and the redundancy of records does not exist anymore. The datasets are made of basically four types of attack classes [24,25]. The categorical attack classes are described in table 4.1.

Table 4.1: Attack categories and their description.

Name of the attack	Description
Denial of Service (DoS)	Denial of Service is an attack category that depletes the victim's resources and reduces the ability to handle legitimate requests – e.g. syn flooding. Relevant features: "Percentage of packets with error", "source bytes" [18,24,25]. Frequency in training dataset: 45,927 & in testing dataset: 7,458.
Probe attack (probe)	Surveillance and another probing attack whose objectives are to collect information about the victim, remotely – e.g. port scanning. Relevant features: "source bytes", "duration of connection" [9, 24, 25]. Frequency in training dataset: 11,656 & in testing dataset: 2,421.
User to Root (U2R)	Unauthorized access to local root user privileges an attack type that is used by an attacker to log into the system as a local user and get administrator access by exploiting some vulnerability in the victim's system – e.g. buffer autoflow attacks. Relevant features: "number of shell prompts invoked", "number of file creation" [9, 18, 24, 25]. Frequency in training dataset: 52 & in the testing dataset: 200.
Root to Local (R2L)	Unauthorized access from a remote system as an administrator. Then the attacker intrudes into a remote machine and get access to the victim's local machine – e.g. password guessing. Relevant features: Host level features: "number of failed login attempts" and network-level features: "service requested", "duration of connections" [9, 24, 25]. Frequency in training dataset: 995 & in testing dataset: 2,754.

In each rows 41 features are unfolding different attributes of the flow and label assigned to each other as normal or an attack type. These features can be primarily classified into four categories [26]. Such as, Basic features which are the attributes of individual TCP connection. Redundant attributes are duration, protocol_type, service, src_bytes, dst_bytes, flag, land, wrong_fragment, urgent.

Content features are the attributes, which suggests the domain knowledge within a connection. Redundant attributes are: hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creation, num_shells, num_access_files, num_outbound_cmds, is_hot_login, is_guest_login.

Traffic features are the attributes which are calculated using only two-seconds window time. Redundant features are: count, serror_rate, rerror_rate, same_srv_rate, diff_srv_rate, srv_count, srv_serror_rate, srv_rerror_rate, srv_diff_host_rate.

Host features are the attributes which are designed to attack and access in more than two seconds: Redundant features are: dst_host count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate,dst_host_rerror_rate, dst_host_srv_rerror_rate [26].

4.5. Algorithms used

A different machine learning algorithm is proven promising to predict intrusion on NSL KDD datasets before. But since shallow learning has a high false-positive rate, this paper only addresses deep learning methods, which are a sub-field of machine learning that advances shallow learning close to artificial intelligence. Deep learning facilitates the modeling of complex relationships and concepts using multilevel representations [20]. This paper we are

going to compare five well-established deep learning algorithms with our approach. We have selected at least one of the Modular Neural Network (MNN), Artificial Neural Network (ANN), Feed Forward Neural Network, Auto Encoder (AE), and Recurrent Neural Network (RNN).

4.5.1. DenseNet (Densely Connected Networks)

Residual Network (ResNet) significantly changed the view of the parametrization of the functions in deep learning. DenseNet is the logical extension of ResNet. Recently researchers have tried to solve the problem of the vanishing gradient of ResNet cause; this method combines features through summation to pass it to the next layer. DenseNet uses not to connect through summation but in a feed-forward fashion. In denseNet, each layer has direct access to the gradient from loss function and the original input signal, which leads to an improved flow of information and gradient throughout the whole network. Moreover, since it has a regularization effect, which reduces overfitting on tasks with similar training set sizes. The most important difference with other deep learning methods is, DenseNet has very narrow layers – e.g. k=8 which refers to the hyperparameter k as the growth rate of the neural network. We have used Rectified Linear Unit (Relu) for the first three layers and the Softmax function for the activation layer for our experiment. As scalar, we can write DenseNet as following:

$$f(x)=f(0)+f'(x)x+12f''(x)x2+16f'''(x)x3+o(x3)$$

4.5.2. CNN (Convolutional Neural Network)

CNN, also known as ConvNet, is a deep learning algorithm that is mostly used for image classification by assigning various aspects or objects in the image and enable to differentiate one from another. The architecture of CNN resembled to the connectivity pattern of neurons in the human brain and was inspired by the organization named Visual Cortex. It consists of several steps for classification of the dataset, such as Convolution, max-pooling, full connection, fully

connected-Relu etc. The convolution plays a significant role in feature extraction and resizing data after multiple steps.

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau$$

4.5.3. GRU (Gated Recurrent Units)

As a solution to the vanishing gradient problem pf standard RNN, GRU uses an update gate function to update and reset gate for nonlinear output. Since in this paper we experimented with multiclass prediction, we choose to use sigmoid as activation function along with GRU for better accuracy. The gate function plays a vital role in updating how much of the past information needs to be passed to the next layer. The update and reset equation of the gate is described below where W(z) is the network's own weight, which is multiplied by x_t when it is plugged into the network unit. The same process applied for h_(t-1), which holds the information for the previous t-1 and multiplied by its own weights U(z).

The update function is: $z_t = \sigma(W^{(z)} x_t + U^{(z)} h_{t-1})$

The reset function is: $r_t = \sigma(W^{(r)} x_t + U^{(r)} h_{t-1})$

4.5.4. Bi-LSTM (Bidirectional Long Short-Term Memory)

In a traditional Neural Network, all the input and outputs are independent of each other. But, in case of predicting the next elements in the series or word in the sentence, the previous features or elements needed to remember for predicting the future element. RNN creates a loop that help to persist these types of information. Bidirectional RNN usually puts together two independent RNN that enable to run input in two directions like past to future and future to past. Bidirectional LSTM also acts as Bidirectional RNN by preserving both historical and prediction results [28].

4.5.5. AE (Autoencoder)

AEs are a specific type of feedforward neural network where the size of the input is the same as the size of the output. AE compresses the input into a lower-dimensional code and then again reconstructs the output back from the representation. It consists of three major components: encoder, code, and decoder. AE is used for mostly unsupervised learning because it doesn't need explicit labels to train on. More specifically, we can call them self-supervised since they generate their own labels from the training data set. The encoder and decoder functions are described as: encoder function is denoted by ϕ, maps the original data Xto a latent space F which is situated at the bottleneck. And the decoder function is denoted by ψ that maps the latent F at the bottleneck to the output.

$$\emptyset : x \rightarrow \mathcal{F}$$

$$\psi : \mathcal{F} \rightarrow x$$

$$\phi, \psi = \text{argmin } \|X - (\psi \, o \, \phi) \, X \|^2$$

4.5.5. Proposed hybrid of CNN and LSTM

Unlike traditional Convolutional Neural Network (CNN), RNN help to create an interaction between the input sequence, and hence this feature of RNN creates a new approach to feature hybrid [28,30]. Many researchers have proposed methods for hybridizing the features using LSTM (a variant of RNN), which can extract the long-term dependencies of the data features in the sequence to improve the recognition accuracy [28, 30, 31, 32, 33, 34, 40]. In this paper, we have proposed an innovative strategy by using multiple convolutional kernels to extract features from the dataset. Moreover, this proposed method establishes an end-to-end mapping of the relationship between the features and the attack types. Our approach consists of two stages, and the first part is feature extraction based on CNN and the feature fusion part based on LSTM in the later part. In the first stage, the forward propagation process is applied as: it

assumes the l layer is a convolution layer and the l-1 layer is a pooling layer or another input layer for the next extraction process. The equation behind the first layer is:

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} \times k_{ij}^l + b_j^l)$$

The of the above equation denotes the jth feature image of the l layer. The latter part shows the convolution operation and summation for all feature maps of the l −1 layer and the jth convolutional kernel of the l the layer, and then add an offset parameter and then passes the activation function f(*). Among them, l is the number of layers, f is the activation function, is an input feature map of the upper layer, b is an offset, and k is the convolutional kernel. For downsampling, assume the l layer as the pooling layer and l -1 is the convolutional layer. The formula is described below:

$$x_j^l = f\left(\beta_j^l \, down\left(x_j^{l-1}\right) + b_j^l\right)$$

In the feature extraction stages, we have used Relu functions for both convolutions a pooling layer. For the first convolution and pooling layer, there were 48 convolutional kernels with 3x3 kernel sizes. After the max-pooling, we again used 16 convolutional kernels and 3x3 kernel sizes. For both times, we use pooling length as 2. We set the output size of the LSTM part as 70. Finally, the classification results of attack types are obtained through the Softmax function. We used Adam optimizer for better stochastic gradient descent. To prevent overfitting issue, we used dropout as .01. The hybrid approach structure is showed in Fig 4.1.

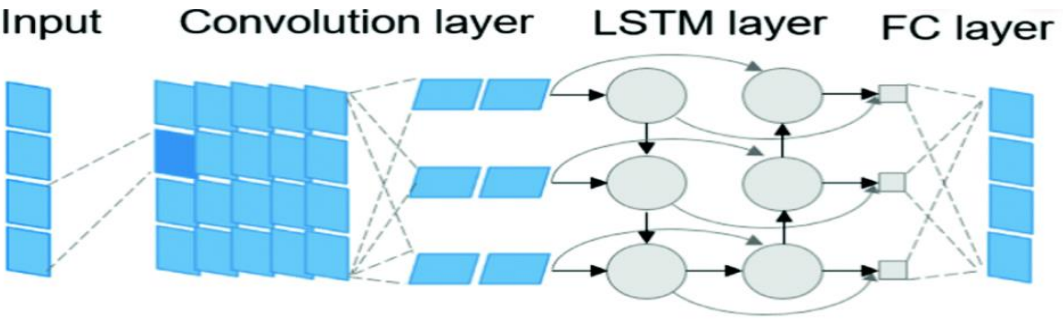

Fig 4.1: Hybrid of CNN and LSTM architecture.

4.6. Experiments and results

For deep learning methods, preprocessing of data always plays a major role. The first challenge was to convert the class labels into four different attack types. We segmented all the raw data into five categories, including normal. Then we randomly selected ten percent of the training dataset and five thousand testing samples. Then, the data was normalized and preprocessed in scalar format to feed the neural networks as input.

