WINDOWS OS VULNERABILITY CLASSIFICATION USING MACHINE LEARNING TECHNIQUES

2024
MASTER THESIS
COMPUTER ENGINEERING

Nooralhuda Abdulhasan Hadi AL-SARRAY

Thesis Advisor
Assist. Prof. Dr. Sait DEMİR

WINDOWS OS VULNERABILITY CLASSIFICATION USING MACHINE LEARNING TECHNIQUES

Nooralhuda Abdulhasan Hadi AL-SARRAY

Thesis Advisor
Assist. Prof. Dr. Sait DEMİR

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

KARABUK
January 2024

I certify that in my opinion the thesis submitted by Nooralhuda Abdulhasan Hadi AL-SARRAY titled "WINDOWS OS VULNERABILITY CLASSIFICATION USING MACHINE LEARNING TECHNIQUES" is fully adequate in scope and in quality as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Sait DEMİR .....
Thesis Advisor, Department of Computer Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science Thesis. Jan 30, 2024

Examining Committee Members (Institutions)	Signature
Chairman : Assist. Prof. Dr. Adib HABBAL (KBU)	.....
Member : Assist. Prof. Dr. Muhammet ÇAKMAK (SNU)	.....
Member : Assist. Prof. Dr.Sait DEMİR (KBU)	.....

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

Assoc. Prof. Dr. Zeynep ÖZCAN .....
Director of the Institute of Graduate Programs

"I hereby state that all the information incorporated in this thesis has been collected and presented in accordance with academic regulations and ethical principles. Moreover, I have conscientiously adhered to the demands specified by these regulations and principles, duly acknowledging all sources referenced in this work that are not original to it."

Nooralhuda Abdulhasan Hadi AL-SARRAY

ABSTRACT

M. Sc. Thesis

WINDOWS OS VULNERABILITY CLASSIFICATION USING MACHINE LEARNING TECHNIQUES

Nooralhuda Abdulhasan Hadi AL-SARRAY

Karabuk University
Institute of Graduate Programs
Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Dr. Sait DEMİR
January 2024, 52 Pages

The speedy development of technology and communication systems leads to the emergence of many challenges, especially in the field of data protection and maintaining information security. It is newly known as the field of cybersecurity, which includes a set of procedures and techniques that seek to maintain data security. Through this study, we have used machine learning to improving cybersecurity of the Windows system. We have used five machine learning classification algorithms (Random Forest, Logistic Regression, Naive Bayes, K-Nearest Neighbors, and SVM) to classify the Windows system's vulnerabilities. We have collected the dataset from two sites exploit-deb and NIST (National Institute of Standards and Technology).

Several parameters were calculated during the study. The results revealed that the highest degree of accuracy was achieved when using the Random Forest algorithm (accuracy: 0,97%, precision: 0,97%, recall: 0,97%, F1-score: 0,97%, and Roc Auc

score: 0,99%), which means achieving an accuracy of 97%. The results highlight the Random Forest algorithm's ability to solve the vulnerability classification problem.

Keywords: Vulnerability Classification, Machine Learning, Random Forest, Logistic Regression, Naive Bayes, K-Nearest Neighbors, SVM.

Science Code : 92432

ÖZET

Yüksek Lisans Tezi

MAKINE ÖĞRENIMI TEKNIKLERINI KULLANARAK WINDOWS İŞLETIM SISTEMI GÜVENLIK AÇIĞI SINIFLANDIRMASI

Nooralhuda Abdulhasan Hadi AL-SARRAY

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi Sait DEMİR
Ocak 2024, 52 sayfa

Teknoloji ve iletişim sistemlerin hızla gelişmesi, özellikle veri koruma ve bilgi güvenliğini sağlama alanında birçok zorluğun ortaya çıkmasına neden olmaktadır. Veri güvenliğini sağlamayı amaçlayan bir dizi prosedür ve teknik içeren siber güvenlik alanı yeni bir alan olarak bilinmektedir. Bu çalışmada, Windows sisteminin siber güvenliğini iyileştirmek için makine öğrenimini kullanılmıştır. Windows sisteminin güvenlik açıklarını sınıflandırmak için beş makine öğrenimi sınıflandırma algoritması (Random Forest, Logistic Regression, Naive Bayes, K-Nearest Neighbors, SVM) kullanılmıştır. Veri seti exploit-deb ve NIST (National Institute of Standards and Technology) sitelerinden elde edilmiştir.

Çalışma sırasında çeşitli parametreler hesaplanmıştır. Sonuçlar, en yüksek doğruluk derecesinin Rastgele Orman algoritması kullanıldığında elde edildiğini ortaya koymuştur (doğruluk: % 0,97, hassasiyet: % 0,97, hatırlama: 0,97, F1-skoru: %0,97 ve Roc Auc skoru: %0,99), %97'lik bir doğruluk elde edilmiştir. Sonuçlar, Rastgele

Orman algoritmasının güvenlik açığı sınıflandırma problemini çözme yeteneğini vurgulamaktadır.

Anahtar Kelimeler: Güvenlik Açığı Sınıflandırması, Makine Öğrenimi, Random Forest, Logistic Regression, Naive Bayes, K-Nearest Neighbors, SVM.

Bilim Kodu : 92432

ACKNOWLEDGMENT

In acknowledgement, I can only thank everyone who stood with me in this important stage of my life.

Firstly, my family, I would like to thank them for standing with me throughout my studies.

I extend my sincere thanks to my supervisor in this study, Dr. Sait DEMİR.

I also extend my sincere thanks to all my friends who supported me, and I would also like to thank the distinguished professors I met at Karabuk University who were an example to follow.

Finally, I would like to thank God. I am grateful to God for what I have achieved.

CONTENTS

Page

Page

LIST FIGURES

Page

LIST OF TABLES

Page

ABBREVITIONS INDEX

OS : Operating System

ICT : Information and Communication Technology

CIA : Confidentiality, Integrity, Availability

ML : Machin Learning

AI : Artificial Intelligent

ICS : Industrial Control Systems

CVE : Common Vulnerabilities and Exposures

CVSS: Common Vulnerability Scoring System

NIST : National Institute of Standards and Technology

NVD : National Vulnerability Database

CNN : Convolutional Neural Network

MCC : Mathews correlation coefficient

RF : Random Forest

K-NN: K-Nearest Neighbors

SVM : Support Vector Machines

ROC : Receiver Operating Characteristics

AUC : Area Under Curve

PART 1

INTRODUCTION

One of the most significant issues that software faces is vulnerabilities; when a malicious hacker discovers vulnerabilities then, they have the ability to sabotage and disrupt; the hacker can stop the service or even take complete control of the system or view important files that may be confidential in operating systems (OS) and software [1]. Vulnerabilities have a high-scale impact and cause economic damage and problems to individuals, governments, and companies.