The Autoencoder produced a very low accuracy of 37.65% without any hyperparameter tuning. For experimental standards, we set the epoch size to every method to 100. We have observed that DenseNet was able to produce 94.98% accuracy with only 20 epochs. Bidirectional LSTM was very much close to DenseNet. It achieved the highest accuracy of 97.32%. For CNN, we used a filter size of 16x16 and 50% dropout, which achieved an accuracy of 95.72% accuracy. The Gated Recurrent Unit (GRU) with a softmax activation function achieved 97.36% accuracy. The hybrid of CNN and LSTM is considered a bidirectional approach and able to outperform all other algorithms by achieving the highest 99.70% accuracy, according to figure 4.2. For the convolutional layer, we used Relu function, and for the activation function, we used softmax for 100 epochs. The Receiver Operating Characteristics (ROC) curve below has plotted the data of the algorithms those have achieved the most accuracy. We have run several combinations of kernel sizes and pooling lengths to obtain the best result.

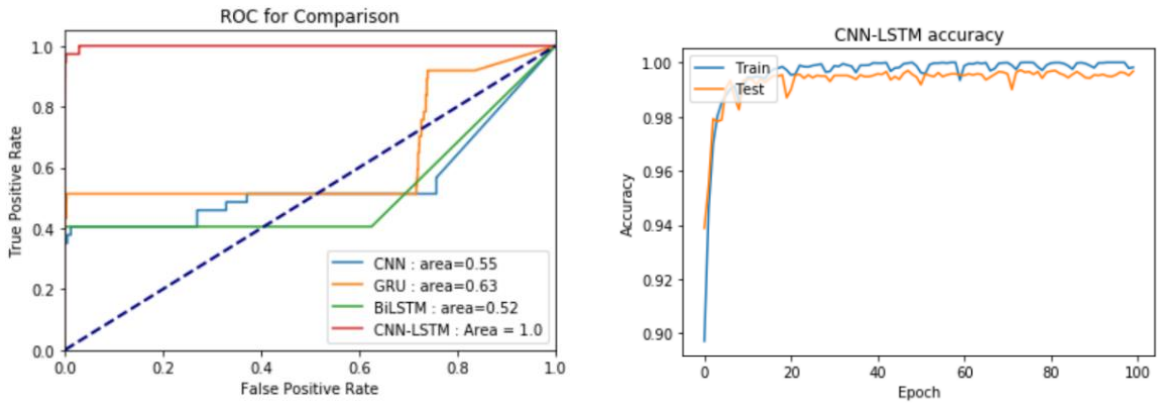

Fig 4.2: Top four algorithms ROC curve (left) and hybrid algorithm accuracy (right)

We can observe the false positive for Probe attack is high. But, other than that, every class label is predicted near perfect. The overall f1 score is promising for all the attack types as plotted on figure 4.3.

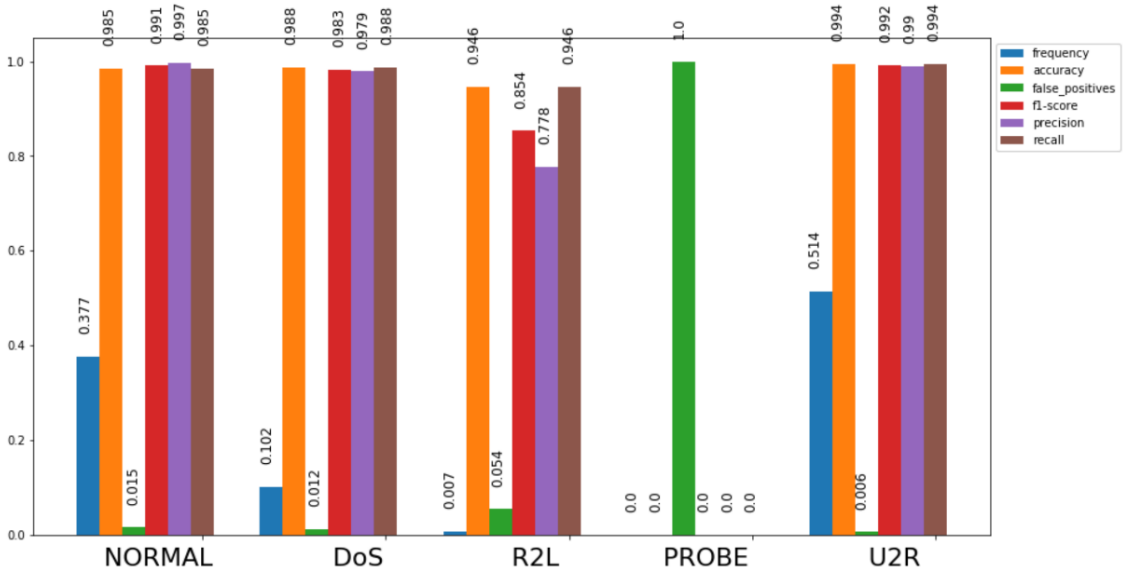

Fig 4.3: Individual class label result analysis.

There was a consistent increment in the accuracy of the training dataset from 93.38% in epoch 1 to 99.70 % in epoch 78. However, we observe that there is a slight decrease in testing accuracy after epoch 80. The ROC curve also shows a higher accuracy of 99 % as in the figure 4.3.

4.7. Conclusion and future work

In this paper, we have successfully established that a hybrid of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) is a very effective way for network intrusion detection. We have achieved an unprecedented higher accuracy on a standard NSL KDD dataset without applying any hyperparameter tuning. It is conspicuous from this research that deep learning methods are the most promising and effective for anomaly detection and intrusion prevention. But, datasets from different domain often does not produce the best model using standard deep learning architecture. Based on the problem statements, neural networks needed to be modified to enhance predictive performance. In this study, our primary goal was to reduce the false negatives for the intrusion detection system. Better accuracy does not always reduce the false alarm in cybersecurity. Our proposed hybrid of two neural network architecture not only reduce the false negatives significantly but also increased the accuracy to 99% which is unprecedented for the NSL KDD dataset. Due to the security issues, it is very scarce to find standard network traffic data from established servers. NSL KDD is one of the publicly available datasets which is standardized with state of the art threats for a long period of time. Enhancing the predictive performance of this dataset also indicates its reproducibility and deployment credibility on a different network. We have future research plans to experiment with different hyperparameter of this hybrid method to achieve better accuracy for other available publicly available cybersecurity datasets. In the future, we will implement this algorithm on a live network and observe the performance from different perspectives to make it more robust and reproducible. The performance comparison with other well-established intrusion detection systems will increase the credibility of deployment in different networks.

4.8. References

1. Heady, R., Luger, G., Maccabe, A. and Servilla, M., 1990. *The architecture of a network level intrusion detection system* (No. LA-SUB-93-219). Los Alamos National Lab., NM (United States); New Mexico Univ., Albuquerque, NM (United States). Dept. of Computer Science.

2. Gurung, S., Ghose, M.K. and Subedi, A., 2019. Deep learning approach on network intrusion detection system using NSL-KDD dataset. *International Journal of Computer Network and Information Security*, *11*(3), pp.8-14.

3. Forouzan, B.A. and Mukhopadhyay, D., 2011. *Cryptography and network security (Sie)*. McGraw-Hill Education.

4. Yin, C., Zhu, Y., Fei, J. and He, X., 2017. A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, *5*, pp.21954-21961.

5. Javaid, A., Niyaz, Q., Sun, W. and Alam, M., 2016, May. A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)* (pp. 21-26).

6. Kuang, F., Xu, W. and Zhang, S., 2014. A novel hybrid KPCA and SVM with GA model for intrusion detection. *Applied Soft Computing*, *18*, pp.178-184.

7. Li, W., Yi, P., Wu, Y., Pan, L. and Li, J., 2014. A new intrusion detection system based on KNN classification algorithm in wireless sensor network. *Journal of Electrical and Computer Engineering*, *2014*.

8. Shyu, M.L., Chen, S.C., Sarinnapakorn, K. and Chang, L., 2003. *A novel anomaly detection scheme based on principal component classifier*. MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING.

9. Seraphim, B.I., Palit, S., Srivastava, K. and Poovammal, E., 2018, December. A Survey on Machine Learning Techniques in Network Intrusion Detection System. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)* (pp. 1-5). IEEE.

10. Peng, K., Leung, V.C. and Huang, Q., 2018. Clustering approach based on mini batch kmeans for intrusion detection system over big data. *IEEE Access*, *6*, pp.11897-11906.

11. Ahmad, I., Basheri, M., Iqbal, M.J. and Rahim, A., 2018. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE access*, *6*, pp.33789-33795.

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

13. Ding, Y. and Zhai, Y., 2018, December. Intrusion detection system for NSL-KDD dataset using convolutional neural networks. In *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence* (pp. 81-85).

14. Denton, A.M., Ahsan, M., Franzen, D. and Nowatzki, J., 2016, December. Multi-scalar analysis of geospatial agricultural data for sustainability. In *2016 IEEE International Conference on Big Data (Big Data)* (pp. 2139-2146). IEEE.

15. Ahsan, M., Gomes, R. and Denton, A., 2019, May. Application of a Convolutional Neural Network using transfer learning for tuberculosis detection. In *2019 IEEE International Conference on Electro Information Technology (EIT)* (pp. 427-433). IEEE.

16. Thing, V.L., 2017, March. IEEE 802.11 network anomaly detection and attack classification: A deep learning approach. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)* (pp. 1-6). IEEE.

17. Ma, T., Wang, F., Cheng, J., Yu, Y. and Chen, X., 2016. A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. *Sensors*, *16*(10), p.1701.

18. Ahsan, M., Gomes, R. and Denton, A., 2018, May. Smote implementation on phishing data to enhance cybersecurity. In *2018 IEEE International Conference on Electro/Information Technology (EIT)* (pp. 0531-0536). IEEE.

19. Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R. and Ghogho, M., 2018, June. Deep recurrent neural network for intrusion detection in sdn-based networks. In *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)* (pp. 202-206). IEEE.

20. Shone, N., Ngoc, T.N., Phai, V.D. and Shi, Q., 2018. A deep learning approach to network intrusion detection. *IEEE transactions on emerging topics in computational intelligence*, *2*(1), pp.41-50.

21. Gomes, R., Ahsan, M. and Denton, A., Fusion of SMOTE and Outlier Detection Techniques for Land-Cover Classification Using Support Vector Machines.

22. Muna, A.H., Moustafa, N. and Sitnikova, E., 2018. Identification of malicious activities in industrial internet of things based on deep learning models. *Journal of information security and applications*, *41*, pp.1-11.