The Windows system is one of the most popular OS used for computers, so it is the main target for most hackers. It is important that vulnerabilities are discovered periodically and fixed before they are discovered by the hacker. The action of protecting information and communication technology (ICT) systems from various cyber-threats or hackers has come to be known as ‘cybersecurity’, Numerous sides are linked with cybersecurity: Measurement to protect ICT, The raw data and information it includes and their transmitting and processing, The system's virtual and physical components, the level of protection achieved through their implementation, and ultimately the related professional field.

The main concern in cybersecurity is comprehending various cyber-attacks and creating defense strategies that safeguard multiple properties. [2], In general, three basic factors must be present for data to be protected, which are confidentiality, availability, and integrity, and it is called the CIA triangle [3].

Cybersecurity is a determination of procedures, technologies, and programs that aim to protect networks, software, systems, data, and hardware from illegal access and malicious hackers [4]. Due to technical advancements, there has been an increase in vulnerabilities, attacks, and unauthorized access. Therefore, to meet the increasing

demand for cybersecurity solutions, it is crucial to develop procedures to enhance security via the improvement of Machine Learning (ML) capabilities. It is beneficial to use ML techniques to enhance cybersecurity [5].

ML is a field of artificial intelligence (AI) [6], it is already a significant factor in the development of current and future information incorporate significant proportion of this use is being generated in other areas of work that incorporate ML. Despite this, the utilization of ML in cybersecurity is still in its early stages, highlighting a significant divergence between research and practice. The fundamental reason for the discrepancy is the current state of the art, which hinders our perception of the task of ML in cybersecurity. ML will always be fully realized without the knowledge of a widespread understanding of its implications and advantages, as the methods employed at the time do not fully benefit the whole system [7] [2].
The dataset in ML is important for obtaining good results, so we were keen to work on a good dataset that represents real vulnerabilities in Windows. We collected the data set through two of the most famous global vulnerability data sites. We focused on selecting the relevant features, directly determining the severity of the vulnerability. Therefore, we will try to use ML algorithms to classify vulnerabilities as a cybersecurity procedure to increase the protection of the Windows system.

1.1. MOTIVATIONS

Users of operating systems face many vulnerabilities that may lead to damage or malfunction in the system or may lead to malicious purposes such as espionage, monitoring, and modification of confidential files by hackers exploiting those vulnerabilities.
Cybersecurity can reduce these risks through ML, as early classification of vulnerabilities represents an important stage in controlling them and determining their severity level.

1.2. THESIS PROBLEM

Data security and information protection have long been an issue. Researchers are trying to develop effective solutions in this field using modern technologies [8]. Protecting data and information is extremely important, especially in the era of advanced digital communications, and with cyber threats increasing and developing rapidly. Researchers and experts need to strive to address these information security challenges and search for innovative and effective solutions to strive to protect data and create a safer digital environment.

Windows is one of the most famous and widely used operating systems in the world. Therefore, protecting the Windows operating system is one of the areas that still faces challenges. However, the numerous vulnerabilities present in this system make it vulnerable to network intrusions and attacks [9].

Modern technologies have opened new ways of understanding and guiding efforts to improve data security. Therefore, this study highlights the application of modern technology represented by machine learning algorithms in Windows system vulnerability classification and analysis.

We can identify the main research questions about the feasibility and effectiveness of using machine learning and artificial intelligence to classify vulnerabilities in the Windows operating system.

The best machine learning algorithm model is then selected to achieve the highest accuracy and efficiency when analyzing and classifying security vulnerabilities. This can help researchers determine about the top methods and technologies to enhance data security and reduce cyber threats.

This research aims to develop and improve methods and techniques for protecting Windows operating systems and improving data security in general.
Through the above, we can define the research questions:

- Should we use ML algorithms to solve Windows vulnerability classification problems?
- Does machine learning provide effective solutions in the field of data protection?
- What is the best machine learning algorithm model to solve this problem and achieve the highest accuracy?

1.3. AIMS OF THE STUDY

The aims of this study include:

- Provide a comprehensive review of relevant literature in the field of cybersecurity and artificial intelligence.
- Proposing innovative and effective solutions in the field of cybersecurity using artificial intelligence technologies.
- Demonstrate the ability of machine learning algorithms to classify vulnerabilities effectively and accurately in operating systems.

1.4. RELATED WORK

Vulnerabilities present significant challenges in terms of detection and prevention. Despite extensive research and application of ML models, current defense mechanisms need help to provide complete protection. Over the past period, many studies have used ML to detect vulnerabilities and malware.

On the other hand, Mandal, Dilek, and Ervan Kosesoy [10], developed methods to predict vulnerability and security in software source code by ML techniques. They used the OWASP Benchmark Test dataset as a testing source and used Java code to train multiple ML processor models. Feature extraction methods TFIDF and Doc2Vec from source code were used. Their study showed that using logistic analysis techniques, decision trees, and multiple inference networks can lead to prediction accuracy of up to 0.97.

Chernis, Boris, and Rakesh Verma [11] present how to discover software vulnerabilities using ML techniques in their study. They analyzed the errors using a C programming source and an ML classifier. They showed that intelligent use of simple features can achieve up to 64% accuracy, compared to 69% for clever use of complex features.

Admu, Omar, and Arfan Awan [12] studied ML algorithms to detect ransomware. The study presents an analysis using the support interval algorithm for detection with an accuracy of 88.2%. They show that most other ML classifiers fail to detect ransomware as effectively as Interval Support does, which shows high effectiveness in classifying ransomware.

Ying Xu [13] mainly addressed ML methods. Developed a method that uses a compound integrative neural network to excerpt features of vulnerability by data pre-processing related to public vulnerability. The study showed that this method has high accuracy in training, recall rate, accuracy, F1 values, and Matthew relation coefficient, which is better than those used by complex integrative neural networks with interval support and other detection methods.

Munonye, K., & Péter, M. [14]. This research utilizes OAuth protocol representation as a learning problem that requires seven classification models to be developed, tuned, and evaluated. Explanatory Data Analytics (EDA) techniques have been applied to extract and analyze OAuth-specific features, allowing the impact of specific features identified in OAuth to be evaluated for each class of deliverables. This research involved training and tuning; over 90% accuracy was achieved in detecting vulnerabilities in OAuth certification and permission processes. Comparability with known vulnerabilities also showed a 54% match. We can see a comparison between similar studies in Table 1.1.

Table 1 1. Comparison between related work.

Authors	Objective	Dataset	ML Techniques	Key Findings	Accuracy
Islam, Rejwana, et al.	Android malware classification using ML techniques	CCCS-CIC-AndMal-2020	RF, Nearest Neighbors, Decision Trees, DBMs, Logistic Analysis	Achieved 95.0% in malware detection using dynamic traits. Model maintained high accuracy with feature exclusion.	95.0%
Mandal, Dilek, et al.	Predicting vulnerabilities in software source code using ML	OWASP Benchmark Test dataset	Logistic Analysis, Decision Trees, Multiple Inference Networks	Prediction accuracy of up to 0.97 in software vulnerability detection using TFIDF and Doc2Vec feature extraction methods from source code.	0.97%
Chernis, Boris, et al.	Discovering software vulnerabilities using ML techniques	C programming source	ML Classifier	Achieved up to 64% accuracy using intelligent use of simple features compared to 69% with clever use of complex features.	64%
Admu, Omar, et al.	ML algorithms to detect ransomware	Not specified	Support Interval Algorithm	Achieved 88.2% accuracy in ransomware detection, outperforming other ML classifiers.	88.2%
Ying Xu	Using compound integrative neural network (CNN) + LSTM to extract features of vulnerability	Public vulnerability data pre-processing	Compound Integrative Neural Network (CNN) + Long Short-Term Memory (LSTM)	High accuracy in training, recall rate, accuracy, F1 values, and Matthew relation coefficient compared to other methods.	88%

Munonye, K., & Péter, M.	Utilizing OAuth protocol representation as a learning problem	Not specified	Seven Classification Models, Explanatory Data Analytics (EDA) techniques	Achieved over 90% accuracy in detecting vulnerabilities in OAuth certification and permission processes. 54% match with known vulnerabilities	90%

PART 2

THEORETICAL BACKGROUND

This section will discuss the standard used to detect vulnerabilities in the Windows system. It provides an overview of ML technology and its role in cybersecurity. Next, we will verify the model metrics and testing and present a design model procedure for vulnerability classification.

2.1. CYBERSECURITY

In recent decades, there has been a great development in the field of technology and industries. Technology has spread everywhere and is closely linked to our lives, and therefore, interest in the issue of protecting systems and technology from cyber-attacks has become a subject of great interest.

Protecting information and communication technology (ICT) systems from various cyber-threats or hackers has become known as 'cybersecurity'. Numerous sides are linked with cybersecurity: measurement to protect ICT; the raw data and information it includes and their transmitting and processing; The system's virtual and physical components, the level of protection achieved through their implementation, and ultimately the related professional field. The main concern in cybersecurity is comprehending various cyber-attacks and creating defense strategies that safeguard multiple properties [2].

In general, three basic factors must be present for data to be protected, as Figure 2.1 shows, which are confidentiality, availability, and integrity, and it is called the CIA triangle [3].

- **Confidentiality** means preventing unauthorized persons from entering the system and preventing access to information.
- **Availability** means ensuring that information assets and systems have timely and reliable access to authorized persons.
- **Integrity** means preventing any unauthorized modification or destruction of information or data.

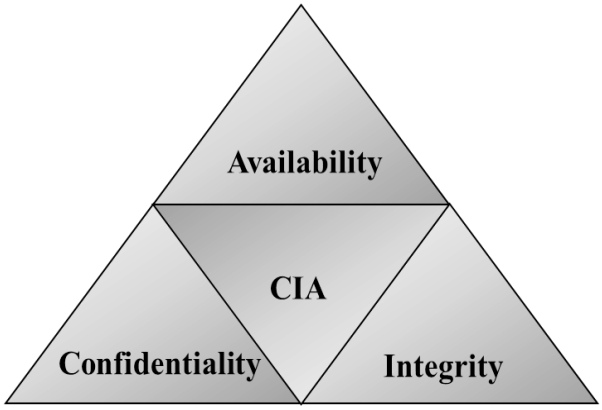

Figure 2 1. CIA triangle for security.

2.1.1. History of Cyber Security and Crime

The history of cybercrime and cybersecurity is a complex journey that reflects technological development and its impact on security. Cybercrime has its roots in the 1940s when the first computers appeared, a period free of computer crime. In the 1950s, we saw the rise of telephone hacking as individuals attempted to hack telephone system protocols to achieve free calling or reduce long-distance call charges [15].

In the 1960s, terms such as "hacking" and the disclosure of security vulnerabilities in computer systems emerged. We also saw ARPANET move to the Internet in the 1980s. During the 1970s, the ARPANET project, the first pre-Internet data exchange system, was founded. In 1971, the first "crawler" computer virus was created that could navigate the ARPANET [16] [17].

In the 1980s, computer viruses and the term "cyber-espionage" began to be used as a result of threats from other governments. In 1985, the US Department of Defense (DoD) approved computer security guide called Trusted Computer System Evaluation Criteria (TCSEC), commonly known as the Orange Book [18] [19].

In the current decade, we have seen cybercrime evolve into an advanced industry that reflects the accelerating transformations and impact of persistent digital threats. New technologies like ransomware and advanced persistent threats (APTs) have emerged. The current decade has witnessed an increase in the number of security breaches, with many high-profile companies and organizations affected. Social engineering techniques and phishing attacks have been increasingly used as the world becomes more digital [15].

2.2. VULNERABILITIES

A vulnerability can be defined as the existence of a vulnerability, design, or implementation flaw that could result in one or more unexpected and undesirable events that compromise the security of a computer system, application, protocol, or network [20]. An attacker can breach the information security of a system by exploiting a vulnerability, which is a weakness in general [20].

2.2.1. Vulnerability Databases

One of the priorities of all organizations is to maintain the security of their system and preserve their confidential information. Because threats are increasingly growing, it is necessary to deal with program vulnerabilities periodically. Most organizations use reliable global databases to view the latest threats, and among these databases is the National Vulnerability Data (NVD) [21]. This database contains lists of security vulnerabilities and contains details for each vulnerability, such as Common Exposures (CVE) or naming systems, which is a system that gives a special identifier for each vulnerability, and information such as trends in new vulnerabilities according to severity, exploitability, and CVSS scores [22] [23].