23. Aldwairi, T., Perera, D. and Novotny, M.A., 2018. An evaluation of the performance of Restricted Boltzmann Machines as a model for anomaly network intrusion detection. *Computer Networks*, *144*, pp.111-119.

24. Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A.A., 2009, July. A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1-6). IEEE.

25. Dhanabal, L. and Shantharajah, S.P., 2015. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, *4*(6), pp.446-452.

26. Aggarwal, P. and Sharma, S.K., 2015. Analysis of KDD dataset attributes-class wise for intrusion detection. *Procedia Computer Science*, *57*, pp.842-851.

27. Chowdhury, M.M. and Nygard, K.E., 2017, May. Deception in cyberspace: An empirical study on a con man attack. In *2017 IEEE International Conference on Electro Information Technology (EIT)* (pp. 410-415). IEEE.

28. Wang, J., Zhang, J. and Wang, X., 2017. Bilateral LSTM: A two-dimensional long short-term memory model with multiply memory units for short-term cycle time forecasting in re-entrant manufacturing systems. *IEEE Transactions on Industrial Informatics*, *14*(2), pp.748-758.

29. Chowdhury, M.M., Tang, J. and Nygard, K.E., 2013. An artificial immune system heuristic in a smart grid. In *the 28th International Conference on Computers and Their Applications*.

30. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R. and Schmidhuber, J., 2016. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, *28*(10), pp.2222-2232.

31. Tsironi, E., Barros, P., Weber, C. and Wermter, S., 2017. An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition. *Neurocomputing*, *268*, pp.76-86.

32. Zhou, X., Hu, B., Chen, Q. and Wang, X., 2018. Recurrent convolutional neural network for answer selection in community question answering. *Neurocomputing*, *274*, pp.8-18.

33. Zhao, R., Yan, R., Wang, J. and Mao, K., 2017. Learning to monitor machine health with convolutional bi-directional LSTM networks. *Sensors*, *17*(2), p.273.

34. Nunez, J.C., Cabido, R., Pantrigo, J.J., Montemayor, A.S. and Velez, J.F., 2018. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition*, *76*, pp.80-94.

35. Chowdhury, M. and Nygard, K., 2018, March. Machine Learning within a Con Resistant Trust Model. In *The 33rd International Conference on Computers and their Applications (CATA 2018)*.

36. Chowdhury, M.M., Nygard, K.E., Kambhampaty, K. and Alruwaythi, M., 2017, December. Deception in cyberspace: Performance focused con resistant trust algorithm. In *2017 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 25-30). IEEE.

37. Chowdhury, M.M. and Nygard, K.E., 2017. An empirical study on a con resistant trust algorithm for cyberspace. In *Proceedings on the International Conference on Internet Computing (ICOMP)* (pp. 32-37). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

38. Gomes, R., Ahsan, M. and Denton, A., 2018, May. Random forest classifier in SDN framework for user-based indoor localization. In *2018 IEEE International Conference on Electro/Information Technology (EIT)* (pp. 0537-0542). IEEE.

39. Ahsan, M. and Nygard, K.E., 2020, March. Convolutional Neural Networks with LSTM for Intrusion Detection. In *CATA* (pp. 69-79).

40. Ahsan, M., Gomes, R., Chowdhury, M. and Nygard, K.E., 2021. Enhancing Machine Learning Prediction in Cybersecurity Using Dynamic Feature Selector. Journal of Cybersecurity and Privacy, 1(1), pp.199-218.

41. Ahsan, M. and Nygard, K.E., 2020, March. Convolutional Neural Networks with LSTM for Intrusion Detection. In CATA (pp. 69-79).

5. ENHANCING MACHINE LEARNING PREDICTION IN CYBERSECURITY USING DYNAMIC FEATURE SELECTOR

5.1. Abstract

The ability to learn and adapt has made the machine learning techniques as mainstream in cybersecurity. Training a model with a comprehensive dataset with multiple attack types is one of the keys to improve the anomaly detection performance. However, issues like high dimensional data possess a significant threat for most of the machine learning techniques allowing real-time response and deployment on legacy systems. Reducing feature size is the key solution to lower the computational complexity of the machine learning models. Removing the insignificant features and selecting the most important features is a tradeoff which can compromise the predictive performance without different statistical and AI-driven test. However, still there a chance that after the deployment, this model will not perform as good as it supposed to be. Dynamic Feature Selector (DFS) is an algorithm which can automate the feature selection process without reducing the predictive performance and updated using meta learner bagging ensemble approach which is proven effective during deployment of machine learning techniques in cybersecurity domain.[3]

5.2. Introduction

Machine learning techniques are becoming very efficient methods in intrusion detection system with their real time response and adaptive learning process. A robust model can be deployed for anomaly detection by using a comprehensive dataset with multiple attack types. However, processing high dimensional data has been a significant challenge for model

[3] The material in this chapter is co-authored by Mostofa Ahsan, Rahul Gomes, Md Minhaz Chowdhury and Kendall E. Nygard. Mostofa Ahsan had primary responsibility for all the experiments and development of conclusions. Rahul Gomes was responsible for editing and proof reading. Md Minhaz Chowdhury and Kendall E. Nygard served as a proof reader [69].

performance. Large scale data with redundant or insignificant features increases the computational time and often decreases goodness of fit which is a critical issue for domain like cybersecurity where time complexity is one of the major factors of real-life development. In this research we have proposed an efficient feature selection algorithm which filters insignificant variables with a stochastic process leveraging the optimized output of a meta learner. Our proposed Dynamic Feature Selector (DFS) uses different statistical analysis and feature importance test to reduce machine model complexity significantly and improve the prediction accuracy as well. We have used two standard datasets used for cybersecurity research namely NSL KDD and UNSW-NB15 to evaluate the performance of DFS. Both experiments have shown significant increase in accuracy, precision and F1 score of the machine learning model. NSL KDD showed an increment from 99.54% to 99.64% with reducing the feature size from 123 to 50 and UNSW-NB15 resulted an increment 90.98% to 92.46% with reducing the feature size from 196 to 47. The proposed approach is thus able to achieve higher accuracy while significantly lowering the number of features required for processing.

5.2. Previous work

Bagging is and ensemble learning method and is the process of generating multiple versions of a predictor by resampling the training data and later aggregating those predictors to get a stable predictor [1]. The training datasets bootstrap replica is generated, and each predictor is generated from each such training replicas. Bootstrap replica is produced by resampling each category. The one-hot encoding of "apple" is [1, 0, 0], "orange" is [0, 1, 0] and "berry" is [0, 0, 1]. This encoding can be written as a 3- dimensional feature vector [1, 0, 0], [0, 1, 0], [0, 0, 1].

This formatted data from the one-hot encoding method is fed into a machine learning algorithm for various diverse applications [7]. Examples of such applications are DNS sequencing [9], text representation as numerical matrix [10]. Most of the times, this encoding method is used as a data pre-processing step.

One-hot vectors can be used to represent DNA sequences as the unit value in the vector can represent the position of a DNA component called nucleotide [10]. Here, the authors also presented how a text can be represented as matrix using one-hot encoding method. Such matrix can be fed into a convolutional neural network for the text's classification.

One-hot encoding was used on the NSL-KDD dataset numerous times [11], [12], [13]. In [11], the NSL-KDD dataset's category features are transformed into numeric values by one-hot encoding method. The authors mentioned that the transformed integer values also indicated the importance the order of the features. In another research work, the categorical features from the NSL-KDD dataset was transformed into numerical values and named as "sub-categorical features" [12]. NSL-KDD's 41 features are transformed into 122 subcategories. In another work, three features from the NSL-KDD dataset (protocol type, flag, service) were transformed into numerical values [13].

In this paper, the features are mutually exclusive and hence one-hot encoding method can be suitable applied [14]. Also, the data used in this paper is free from dirty categories, which is a weakness of one-hot encoding [8].

A wrapper method can search for a good feature set [15]. The selected features goodness is evaluated, in this method, via a learning algorithm [15]. Other way of saying this is, wrapper method uses a search algorithm finding a subset of features. These features are than passed (by the wrapper method) to a predictive model that evaluates the subset [16]. The wrapper method is

superior to the other comparable methods, evaluating the goodness of the selected features [15], [16], [17], [18], [19]. This method can also be used to select the best subsets from the features. Wrapper method is also combined with other methods of feature selection and feature construction for an enhanced performance [18].

A Wrapper method-based feature selection approach has previously been tested on NSL-KDD dataset [20]. The authors claimed that they did not use KDD-CUP as it is the earlier version of NSL-KDD data set and NSL-KDD is superior to KDD-CUP dataset. The authors reduced the features present in this dataset by a percentage close to 60%. In our research work presented in this paper, one of the two datasets used is NSL-KDD dataset. The other dataset is UNSW-NB15.

There are many research works that used one of these two datasets rather both, NSL-KDD and UNSW-NB15 datasets [21], [22], [23], [24], [25]. The use of these both datasets are not so common and there are few which used them both [11], [26], [27], [28], [29], [30]. A partial list of where these two datasets and their predecessors were used was presented in [30].

In [26], the authors applied a Hybrid Filter-Wrapper feature selection method to detect distributed denial of service detection. During the application of this hybrid method, they claimed to reduce the number of features from 40 to 9 with a high accuracy of DDoS detection.

These two datasets have also been used to verify the performance of a network anomaly detector by analyzing network traffic [28]. For UNSW-NB15 dataset, the anomaly detector's training phase used 206138 records and the testing phase used 51535 records. For the NSL-KDD dataset, there were 118813 records for training and 29704 records for testing. The detector performed better on the UNSW-NB15 dataset, compared to the NSL-KDD dataset. The authors mentioned that the reason of such outperformance is, UNSW-NB15 has more records.

A deep belief network was used for cyber-attack detection using port scanning method where the model was tested using two security datasets UNSW-NB15 and NSL-KDD [29]. The proposed algorithm had high accuracy with low false alarm. The authors mentioned to use this algorithm with real time attack data. This proves that these datasets worked as a close candidate of real time attack data. In another research work, an intrusion detection system, based on random forest classifier, was developed, and tested using three datasets. Two of the datasets were UNSW-NB15 and NSL-KDD [30].