Many national and international organizations worldwide contribute significantly to collecting and managing information related to discovered vulnerabilities. Each new vulnerability is usually assigned a unique identifier in the Vulnerability Database (VDB), and a detailed description of the detected problem is provided. Unlike databases restricted to a specific vendor or product, these repositories, as noted in reference [24], have broad coverage [25].

The Common Vulnerabilities and Exposures (CVE) database stands out among the vast, popular, and trusted databases. MITRE has overseen maintaining the CVE List since 1999. The identification number is assigned, a description is provided, and links are provided to a security vulnerability that is publicly known, as described in [25].

NVD is one of the most popular open vulnerability databases and provides a comprehensive vulnerability reporting system. The system allows whistleblowers to report new vulnerabilities using the CVE-ID 2 (Combined Vulnerability and Exposure Identifier) assigned to the vulnerability in the NVD database by providing a short text summary with a title and short description. This approach enables the community to effectively share security information as the system ensures easy reporting and verification of vulnerabilities, which helps improve digital security and better understanding current of security challenges [26].

Information about CVE vulnerabilities was calmed by the US NVD, which was launched by the National Institute of Standards and Technology (NIST) in 2005. CVE and NVD are supported by the US Cybersecurity and Infrastructure Agency (CISA) and are free and open to the public [27].

The daily disclosure of many security vulnerabilities highlights the importance of these rules. As part of its comprehensive analysis, NVD augments each record with additional information, the Common Vulnerability Scoring System (CVSS), which is developed on a group of measurement and subsidizing the CVSS v2.0 [28] and v3.X [29][30] standards.

2.3. MACHINE LEARNING

Over recent years, ML and AI have gained new importance, driven by ever-increasing amounts of data and computing power, as well as the discovery of improved learning algorithms [31].

ML is the algorithms and statistical utilized by PC structures to carry out obligations without direct programming. Learning algorithms are used in many programs that we use each day. ML is used to train machines to address records extra efficiently. Sometimes, after reading the data, we can no longer interpret the information acquired from it; in that case, we observe device learning. The ML algorithms are used for diverse functions like statistics mining, photograph processing, and predictive analytics. The important benefit of the system getting to know is that it can do its paintings automatically when a set of rules learns what to do with data [31].

2.3.1. Machine learning type

The prospects for ML are interesting in our modern era, as this field contributes to the development of intelligent systems capable of understanding and processing data effectively. ML is a paradigm shift in computing, allowing systems to gain experience and improve over time without direct human intervention. ML is divided into three basic types, as shown in Figure 2.2.

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning [32].

These diverse ML types enable multiple applications and uses, such as prediction, classification, natural language processing, computer vision, smart games, and smart robots. These ongoing innovations represent increasing technological and artificial intelligence development challenges and opportunities.

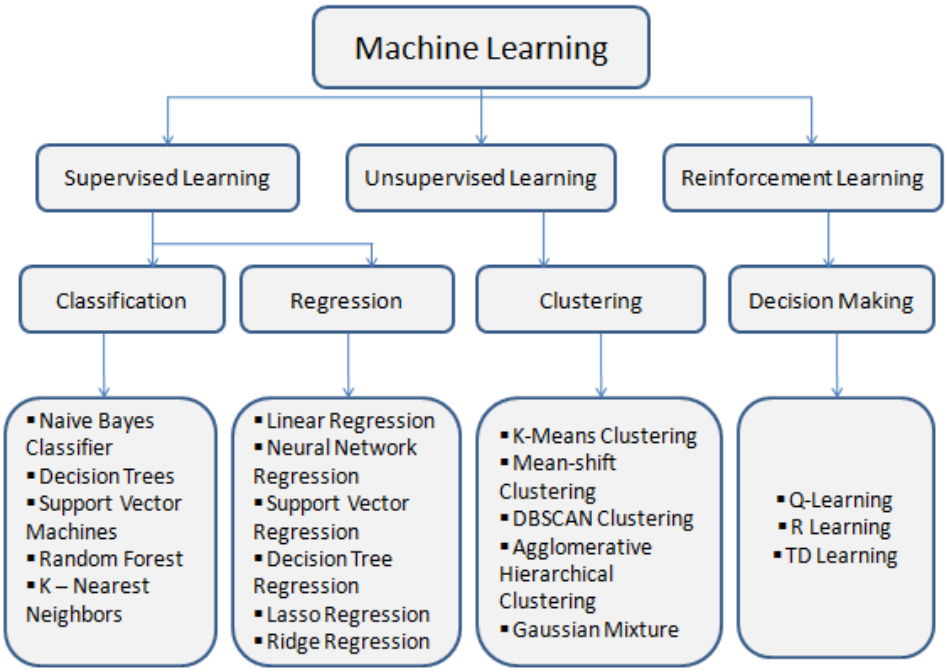

Figure 2. 2. Types of Machine Learning [32].

2.4. MACHINE LEARNING IN CYBERSECURITY

ML technology is becoming an essential tool in cybersecurity. With the proliferation of small personal devices such as smartphones and massive services such as cloud computing and online banking, a huge amount of data is created, exchanged, and processed to achieve results in specific applications. As a result, data and device security and user privacy in cyberspace have become a critical concern for individuals, businesses, and governments [33].

However, algorithms used in machine learning may be vulnerable to attacks during the training and testing phases, leading to performance degradation and increased instances of security breaches. To avoid these problems, a comprehensive review of research and work conducted from 2013 to 2018 on using machine learning in cybersecurity was conducted.

This survey reviews the basics of cyber-attacks and defenses, as well as common machine learning algorithms and proposed feature extraction and data analysis schemes to enhance cyber security, using dimensionality reduction features and classification and detection techniques. The review covers ML concepts in cybersecurity, with an emphasis on the use of algorithms, feature extraction schemes, and output charting. In addition, it provides a complete overview of competitive machine learning, including the security real estate of deep learning methods. Also reviews publicly available databases within computer security research and discusses current issues and future research directions [34].

In summary, developing the security field using machine learning is a critical field that requires further exploration and research. This field has a major role in better understanding the weaknesses of machine learning techniques against security threats and defense devices to enhance public security in cyberspace, which is considered one of the biggest problems of the era. A comprehensive study of recent works on cybersecurity using machine learning is a valuable and informative resource for researchers and engineers seeking to enhance cybersecurity and develop solutions in data security [35].

The advanced analytical tools machine learning provides are essential in meeting new cybersecurity challenges by detecting and mitigating cyber threats. The importance of this role depends on their ability to analyze and classify large amounts of data, extract patterns and adapt to new threats. The importance of machine learning in solving cybersecurity problems is explained in detail below:

- **Anomaly and Threat Detection:** Machine learning algorithms help detect anomalies through analysis that represent potential vulnerabilities in network traffic, user behavior, and system logs. These algorithms can detect deviations from normal patterns and generate alerts based on historical data for further investigation [36].
- **ML Vulnerability Classification:** ML techniques can also improve the vulnerability classification process. When analyzing logs and data related to network vulnerabilities, the system uses machine learning models to prioritize

and classify vulnerabilities by severity. This helps to increase the effectiveness of vulnerability patches, focus more on resolving high-priority vulnerabilities, and further strengthen network defense [37].

- **Phishing and Malware Detection:** ML techniques were used to identify phishing emails and classify fake messages. It detected malicious URLs and classified malware and it succeeded [38]. ML models can differentiate between legitimate and malicious entities by analyzing email content, website characteristics, and file attributes [39][40].
- **Real-Time Response:** ML models can facilitate real-time response to cyber threats by automating decision-making processes, such as dynamically adjusting access controls, blocking suspicious activity, and initiating response actions based on identified threats [41].
- **Behavioral Analysis:** Through machine learning, behavioral analysis tools can be developed, which help in understanding and analyzing threats and unauthorized access attempts by classifying user activities and detecting deviations from normal behavior, and this has a significant role in developing the security field [42].
- **Adaptable Security Measures:** Through continuous learning and periodic data updating, machine learning can adapt to changing threat environments, contributing to the detection and classification of cyber threats [43].
- **Reducing False Positive Results:** Using ML can reduce incorrect or false alerts in cybersecurity systems and improve the efficiency of security operations because machine learning is characterized by high accuracy in results, and the possibility of error is low [44].
- **Optimizing Resources:** Improving the allocation of machine learning resources helps in data security to focus on essential and high-priority threats and make appropriate decisions [45].
- **Continuous Improvement:** Machine learning models are also characterized by their ability to be improved and developed over time, which enhances accuracy and effectiveness in identifying and mitigating cybersecurity risks [46].

Through the positive points we mentioned in this data, it is possible to know the crucial role of machine learning in data security, confront challenges, and strengthen defense against cyber-attacks and various vulnerabilities.

2.5. CHALLENGES AND OPPORTUNITIES

- **Limited evaluation scenarios:** One of the most prominent challenges facing learning in data security is the need for evaluable scenarios. This makes it difficult to evaluate the effectiveness of ML algorithms in detecting threats in ICS (Industrial control system) environments, as it is considered essential to evaluate the performance of machine learning on actual, pre-existing scenarios [47].
- **Relying on labeled data sets:** Another challenge is data classification. In the context of ICS security, relying on labeled data sets is complex, raising concerns about data availability. This reliance limits the possibility of applying supervised learning approaches and data representation for attack detection models [48].
- **Failure to consider practical implementation:** Focusing on practical aspects and building systems and models for deep learning alone without paying attention or considering the theoretical aspect constitutes a disconnect between theoretical performance and actual application in various ICS environments [47].
- **Inadequate performance measures:** The absence of standardized performance measures is a challenge, hampered by the lack of comprehensive evaluation criteria, including time-based measurements. This hinders the assessment of the suitability of ML algorithms for real-time detection in ICS environments [49] [50].

2.6. DISPOSITION

- **Diversity of attack scenarios to be evaluated:** Collaborating on creating comprehensive datasets can contribute to the practical evaluation of machine

learning algorithms, as diverse and realistic attack scenarios can be developed accurately to reflect potential threats in ICS [51].

- **Exploring unsupervised and semi-supervised learning:** ML algorithms that leverage unlabeled data must be developed to reduce reliance on labeled datasets and enhance the adaptability of models to threats in ICS [48].
- **Incorporating practical considerations into model development:** Addressing the deficiency in practical implementation considerations requires incorporating real-world constraints into developing of attack detection models for industrial control systems.
- **Unifying comprehensive performance metrics:** Comprehensive evaluation criteria should be standardized to include accuracy and time-based metrics, enabling effective evaluation of real-time ML algorithms in ICS environments [49] [50].

PART 3

METHODOLOGY

3.1. OVERVIEW

Vulnerabilities cause much damage to systems, especially if they are discovered early by hackers without the knowledge of the system user. Therefore, weak points must be detected and classified early, as their seriousness can be reduced if they are classified and treated in advance. In this study, we proposed a cybersecurity procedure that classifies vulnerabilities using ML algorithms. ML helps classify the severity of vulnerabilities.

3.2. THE PROPOSED WORK

To prepare a model in ML, the first step is to select an appropriate dataset to achieve accurate results. In this case, we have gathered data on vulnerabilities in Windows from the past six years. We focused on making the data real. Based on real data and actual vulnerabilities, we collected 12 characteristics for each vulnerability. Then, the most important characteristics directly related to the strength of the weaknesses were selected. The data was then classified into 4 groups according to the strength of the risk for each security vulnerability. After selecting the appropriate data set, it is the role of choosing the appropriate algorithm. Since our problem is classifying weak points, we chose the 5 most popular algorithms for solving classification problems, where we tested the representation of five ML algorithms to reach the highest possible accuracy.

The methodology that was followed in the research includes several important steps to achieve the research objectives, which are as follows:

- **Data selection and collection:** The research began by selecting an appropriate data set to achieve the research objectives: vulnerabilities in the Windows operating system over the past six years. Data on these points has been collected from reliable and well-documented sources to ensure data accuracy.
- **Data analysis and selection of important features:** The collected data was analyzed, and the importance of each feature for vulnerability classification was evaluated. Based on this analysis, 12 key features were selected for each gap.
- **Data Classification:** After collecting the data and identifying important characteristics, the data was classified into 4 groups based on the severity of each vulnerability. This classification helps understand priorities and focus on areas of high importance.
- **Choosing the appropriate algorithm:** After classifying the data, appropriate algorithms were selected to solve the classification problem. Five algorithms were selected among the most popular in machine learning and considered suitable for dealing with the problem of vulnerability classification.
- **Algorithm testing and performance improvement:** The selected algorithms were tested using the collected data, and the performance of each algorithm was estimated. Performance is optimized using different data representations and comparing the results to reach the highest possible accuracy. The steps of work illustrated in Figure 3.1.