Feature selection was applied on KDD dataset, but mostly on the KDD99 version rather NSL-KDD [31]. For example, a wrapper method's variation was used for feature selection using both KDD99 and UNSW-NB15 dataset [27]. To be noted, the authors used KDD99 instead of NSL-KDD. Their proposed method reduced the number of features to 18 and 20 respectively for KDD99 and UNSW-NB15.

Feature reduction using principal component analysis (PCA) is conducted in several research on both NSL KDD and UNSW-NB15 datasets [60,61,62,63,64,65]. But the information loss due to PCA, those experiments failed to achieve better accuracy than other techniques like wrapper method and correlation elimination technique [62,65]. One of the key issues associated with PCA is that it assumes, the dataset is a linear combination of original features [66]. Hence, for nonlinear features, PCA does not produce sensible result. Moreover, PCA uses variance as a measure of how important an individual dimension is. As a result. High variance axes are considered as principal components and low variance axes are considered as noises [67]. The other issue associated with PCA is, the dataset needs to be standardized before performing PCA on it which is a better way to achieve higher accuracy, but often hinder the deployment process

due to its decompression technique. Because, after the application of PCA, the independent variables become less interpretable as original features [68].

5.4. Algorithms used for dimensionality reduction

Feature selection methods among other benefits contribute towards increasing classification accuracy [32]. It is helpful in reducing the number of irrelevant features that when included in the predictive model would increase computational complexity and training time but provide negligible or no increase in prediction accuracy [33]. A combination of methods for feature selection in this research has been discussed in this section.

5.4.1. Univariate feature selection

To select the most relevant features, univariate feature selection method utilizes univariate statistical tests to return a list of features which are ranked depending on the scoring function used. This can be an effective pre-processing step to retrieve the most significant features in a dataset that contribute to prediction accuracy significantly. To perform the feature selection process, a one-way ANOVA f-test was performed. Like Naïve Bayes Classifier, the one-way ANOVA (Analysis of variance) f-test ensures that there is no relationship between the feature attributes used to accurately classify the dependent attribute. There is higher degree of variance if the means obtained from groups of data is different from the global mean derived from the dataset. Thus, it successfully returns the ratio of the inter-group to intra-group variability in a sample. ANOVA was selected over the t-test to give more stability and reduce the type 1 error while comparing population means of multiple groups. Features with a score higher than a percentile value of 97 was considered useful in this analysis.

5.4.2. Correlated feature elimination

Another feature engineering method applied to reduce the dimensionality of the dataset was the elimination of highly correlated features. Consider two features a and b where a = x_1. . . .,x_n and b = y_1. . . .,y_n. The degree of similarity between these two features can be represented using a correlation coefficient. The Pearson Correlation coefficient was used to express the correlation between features. It was the most logical choice due to two reasons. The complexity of Pearson Correlation is linear making it efficient. Furthermore, the features on which Pearson Correlation was applied were mostly binary which reduces any drawbacks of having outliers. It is obtained as shown in Eq. 1.

$$r = \frac{\sum_{i=1}^{n}(a_i - \overline{a})(b_i - \overline{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \overline{a})^2(b_i - \overline{b})^2}}$$

Here a and b denote the mean of all records in features a and b respectively. The Pearson Correlation returns a value from -1 to 1. Values closer to 1 denotes a high degree of correlation between the two features. In this research certain features which have a degree of collinearity greater than 0.8 with several other unique features were potential candidates for being dropped.

5.4.3. Gradient boosting

Gradient boosting algorithms especially XGBoost are especially useful for classification due to their efficiency and scalability [34]. Gradient boosting algorithms is an ensemble technique which works by fitting additive trees on top of existing decision trees and minimizing the line of steepest descent [35]. Boosting as the name suggests works by increasing the strength of weak learners. Once a decision tree is derived from preliminary classification of a dataset, the loss function for that tree is also calculated. This loss function is derived from the coefficients that are used to fit the model. In subsequent iterations, the model works to decrease this loss

function by increasing the prediction accuracy for classification. For regression problems, the model tries to reduce the difference between the observed and predicted values. In this research, XGBoost was implemented on the two datasets. The algorithm was able to identify the features which were highly important and contributed significantly towards classification. The trained model also returned a set of features that provided little or no contribution towards classification. This information was used in the later stages to exclude less important features for downstream analysis.

5.4.4. Information gain ratio

Information gain or mutual information refers to the prob- ability theory of the mutual dependency between variables. In this scenario, it represents how much information can be obtained about the dependent feature from any of the independent features selected for evaluation. Information gain is a metric used in decision trees to evaluate how good a split has been made in a decision tree for classify the dataset. A higher information gain is derived when their pure split in a node. A pure split denotes that the node is split in such a way that all the records belong to a single class. An impure split denotes a node that produces a split where the records are evenly distributed among all the classes. An impure split is not useful since it is not able to classify records with higher accuracy. Although information gain is a useful metric for evaluation, it suffers from cardinality where it favors attributes with larger number of values. To compensate for this problem, information gain ratio is used to decide a successful split. It is a ratio of the Information Gain to Split Entropy. Split entropy is also referred to as the Intrinsic Value. Eq. 2 denotes the Information Gain from X on Y. Here $H(Y)$ denotes the entropy of feature Y and $H(Y\,x)$ denotes the entropy of Y given the attributes of feature x. Information gain ratio is obtained by dividing the information gain by intrinsic information

$$IG(Y, x) = H(Y) - H(Y|x)$$

5.4.5. Wrapper method

Wrapper methods were also included in the feature selection process. There are several benefits which wrapper methods provide in the feature selection process compare to filtering techniques offered by ANOVA and information gain used earlier. Since it evaluates all possible combinations of the features to determine their importance w.r.t other features it reduces the chances of biases caused when by filter methods such as the ANOVA which treats every feature as independent from another. However, since it is a greedy approach, it suffers from being computationally intensive. The machine learning algorithm used in the wrapper method was random forest classifier. It works by creating multiple decision trees. These trees individually classify the records to belong to a certain class. However, the final decision regarding which class the record belongs to is taken by majority of the votes from the decision trees. This offers a tremendous advantage over a single decision tree classifier, since there is a high possibility of a single decision tree to incorrectly classify a record compared to majority of decision trees. Random forest by default uses approximately 500 decision trees which increases its complexity. It also uses a several features to see which combination of features yields better results. The two wrapper methods used in this research were forward selection and backward elimination. The training process can be stopped after a certain number of iterations have been completed or if the model stops finding any substantial increase in accuracy for a set number of iterations. Random forest also returns how important each feature is in building by computing the GINI importance value of each feature.

In the forward selection process, the algorithm begins by a single feature among all other features that produces the best classification result. Once that feature is selected, the algorithm goes another iteration to locate another feature, which when paired with the first feature would

further increase the classification accuracy of the model. This process keeps repeating for certain number of iterations to identify the combination of features which yields the highest accuracy. The algorithm also returns features which zero importance denoting that those features provide no contribution to enhancing the classification power of the model and hence should be discarded.

Like forward selection, the backward elimination wrapper method also works to select a group of important features. However, instead of selecting one feature at a time and adding features in subsequent iterations, the backward elimination begins by selecting all the features together and then removing features one at a time with each iteration that have little or no effect on increasing the effectiveness of the model.

Forward selection does suffer from a drawback. Since the features are added incrementally to the model, there are scenarios where a combination of features may decrease accuracy and the best possible group may not be discovered since the combination did not include features selected by forward selection in the early stages of iteration . A similar problem may arise with backward elimination method. Using forward and backward selection methodology together allows us to verify if the features selected by both these methods are consistent and reduce this drawback to some extent.

We implemented both these techniques on the two datasets to select another set of best possible features that can be used for classification.

5.5. Experiments

In this section, we test the efficacy of the proposed feature engineering model and compare the results obtained with five of the conventional machine learning algorithms used for cybersecurity dataset knowledge discovery. Hybrid CNN+LSTM, Gated Recurrent Units (GRU), Bi-LSTM, Decision Trees and Random Forest prediction accuracies were compared before and

after application of the proposed feature engineering steps on two cybersecurity datasets. These datasets are NSL-KDD and UNSW-NB15.

5.5.1. Datasets used

One of the two datasets used in this research was the NSL-KDD dataset [36]. This dataset was the successor of the KDD'99 dataset and developed to address certain issues that were present in its predecessor [37]. The KDD'99 dataset developed on DARPA'98 Intrusion Detection System (IDS) evaluation program data contained a significant amount of synthetic data. Study conducted on this dataset revealed that almost 78% of records were duplicated on train dataset and 75% of records were duplicated on test dataset. This redundancy introduced unnecessary bias to records that were more widely available in the dataset compared to records that were not present in a significant amount to provide sufficient information for the machine learning model to train on. The NSL-KDD solved this issue by not including duplicate records in both test and train datasets.

NSL-KDD consists of separate test and training datasets. The training dataset has 125,973 records and the test dataset has 22544 records, each having 42 attributes that could be used for prediction. There are three categorical attributes protocol_type, service, and flag. Variables in these categorical attributes were one-hot encoded before using them as input for training. For example, the protocol_type contains TCP, UDP, and ICMP which, when one-hot-encoded can be represented as [1, 0, 0], [0, 1, 0] and [0, 0, 1] respectively. Attribute protocol type, service, and flag each comprised of 3, 70 and 11 variables respectively in the training dataset. In the test dataset, the service category has 64 variables instead of 70. One- hot-encoding assumes that both training and testing dataset consists of equal number of variables in an attribute column. However, this was not true for the service attribute since the number of variables in test

and training data were different. To address this issue, dummy records were created in the test dataset with the variables before applying one-hot-encoding. The application of one-hot-encoding on categorical variables mentioned above along with the existing continuous attributes now yielded 124 training features from the previously existing 42 attributes.

UNSW-NB15 [38] dataset was also used to verify the use- fulness of this feature engineering methodology. This dataset was generated in the Cyber Range Lab of the Australian Centre for Cyber Security using the IXIA PerfectStorm tool. The 100 Gb of raw traffic data captured by the tcpdump tool had 49 attributes used to identify 9 different types of attacks. The attributes were collected by the lab using Argus, Bro-IDS and a collection of 12 models developed specifically for extracting these features.

In this research the training dataset that consisted of 82,332 records and testing dataset having 175,341 records were merged. This was followed by an 80-20 split where 80% of the records selected at random were used for training and 20% used for testing. This additional step was carried out to give the proposed feature engineering model more information for training. Implementing one-hot-encoding on the categorical variables generated 196 trainable features.