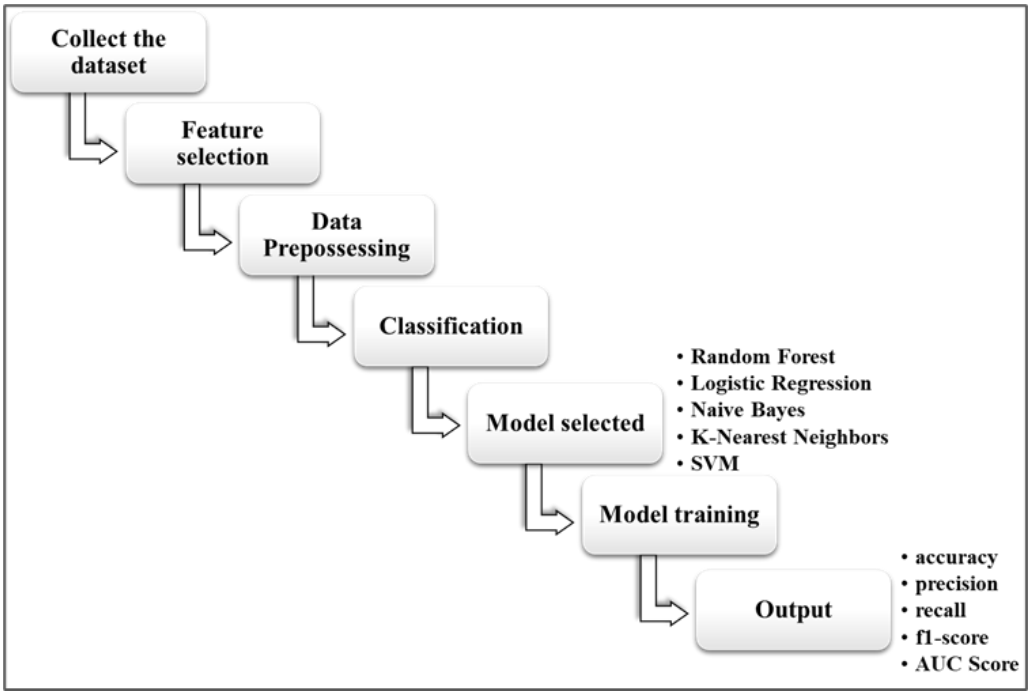

Figure 3 1. Steps of works.

3.3. TECHNICAL REQUIREMENTS

This study was implemented on a Windows 10 (64-bit) laptop, which contains a computer processor (CPU) and chipset 2.60 GHz Intel(R) Core (TM) i7-10750H. Intel(R) Core (TM) i7-7600U CPU @ 2.80GHz - 2.90GHz, with 16GB of RAM. Spider program is a development environment that uses the Python programming language.

3.4. DATASET

3.4.1. Dataset Collection

Vulnerability data is concentrated through CVE identifiers. Each vulnerability is accompanied by a summary that explains the affected software, the impact's nature, and the importance of specific scripts. These vulnerabilities are collected in an NVD, loaded with additional useful information to understand the why better [52].

In this study, we collected our dataset from the exploit-deb website [53] and also depended on NIST to take some data features; considering the period between 2018 and 2023, At this time, we have collected (190) vulnerabilities on the Windows OS in CSV file.

3.4.2. Dataset Details

The dataset has 12 features, each with a different value and data type, as Table 3.1 shows us.

Table 3 1: Features details.

Features	Description	Type
EDB_ID	This is the ID of vulnerability in ExploitDB website	Integer
CVE	The Common Vulnerabilities and Exposures (CVE) is a general identifier for vulnerabilities.	Integer
CVSS	Common Vulnerability Scoring System (CVSS) or a custom severity scale is a general measure of the strength of a vulnerability.	Integer
Title	The title of vulnerability	String
EDB_Verified	This Feature indicates whether the vulnerability has been verified, exploited, or tested. Column has two values (0,1)	Integer
Type	The type of vulnerability (remote, local, Dos, etc.) and the categorizing vulnerabilities by type can be useful for analysis.	String
Platform	The platform means in which platform this vulnerability can work (Windows 10, Windows 11, etc.).	String
Date	The date of vulnerability was published.	Date
Version	The version of the vulnerability.	String
Author	The name of the author of this vulnerability.	String
Description	Describe of the vulnerability, such as security issues, and provide more details.	String
Source	The link of the vulnerability source	String

3.4.3. Feature Selection

The vulnerabilities have many features, but not all of these features directly affect the severity of a vulnerability, so we need to choose effective features that have a direct impact to classify vulnerabilities. We do this by using feature selection method; after performing the feature selection process, as shown in table 3.2.

Table 3 2: Feature selection for dataset.

Features	Description	Data type
EDB_ID	The ID used to differentiate between each of the vulnerabilities, as each vulnerability has a different ID	Int.
CVSS	We are classifying vulnerabilities into four classes based on their strength level, based on this value which ranges from 0 to 10.	Int.
Type	Represents the type of vulnerability	String
Platform	The platform on which the vulnerability runs.	String

3.4.4. Dataset Preprocessing

In this last step, where the attributes are selected, the text indicates that there are 4 attributes, and among them, 2 attributes have the data type "text" (genre, platform). It is noted that ML algorithms cannot directly deal with text data; rather, they must convert them to a digital representation before using them in learning processes.
One-hot encoding technology is used to convert text-type features into a digital representation. In this technique, an empty vector is used where one of the elements is set to the value 1, while all the others are set to the value 0. This creates a numerical representation of text data, which is commonly used to represent texts that contain a specific set of values [54] [55].

The text notes that although hot coding results in high-dimensional feature vectors when cardinality is high (a large number of possible values), it remains popular and widely used because of its simplicity. It is also points out that this technique is mainly used in neural network models that require the input to be represented exponentially in the range of [0,1] or [-1,1].