Finally, the prediction class label was converted to a bi- nary class label where 0 represented a normal packet and 1 represented that the packet was malicious.

5.5.2. Output from algorithms used in dimensionality reduction

Univariate feature selection was applied by using one-way ANOVA F-test with the second percentile method on both NSL KDD and UNSW NB-15 datasets. Out of the 123 input features of the NSL KDD dataset ANOVA F-test suggested only 13 features and from 196 features of the UNSW NB-15 dataset the ANOVA F-test resulted a list of only 20 features. These features are listed in Table 5.1.

Table 5.1: Important features from ANOVA test.

Number of features	UNSW Dataset	NSLKDD Dataset
1	rate	logged in
2	sttl	count
3	dload	serror rate
4	swin	srv serror rate
5	stcpb	same srv rate
6	dtcpb	dst host srv count
7	dwin	dst host same srv rate
8	dmean	dst host serror rate
9	ct srv src	dst host srv serror rate
10	ct state ttl	service http
11	ct src dport ltm	service private
12	ct dst sport ltm	flag S0
13	ct dst src ltm	flag SF
14	ct src ltm	
15	ct srv dst	
16	proto tcp	
17	service dns	
18	state CON	
19	state FIN	
20	state INT	

Table 5.2: Pearson Correlation from UNSW NB-15 dataset

Feature	Correlated feature	Correlation Score
sbytes	spkts	0.96407
dbytes	dpkts	0.97297
sloss	spkts	0.97157
sloss	sbytes	0.9957
dloss	dpkts	0.97932
dloss	dbytes	0.99661
dwin	swin	0.98066
synack	tcprtt	0.94583
ackdat	tcprtt	0.91937
ct dst ltm	ct srv src	0.84054
ct src dport ltm	ct srv src	0.86202
ct src dport ltm	ct dst ltm	0.96134
ct dst sport ltm	ct srv src	0.81475
ct dst sport ltm	ct dst ltm	0.87092
ct dst sport ltm	ct src dport ltm	0.90837
ct dst src ltm	ct srv src	0.9539
ct dst src ltm	ct dst ltm	0.85705
ct dst src ltm	ct src dport ltm	0.87201
ct dst src ltm	ct dst sport ltm	0.83627
ct ftp cmd	is ftp login	0.9991
ct src ltm	ct dst ltm	0.90098

Based on the linear complexity, Pearson correlation was chosen for feature elimination process. We performed Karl Pearson Correlation test on both NSL KDD and UNSW- NB15 datasets. The correlation threshold limit was set to 80 percentiles. Features which have correlation value more than the set threshold were listed as highly correlated features. We listed these correlated features for both of the datasets in Table 5.2 and Table 5.3. Features appeared multiple times in these tables are indicating that, these features are highly correlated with multiple features. Hence these tables will help us to only remove the necessary features and to select the important features individually.

A benefit of using gradient boosting tree is retrieving importance scores for each attributes relatively straight forward manner when the boosted tree is constructed. Importance primarily calculated for every single decision tree by the amount that each attribute split improves the performance measure which is weighted by the number of observations the node is responsible for. This feature importance is then averaged for all the decision trees within the model. We experimented both of the NSL KDD and UNSW-NB15 dataset with this Gradient Boosting Feature Importance approach. Out of the 123 features obtained from one hot encoding of NSL KDD dataset as input the XGBoost classifier suggested 51 very important features with importance highest importance score and 72 of these features were assigned zero importance. The UNSW-NB15 dataset after one hot encoding had 196 features which were used as the input of XGBoost classifier and returned 47 features as important and 149 features having zero importance.

Table 5.3: Pearson Correlation from NSL-KDD dataset

Feature	Correlated Feature	Correlation Score
num root	num compromised	0.964065
is guest login	hot	0.972966
srv serror rate	serror rate	0.97157
srv rerror rate	rerror rate	0.995699
dst host same srv rate	dst host srv count	0.979316
dst host serror rate	serror rate	0.996612
dst host serror rate	srv serror rate	0.980664
dst host srv serror rate	serror rate	0.945829
dst host srv serror rate	srv serror rate	0.919371
dst host srv serror rate	dst host serror rate	0.840543
dst host rerror rate	rerror rate	0.862021
dst host rerror rate	srv rerror rate	0.961335
dst host srv rerror rate	rerror rate	0.814751
dst host srv rerror rate	srv rerror rate	0.870924
dst host srv rerror rate	dst host rerror rate	0.908371
service ftp	is guest login	0.953904
flag REJ	rerror rate	0.857049
flag REJ	srv rerror rate	0.872005
flag REJ	dst host rerror rate	0.836266
flag REJ	dst host srv rerror rate	0.999104

For both datasets, the cut off was set by the cumulative feature importance of the features as 99%. The cumulative feature importance is the weighted average of individual feature importance. UNSW dataset implies that, 47 features are dominating 99% of the total feature importance. In NSL-KDD, 51 features are dominating 99% of the feature importance. The graphs shown in Fig 5.1 and Fig 5.2 highlight these features.

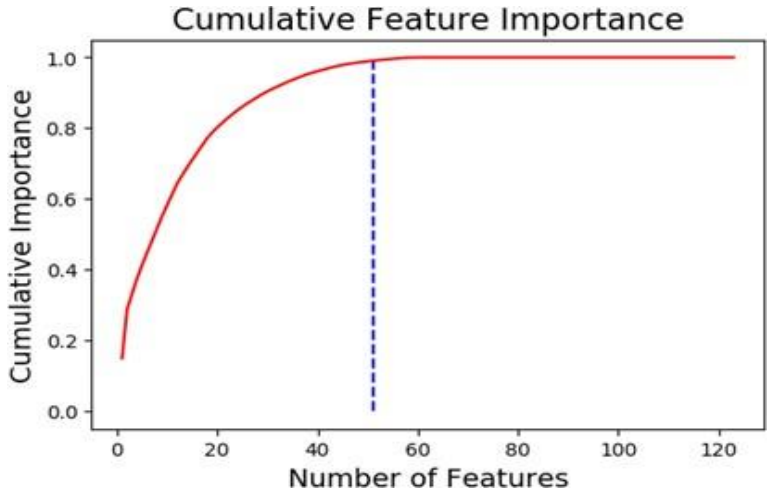

Fig 5.1: Gradient Boosting importance for NSL-KDD dataset.

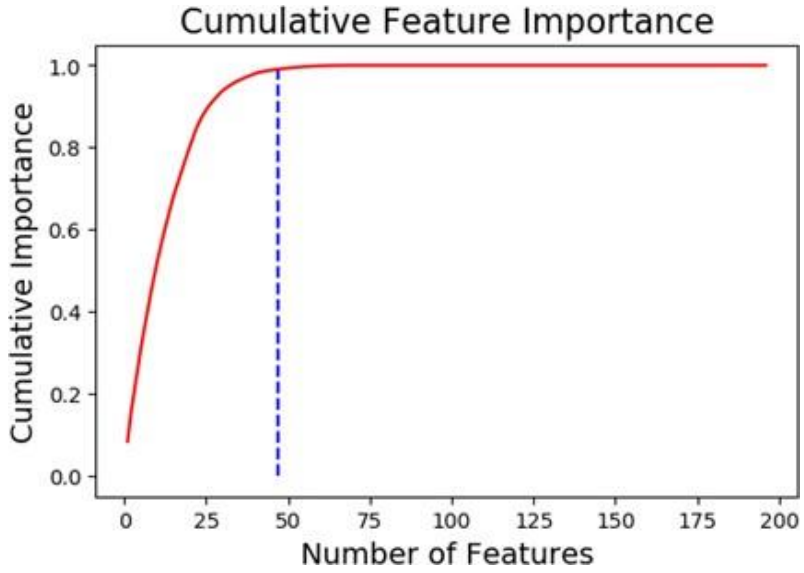

Fig 5.2: Gradient Boosting importance for UNSW dataset.

Information gain mostly calculates the entropy reduction from transforming dataset. Besides being used in Decision Tree Classifiers to define the best split ratio, it is also used to evaluate every feature in the training dataset in the context of target variable. We calculated both Information Gain and Gain ratio to check for any imbalance dependency between features. The one hot encoded dataset was used as the input of this experiment. For the dataset we limited our Information gain output list size to same as the Important feature list size. We documented and

listed the top 51 Information gain output list for NSL-KDD dataset and 47 feature list for the UNSW- NB15 dataset.

Wrapper method employs a greedy approach which works by the evaluating multiple subsets of features with a machine learning algorithm. This algorithm employs a search strategy to find the space of possible feature subsets and evaluates the performance by multiple iterations. In this experiment, Random Forest classifier was used to evaluate the performance for every possible combination of the subset of both the datasets. We performed both forward sequential feature selection process and backward sequential feature elimination process for both of the datasets. to select the most important features. The output size was limited to 47 features for the UNSW-NB15 and 50 features for the NSL KDD dataset based on Area Under the Curve (AUC) score. The features returned by the wrapper method are shown in Table 5.4.

5.5.3. Meta learner

Meta learning which is also inspired by cognitive psychology is a proven effective method for machine learning techniques to excel at mastering predictive skill [39], [40]. When a meta learner is applied to the machine learning algorithms, it uses prior experience to change certain aspects of an algorithm/s which can be simplified in terms like, meta learning is how the algorithms learns how to learn [41]. Machine learning algorithms are getting better over the time, but still lacks versatility which is achieved by intelligent amalgamation of meta learning along with similar techniques such as reinforcement learning, transfer learning and active learning [42], [43]. Researchers also focus on using meta learning for hyperparameter tuning, neural network optimization, specifying best network architecture and special cases like few-shot image recognition. In this research, we have used the meta learner to optimize hyperparameters through a bagging ensemble process [44]. We have introduced a new algorithm to find out the best

features after carefully investigating the performance and robustness of the predictive model. This dynamic feature selector algorithm takes the output of ANOVA F-test, Pearson Correlation, Gradient Boosting, Information Gain and Wrapper Method as inputs and finds out the best subset of features after evaluation through a bagging algorithm. In this meta learner approach, the bagging algorithm uses five different algorithms namely, a hybrid of Convolutional Neural Network (CNN) and Long Short term Memory (LSTM) proposed in [45], BiLSTM, Gated recurrent units (GRU), Decision Tree and Random Forest. Since our dataset have multiple features which is addressed to sessions, so we selected different Recurrent Neural Network (RNN) predict the intrusions. CNN, which is also known as ConvNet, is very efficient in image classification and predicting multivariate datasets. Unlike CNN, Recurrent Neural Networks help to create interaction between input sequence. In case of predicting the next element of the series or batch, the previous features or elements of the series or batch needed to remember for future element, and RNN creates a loop which helps to keep track of these information [46], [47], [48], [49], [50]. Bidirectional Long Short Term Memory (BiLSTM) usually brings together two independent RNNs which enable running input in two directions as future and past [51], [52]. BiLSTM is a modified version of LSTM which is proven efficient is multiple domains for binary prediction. Researchers have devised multiple techniques to hybridize the features using LSTM which can extract the long-term dependencies of the data features within a sequence to improve the prediction accuracy [53], [54], [55], [56], [57], [58]. In this research we have used a relevant strategy consists of multiple convolutional kernels to extract features from the dataset and establish end to end mapping of the relationship between the features and attack types. The hybrid of CNN and LSTM is developed in two steps such as, feature extraction based on CNN

followed by a feature fusion part based on LSTM. This specific hybrid approach was proven effective on NSL KDD dataset without any feature reduction before [45].