The text also explains the concept of a Uniform Vector, which is a matrix with one row and several columns, where all values in the row are zero except for one cell containing the number 1 and is used to identify a word uniquely.
In this study a set of words such as: [DOS, LOCAL, REMOTE, WEBAPPS] would be represented by 4 of these encodings [1, 0, 0, 0] as shown in Figure 3.2

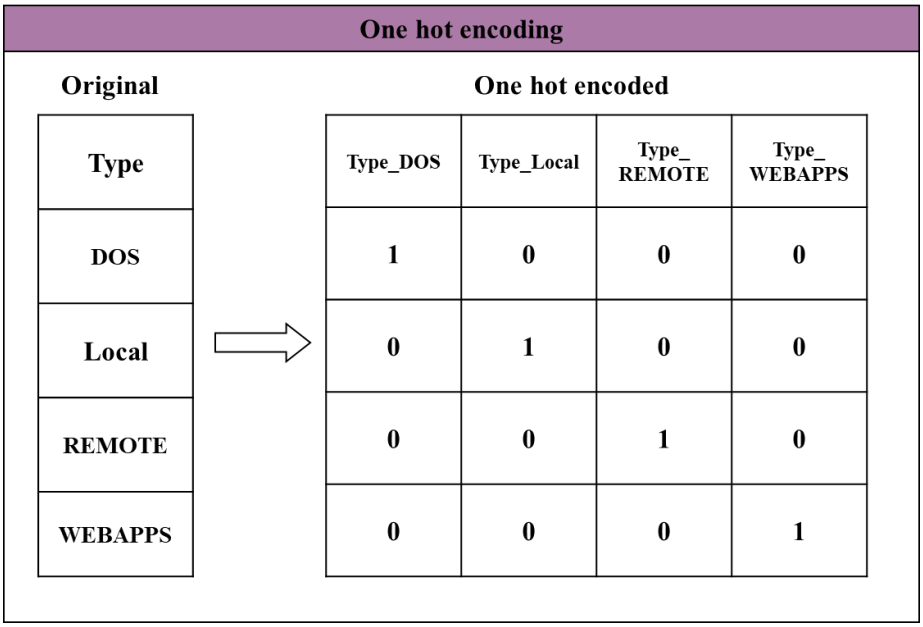

Figure 3 2. One hot encoding technique.

3.4.5 Classification

We classify by adding a class column to the dataset file. This column contains four values (-1,0,1,2), "-1" means that it is low risk, "0" means that the vulnerability risk is medium, "1" means that it is high risk, and "2" means the risk is critical based on CVSS values [56]. Figure 3.4 show as classes category.

We have four different classes of vulnerabilities based on the risk level (low, medium, high, critical). In Figure 3.3 we can see the addition process.

Classification operation

Original

Type_DOS	Type_Local	Type_ REMOTE	Type_ WEBAPPS	CVSS
1	0	0	0	3
0	1	0	0	4.4
0	0	1	0	7.5
0	0	0	1	9.8

Add class column

Class
-1
0
1
2

Figure 3 3. Classification opration by add class column.

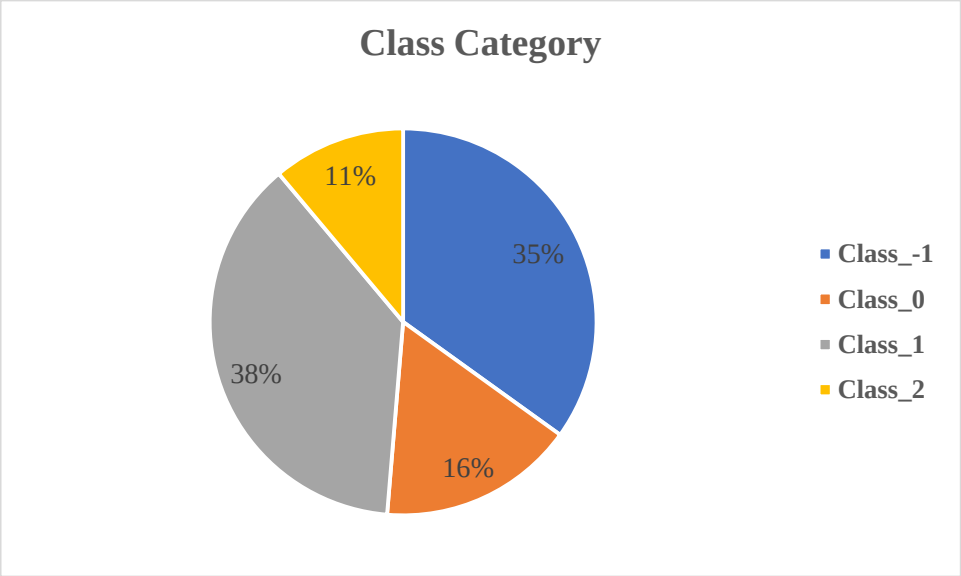

Figure 3 4. Classes category.

3.5. MODELS TRAINING

We prepared the data then chose five different algorithms to train the model and obtain the results. (Random Forest (RF), Logistic Regression, Naive Bayes, K-Nearest Neighbors, Support Vector Machines (SVM)). These algorithms are among the most popular algorithms for solving classification problems in ML. So we tested

the performance accuracy of each algorithm separately to find out which gives the highest value in accuracy.

3.5.1. Random Forest

Random Forest is one of the most superior algorithms in the field of learning, having achieved great success in various fields. Random Forest's greatest strengths are its versatility and ability to handle classification and playback tasks. This power makes it capable of performing broad predictive analysis functions in the medical field. This algorithm particularly shines in medical diagnosis and disease prediction, excels at handling complex and relational data, and fortifies its strength against challenges such as overfitting [57].

As an example, email is an important means of communication and is considered an important part of business, education, entertainment and other fields in various countries. As we all know, everything has its advantages and disadvantages, and although it is beneficial to society, everyone has to deal with its disadvantages, and that includes spam. These contain links that attempt to compromise computer systems, steal data and fraudulently access information. Random Forest algorithms can be used to detect infected and unwanted emails and prevent them from reaching users' inboxes, thus helping to maintain security [58].

In the “Random Forest” standard as a reliable algorithm, it is considered one of the best ML algorithms for data classification. Achieving excellence in dealing with complex and dimensional data, identifying important features, and contributing to accurate predictions makes it a valuable asset to the diagnostic and predictive talent pool in security [59].