Tree based algorithms sometime outperforms neural net- works because they approach problems in a similar way by deconstructing them piece-by-piece, instead of finding one complex decision boundary that can separate the entire dataset like Support Vector Machine (SVM) or Logistic Regression. Tree-based methods progressively split the feature space along various features to optimize the total gain whereas neurons of neural networks computes the probability of specific section of feature space with various overlapping. To compare the performance of neural networks we also introduced two tree- based algorithms; Decision Tree and Random Forest to get deterministic view.

Algorithm 1: How to write algorithms

Result: Write here the result
L1[ ]← ANOVA F-test;
L2[ ] ← Most Important;
L3[ ] ← Information Gain;
L4[ ] ← Zero Importance;
L5[ ] ← Information Gain Ratio;
L6[ ] ← Wrapper Method;
D1[Dictionary] ← Correlated feature;
Exclude the zero important features;
if *L3 == L5* **then**
    Remove L5;
else
    L3← $(L3 \cup L5)$;
end
while *Accuracy([i+1] > [i])* **do**
    Sort (L1, L2, L3, L6);
    $L[\,] \leftarrow \cap_n L[\,]1,2,3,6$;
    If any common between L and D1, exclude the lower important features based on XGBoost importance table(Below) according to most L1 or L2 or L3 or L6 ;
    Run the bagging algorithm; Save accuracy in P[i];
    Remove lower length list from L1, L2, L3, L6;
end

Table 5.4: Importance of final selected features of NSL-KDD

Features	XGBoost Importance	Information Gain
dst host srv count	383.4	0.417932243
count	275.3	0.416429782
diffic	1223.5	0.261177691
dst host count	309.4	0.208878373
dst host diff srv rate	306.4	0.451766677
dst host rerror rate	269	0.097905873
dst host same src port rate	206.1	0.235775166
dst host same srv rate	336.4	0.400381603
dst host serror rate	273.2	0.398612212
dst host srv diff host rate	309	0.261385661
duration	196.3	0.06468874
flag SF	58.4	0.489171442
logged in	123.9	0.30634371
root shell	176.5	0.001810072
service http	98.4	0.00112
src bytes	1321.7	0.720243598

5.5.4. Dynamic feature selection

To summarize the feature selection process, we begin by excluding all the zero important features from one hot encoded total feature list of both the datasets. This is followed by finding out the common features of ANOVA F-test as L1, most important feature list as L2, wrapper method as L6 and Information gain list as L3. We selected 13 features from NSL- KDD and 20 features from UNSW datasets to begin with. This was followed by a check of any of the common features that were present in the Correlation dictionary D1. If common feature pairs were present, we analyzed the pair and removed one of them according to the most important feature weight. Features were removed if they had a correlation more than 80%. Then we feed the bagging algorithm the selected features and record the accuracy. After one iteration, the lowest length list of ANOVA F-test, most important, Information gain list, and wrapper method was removed and the common features from the rest of the three list were evaluated. Two new features were added during each iteration to the list at a time and the accuracy of the current run

was compared with the previous accuracy stored in P. If accuracy is decreased, we stop the experiment and stick with the first prediction. If the accuracy is increased, then we compare the latest feature list with Most important feature list and try to add the uncommon variables one by one according to importance weight and perform the bagging algorithm until the prediction accuracy stops increasing. When the accuracy stopped increasing, we extract the final feature list. Table 5.5 and table 5.4 shows the list of features that were considered most important in the dynamic feature selection process.

Table 5.5: Importance of final selected features of UNSW

Features	XGBoost Importance	Information Gain
sbytes	2423.9	0.472181
dtcpb	1575.6	0.381139
ct srv src	1572.6	0.082749
stcpb	1519.7	0.381261
ackdat	1311.9	0.340099
ct srv dst	1215.6	0.097382
tcprtt	1098.5	0.359983
dur	1033.7	0.539205
ct src ltm	911	0.074548
dload	891.1	0.493778
ct dst ltm	712.2	0.083325
rate	664.1	0.539682
dmean	663.9	0.283712
response body len	506.2	0.035949
ct src dport ltm	462.9	0.094077
ct dst sport ltm	379.1	0.150217
sloss	321.8	0.109831
ct state ttl	281.1	0.313659
dwin	231.4	0.058227
dpkts	177.1	0.242715
ct flw http mthd	152.8	0.000854
is ftp login	115.6	8.39E-05
service dns	102.3	0.036725
trans depth	92.3	9.79E-05
service http	91.9	5.87E-05
proto tcp	79.5	0.067707
state CON	49	0.061645
state FIN	25.4	0.047925
state INT	22.2	0.147284

5.6. Results and discussion

We have conducted several experiments with both UNSW- NB15 and NSL KDD dataset and in every experiment resulted consistent increment in performance using Dynamic Feature Selection algorithm. Fig 5.3 and Fig 5.4 show the prediction of NSL KDD and UNSW-NB15 dataset respectively without using Dynamic Feature selector Fig 5.5 and Fig 5.6 represents the output of Dynamic Feature Selector algorithm of NSL KDD and UNSW-NB15 dataset respectively. NSL KDD shows increment in accuracy for all of the algorithms using dynamic feature selector. The best algorithms are selected as Random forest whose accuracy jumped from 99.54% to 99.64% with reducing the feature size from 123 to 50.

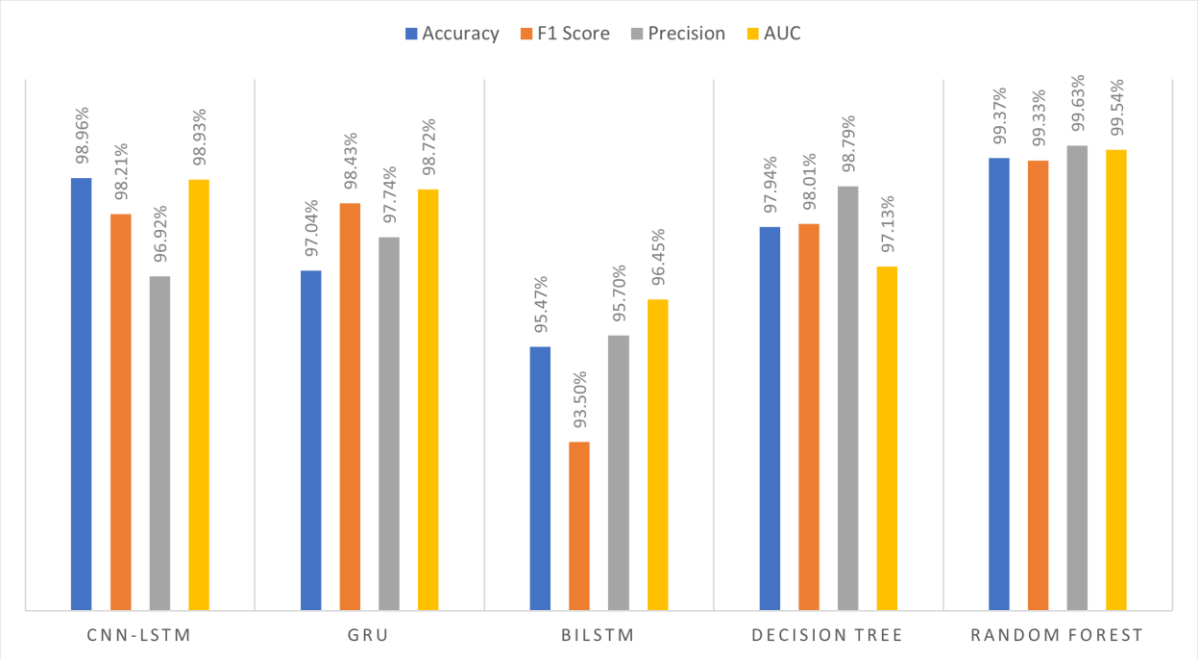

Fig 5.3: Performance without dynamic feature selector using NSL KDD data.

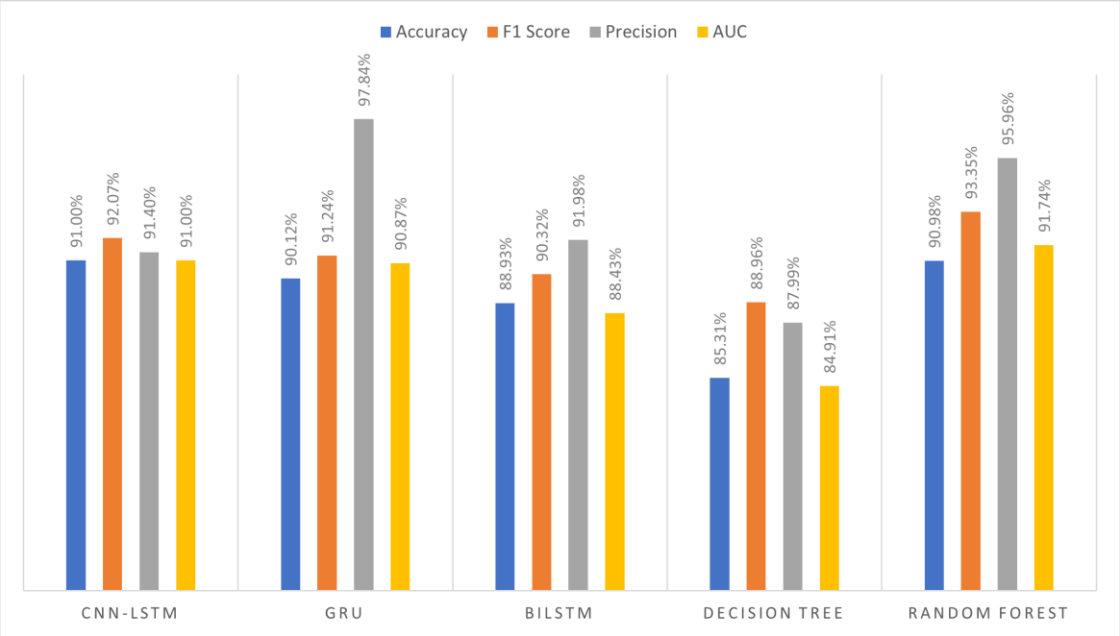

Fig 5.4: Performance without dynamic feature selector using UNSW-NB15 data.

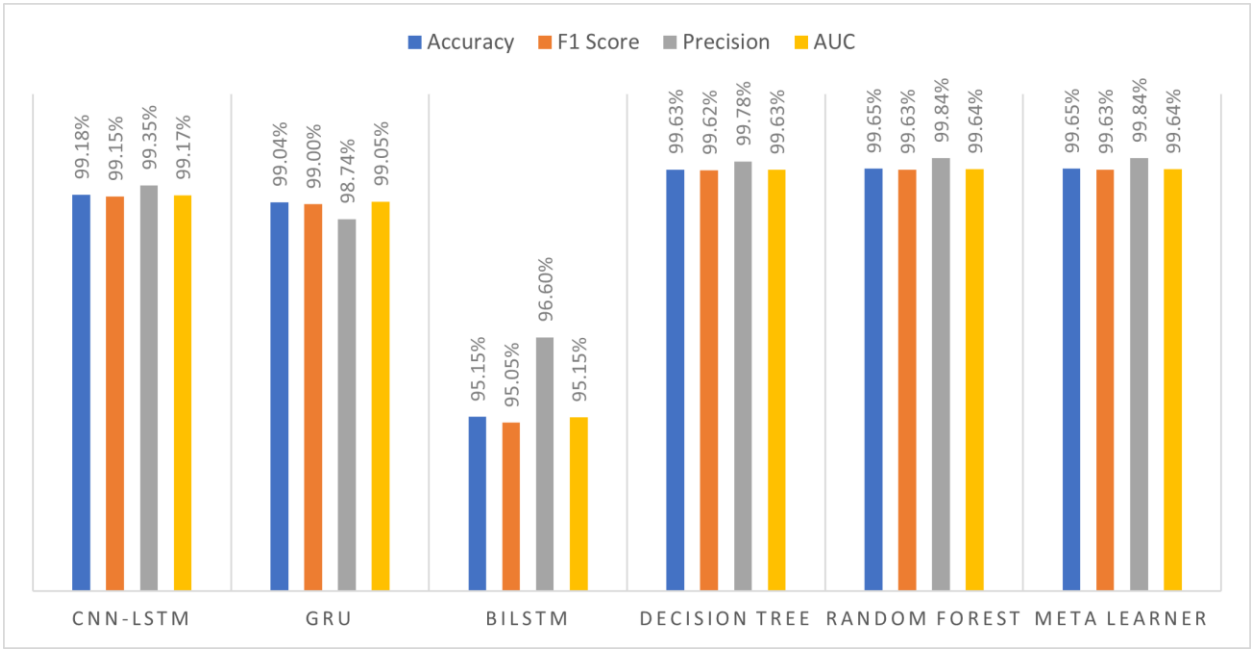

Fig 5.5: Performance with dynamic feature selector using NSL KDD data.

The UNSW-NB15 dataset also showed consistent performance increase using the best algorithm as Random Forest from 90.98% to 92.46% with reducing the feature size from 196 to 47. From table 6 we can notice that GRU produced a better precision, but the meta learner

selected the Random Forest algorithm to produce output based on the overall performance metrics such as F1 score, accuracy and AUC. Accuracy is the most intuitive performance measure and a simple way to observe prediction, but it is often misleading due to specificity and sensitivity [59]. Also, since the class distribution is uneven, F1 score is a better indicator to select robustness of the model. For both of the dataset, dynamic feature selector shows an increment of F1 score and accuracy which clearly states that, this method reduces the feature size effectively with increment in performance as well.

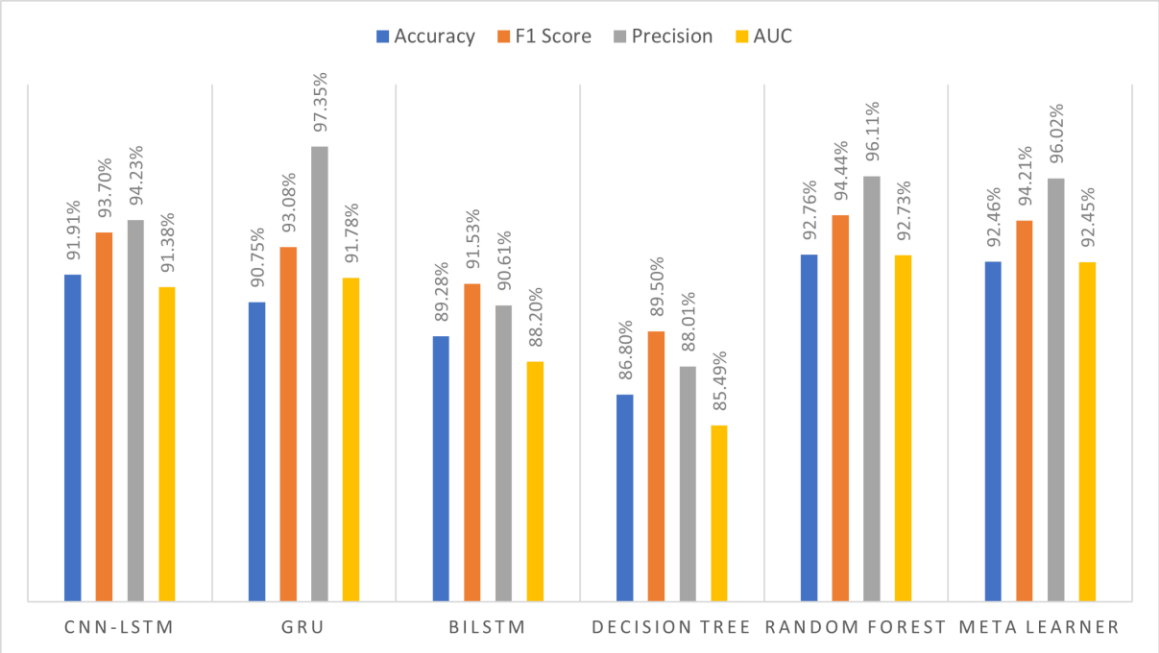

Fig 5.6: Performance with dynamic feature selector using UNSW-NB15 data.

5.7. Conclusion

Network security has become an essential issue in any distributed system. Although a lot of machine learning algorithms have been experimented to increase the efficacy of Intrusion detection, it is still a major challenge for existing intrusion detection algorithms to achieve good performance. In this research we have dealt with two high dimensional network traffic datasets and proposed a novel Dynamic Feature Selector (DFS) algorithm which is based on feature

selection and meta learning technique. Our approach combined five prominent algorithms proven effective in previous for two well established dataset of NSL KDD and UNSW-NB15. Primarily we conducted multiple feature engineering process using statistical analysis like univariate test and Pearson coefficient test. Later on, we conducted experiment with XGBoost importance, wrapper technique and information gain ratio and documented all the outputs generated from these algorithms. These algorithms have helped to make better decisions to reduce feature size in numerous studies but failed to produce better output when applied individually. In this research DFS has introduced a novel logic to use bagging ensemble technique as a meta learner which is used as optimizer for feature selector and ranking from multiple feature selection process. The output of different feature selection process and algorithms are filtered in a stochastic process with the help of Meta learner consists of five state of art classification techniques to increase the performance of the model. With the usage of DFS, we were able to reduce the number of features by more than 50% and increase the predictive performance significantly. Besides that, DFS also suggested the best algorithms out of the five machine learning algorithms which is reproducible and deployment ready. Future research will include the extension of DFS algorithm for reinforcement learning to prevent the intrusion in network. We are very much hopeful that, self-learning techniques will enhance the performance of DFS and extend its application to datasets from other domains to produce a robust machine learning model. Future work will be conducted by implementing DFS on different live network to document its deployment performance and adjust hyperparameters of meta learner to enhance cybersecurity.

5.8. References

1. Breiman, L., 1996. Bagging predictors. *Machine learning*, *24*(2), pp.123-140.

2. Hu, H., Li, J., Plank, A., Wang, H. and Daggard, G., 2006. A comparative study of classification methods for microarray data analysis. In *Proceedings of the 5th*

Australasian Data Mining Conference (AusDM 2006): Data Mining and Analytics 2006 (pp. 33-37). ACS Press.

3. Niranjan, A., Prakash, A., Veena, N., Geetha, M., Shenoy, P.D. and Venugopal, K.R., 2017, December. EBJRV: An Ensemble of Bagging, J48 and Random Committee by Voting for Efficient Classification of Intrusions. In *2017 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)* (pp. 51-54). IEEE.

4. Chen, T. and Guestrin, C., 2016, August. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

5. Belouch, M. and hadaj, S.E., 2017, March. Comparison of ensemble learning methods applied to network intrusion detection. In *Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing* (pp. 1-4).

6. Liu, J., Kantarci, B. and Adams, C., 2020, July. Machine learning-driven intrusion detection for contiki-NG-based IoT networks exposed to NSL-KDD dataset. In *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning* (pp. 25-30).

7. Seger, C., 2018. An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing.

8. Cerda, P., Varoquaux, G. and Kégl, B., 2018. Similarity encoding for learning with dirty categorical variables. *Machine Learning*, *107*(8), pp.1477-1494.

9. Khan, S., 2019. DeepAcid: Classification of macromolecule type based on sequences of amino acids. *arXiv preprint arXiv:1907.03532*.

10. Ding, Y. and Zhai, Y., 2018, December. Intrusion detection system for NSL-KDD dataset using convolutional neural networks. In *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence* (pp. 81-85).

11. Kim, T., Suh, S.C., Kim, H., Kim, J. and Kim, J., 2018, December. An encoding technique for CNN-based network anomaly detection. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 2960-2965). IEEE.

12. Su, T., Sun, H., Zhu, J., Wang, S. and Li, Y., 2020. BAT: deep learning methods on network intrusion detection using NSL-KDD dataset. *IEEE Access*, *8*, pp.29575-29585.

13. Cohen, J., Cohen, P., West, S.G. and Aiken, L.S., 2013. *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge.

14. Tran, B., Xue, B. and Zhang, M., 2017, August. Class dependent multiple feature construction using genetic programming for high-dimensional data. In Australasian Joint Conference on Artificial Intelligence (pp. 182-194). Springer, Cham.

15. Krishna, G.J. and Ravi, V., 2019, January. Feature subset selection using adaptive differential evolution: an application to banking. In Proceedings of the ACM India Joint International Conference on Data Science and Management of Data (pp. 157-163).

16. Wang, L., Zhou, N. and Chu, F., 2008. A general wrapper approach to selection of class-dependent features. IEEE Transactions on Neural Networks, 19(7), pp.1267-1278.

17. Tran, B., Zhang, M. and Xue, B., 2016, December. Multiple feature construction in classification on high-dimensional data using GP. In 2016 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1-8). IEEE.

18. Hariharakrishnan, J., Mohanavalli, S. and Kumar, K.S., 2017, January. Survey of pre-processing techniques for mining big data. In 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP) (pp. 1-5). IEEE.

19. Enache, A.C., Sgarciu, V. and Petrescu-Niţă, A., 2015, May. Intelligent feature selection method rooted in Binary Bat Algorithm for intrusion detection. In 2015 IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics (pp. 517-521). IEEE.

20. Camargo, C.O., Faria, E.R., Zarpelão, B.B. and Miani, R.S., 2018, June. Qualitative evaluation of denial of service datasets. In Proceedings of the XIV Brazilian Symposium on Information Systems (pp. 1-8).

21. Bachl, M., Hartl, A., Fabini, J. and Zseby, T., 2019, December. Walling Up Backdoors in Intrusion Detection Systems. In Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks (pp. 8-13).

22. Liu, H., Liu, Z., Liu, Y. and Gao, X., 2019, November. Abnormal Network Traffic Detection based on Leaf Node Density Ratio. In Proceedings of the 2019 the 9th International Conference on Communication and Network Security (pp. 69-74).

23. Faker, O. and Dogdu, E., 2019, April. Intrusion detection using big data and deep learning techniques. In Proceedings of the 2019 ACM Southeast Conference (pp. 86-93).

24. Thejas, G.S., Jimenez, D., Iyengar, S.S., Miller, J., Sunitha, N.R. and Badrinath, P., 2020. COMB: A Hybrid Method for Cross-validated Feature Selection. In ACM Southeast Regional Conference (pp. 100-106).

25. Belouch, M., Elhadaj, S. and Idhammad, M., 2018. A hybrid filter-wrapper feature selection method for DDoS detection in cloud computing. Intelligent Data Analysis, 22(6), pp.1209-1226.

26. Khammassi, C. and Krichen, S., 2017. A GA-LR wrapper approach for feature selection in network intrusion detection. computers & security, 70, pp.255-277.

27. Tun, M.T., Nyaung, D.E. and Phyu, M.P., 2020, July. Network Anomaly Detection using Threshold-based Sparse. In Proceedings of the 11th International Conference on Advances in Information Technology (pp. 1-8).

28. Viet, H.N., Van, Q.N., Trang, L.L.T. and Nathan, S., 2018, June. Using deep learning model for network scanning detection. In Proceedings of the 4th International Conference on Frontiers of Educational Technologies (pp. 117-121).

29. Primartha, R. and Tama, B.A., 2017, November. Anomaly detection using random forest: A performance revisited. In 2017 International conference on data and software engineering (ICoDSE) (pp. 1-6). IEEE.

30. Mohammadi, S., Mirvaziri, H., Ghazizadeh-Ahsaee, M. and Karimipour, H., 2019. Cyber intrusion detection by combined feature selection algorithm. Journal of information security and applications, 44, pp.80-88.

31. Liu, H. and Yu, L., 2005. Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on knowledge and data engineering, 17(4), pp.491-502.

32. Guyon, I. and Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), pp.1157-1182.

33. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y. and Cho, H., 2015. Xgboost: extreme gradient boosting. R package version 0.4-2, 1(4).

34. Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics, pp.1189-1232.

35. Rios, A.L.G., Li, Z., Bekshentayeva, K. and Trajković, L., 2020, October. Detection of denial of service attacks in communication networks. In 2020 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-5). IEEE.

36. Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A.A., 2009, July. A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1-6). IEEE.

37. Moustafa, N. and Slay, J., 2015, November. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 military communications and information systems conference (MilCIS) (pp. 1-6). IEEE.

38. Nichol, A., Achiam, J. and Schulman, J., 2018. On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999.

39. Finn, C., Abbeel, P. and Levine, S., 2017, July. Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning (pp. 1126-1135). PMLR.

40. Lemke, C., Budka, M. and Gabrys, B., 2015. Metalearning: a survey of trends and technologies. Artificial intelligence review, 44(1), pp.117-130.

41. Cruz, R.M., Sabourin, R., Cavalcanti, G.D. and Ren, T.I., 2015. META-DES: A dynamic ensemble selection framework using meta-learning. Pattern recognition, 48(5), pp.1925-1935.

42. Lin, S.C., Yuan-chin, I.C. and Yang, W.N., 2009. Meta-learning for imbalanced data and classification ensemble in binary classification. Neurocomputing, 73(1-3), pp.484-494.

43. Dvornik, N., Schmid, C. and Mairal, J., 2019. Diversity with cooperation: Ensemble methods for few-shot classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3723-3731).

44. Ahsan, M. and Nygard, K.E., 2020, March. Convolutional Neural Networks with LSTM for Intrusion Detection. In CATA (pp. 69-79).

45. Fu, R., Zhang, Z. and Li, L., 2016, November. Using LSTM and GRU neural network methods for traffic flow prediction. In 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC) (pp. 324-328). IEEE.

46. Dey, R. and Salem, F.M., 2017, August. Gate-variants of gated recurrent unit (GRU) neural networks. In 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS) (pp. 1597-1600). IEEE.

47. Chang, L.Y. and Chen, W.C., 2005. Data mining of tree-based models to analyze freeway accident frequency. Journal of safety research, 36(4), pp.365-375.

48. Aldous, D., 1993. Tree-based models for random distribution of mass. Journal of Statistical Physics, 73(3), pp.625-641.

49. Aldous, D. and Larget, B., 1992. A tree-based scaling exponent for random cluster models. Journal of Physics A: Mathematical and General, 25(17), p.L1065.

50. Yang, Y., Morillo, I.G. and Hospedales, T.M., 2018. Deep neural decision trees. arXiv preprint arXiv:1806.06988.

51. Zhang, L. and Xiang, F., 2018, June. Relation classification via BiLSTM-CNN. In International Conference on Data Mining and Big Data (pp. 373-382). Springer, Cham.

52. Sharfuddin, A.A., Tihami, M.N. and Islam, M.S., 2018, September. A deep recurrent neural network with bilstm model for sentiment classification. In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP) (pp. 1-4). IEEE.

53. Yang, Y., Tong, Y., Ma, S. and Deng, Z.H., 2016, November. A position encoding convolutional neural network based on dependency tree for relation classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (pp. 65-74).

54. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R. and Schmidhuber, J., 2016. LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, 28(10), pp.2222-2232.

55. Tsironi, E., Barros, P., Weber, C. and Wermter, S., 2017. An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition. Neurocomputing, 268, pp.76-86.

56. Zhou, X., Hu, B., Chen, Q. and Wang, X., 2018. Recurrent convolutional neural network for answer selection in community question answering. Neurocomputing, 274, pp.8-18.

57. Zhao, R., Yan, R., Wang, J. and Mao, K., 2017. Learning to monitor machine health with convolutional bi-directional LSTM networks. Sensors, 17(2), p.273.

58. Nunez, J.C., Cabido, R., Pantrigo, J.J., Montemayor, A.S. and Velez, J.F., 2018. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. Pattern Recognition, 76, pp.80-94.

59. Powers, D.M., 2020. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061.

60. Lakhina, S., Joseph, S. and Verma, B., 2010. Feature reduction using principal component analysis for effective anomaly–based intrusion detection on NSL-KDD.

61. Benaddi, H., Ibrahimi, K. and Benslimane, A., 2018, October. Improving the intrusion detection system for nsl-kdd dataset based on pca-fuzzy clustering-knn. In 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 1-6). IEEE.

62. Ikram, S.T. and Cherukuri, A.K., 2016. Improving accuracy of intrusion detection model using PCA and optimized SVM. Journal of computing and information technology, 24(2), pp.133-148.

63. Gao, J., Chai, S., Zhang, B. and Xia, Y., 2019. Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis. Energies, 12(7), p.1223.

64. Kumar, V., Sinha, D., Das, A.K., Pandey, S.C. and Goswami, R.T., 2020. An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset. Cluster Computing, 23(2), pp.1397-1418.

65. Gottwalt, F., Chang, E. and Dillon, T., 2019. CorrCorr: A feature selection method for multivariate correlation network anomaly detection techniques. Computers & Security, 83, pp.234-245.

66. Lee, S., 2010. Drawbacks of Principal component analysis. arXiv preprint arXiv:1005.1770.

67. Desale, R.P. and Verma, S.V., 2013, February. Study and analysis of PCA, DCT & DWT based image fusion techniques. In 2013 International Conference on Signal Processing, Image Processing & Pattern Recognition (pp. 66-69). IEEE.

68. Pechenizkiy, M., Tsymbal, A. and Puuronen, S., 2004, June. PCA-based feature transformation for classification: issues in medical diagnostics. In Proceedings. 17th IEEE Symposium on Computer-Based Medical Systems (pp. 535-540). IEEE.

69. Ahsan, M., Gomes, R., Chowdhury, M. and Nygard, K.E., 2021. Enhancing Machine Learning Prediction in Cybersecurity Using Dynamic Feature Selector. Journal of Cybersecurity and Privacy, 1(1), pp.199-218.