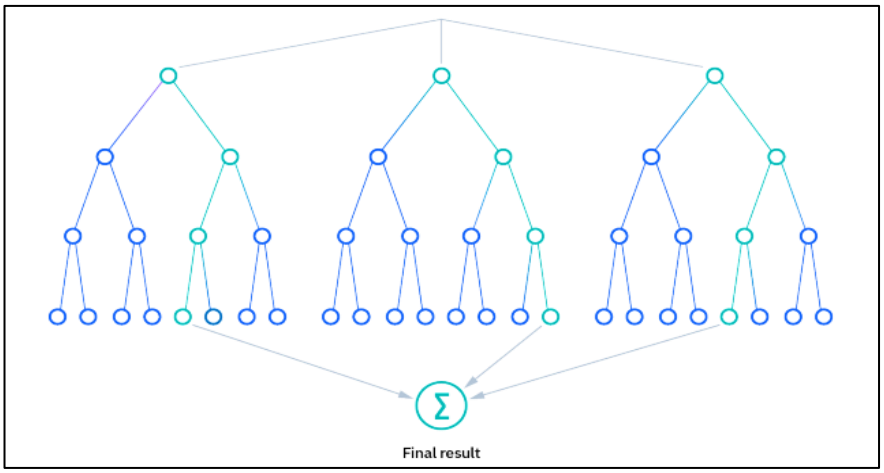

Figure 3 5. Random forest tree [60].

3.5.2. Logistic Regression

Logistic regression is a statistical classification technique that separates data categories. Logistic regression is very famous as an effective algorithm in solving classification challenges, and it finds wide applications in areas such as finance and natural language processing. It can also be used in the field of cybersecurity to classify attacks.

The logistic regression model is derived from the logistic function also called the sigmoid function and selects numerical values between 0 and 1. This design models the possibility of a specific category or event occurring. The formula defines the logistic function:

$$\sigma\,(z) = \frac{1}{1+ e^{-z}} \tag{3.1}$$

Where z is a linear coalition of input features and model criterion [60].

A logistic regression classifier with a weight threshold is used as part of the classification of flow systems, and this allows the automatic classification of flow systems according to specific criteria. In this approach, a classifier is trained using labeled data and flow characteristics to predict the flow regime. The classifier learns

the relationship between input features and the flow system and then uses this knowledge to classify new, unseen data [61].

Training a logistic regression classifier involves optimizing the model's parameters using techniques such as gradient descent or other optimization algorithms. This helps in classifying the data. Also, after training, the classifier can predict the flow regime of new data points based on their feature values [60].

Threshold weight logistic regression classification can be a valuable tool for automatic classification tasks, including vulnerability domain classification and cybersecurity, where it can have an effective role in vulnerability classification.

3.5.3. Naive Bayes

Naive Bayes is a simple technique for building classifiers. The model assigns class labels to problem instances, represented as a vector of feature values, where the class labels are drawn from a finite set. There is no single algorithm for training such classifiers; rather, a group of algorithms is based on a common principle. All Naive Bayes classifiers assume that the value of a given feature is independent of the value of another feature given a class variable. For example, if the fruit is red, round, and about 10 centimeters in diameter, it can be considered an apple. The Naive Bayes classifier considers that regardless of any correlations between color, roundness, and diameter features, each of these features independently affects the probability that a fruit is an apple [62].

The advantage of the Naive Bayes classifier is its ability to handle multiple input variables and explore probabilistic relationships between them quickly and efficiently. In many practical applications, the parameters of Naive Bayes models are estimated using the maximum likelihood method; In other words, one can use Naive Bayes models without accepting Bayesian probabilities or using any Bayesian methods.

This classifier is widely used in fields such as email classification and text sentiment analysis, solving the challenge of classifying large and complex data [63].

3.5.4. K-Nearest Neighbors

The KNN algorithm is a machine learning algorithm often used to classify data and is based on proximity, where points are shared with similar points in the training set. ANN belongs to the family of human algorithms and is characterized by its simplicity and efficiency [64].

How KNN works:

- Choosing a K value: determines the number of neighbors (K) whose proximity is checked [65].
- Distance calculation: The program calculates the distance between all points and all points in the training set using different distance metrics.
- Classification: Point K is chosen near the largest section, and the prime is over the majority at those points. For example, if more K points fall into class A, the fine point is classified as type K, as Figure 3.6 shows how the data is classified according to KNN [66].

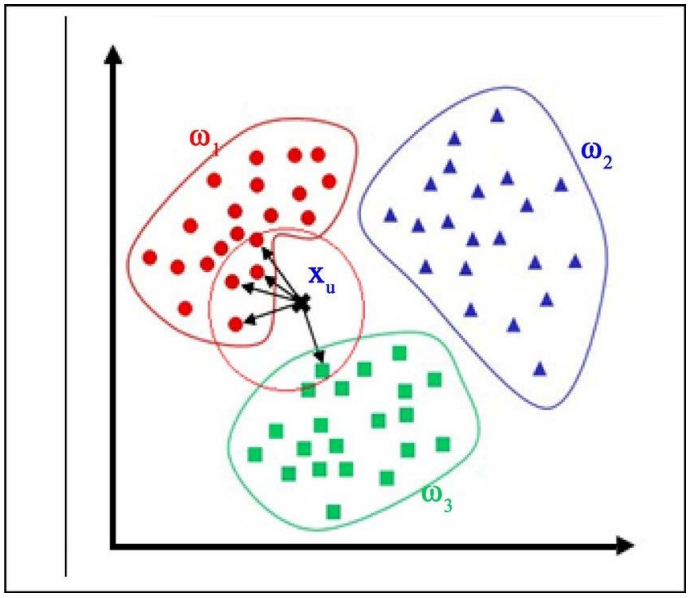

Figure 3 6. K-NN algorithm classification [67].

3.5.5. Support Vector Machines (SVM)

There is only one super-support algorithm (SVM), perhaps the most elegant one in the classification field. Each object in space classified is represented as a point in 2D, where influences affecting that point are usually specified. An SVM performs the classification task by drawing up to a break a 2D line or 3D plane that collects all the points of one class on one side and all the points of the other class on the other side.

Try SVM to find the best dividing line separating the two classes, maximizing the distance to the points in each class; this distance is known as the margin. Points that lie exactly on the edge of the margin excluding what remains of support. SVM requires a training set or set of points that have been correctly labeled. Therefore, the strength is based on saying that SVM is an algorithm that helps supervision [67].

In the background, the SVM solves a convex enabling problem that maximizes the margin and sets the constraints that points in each class fall on the right side of the interval. SVM is easy to understand, interpretable, and non-controversial [67].

Can compare the algorithms in Table 3.3. We prepared this table to comparison between the algorithms.

Table 3 3. Comparison of the algorithms.

Criteria	Random Forest (RF)	Logistic Regression	Naive Bayes	K-Nearest Neighbors (KNN)	Support Vector Machines
Type	Ensemble (Bagging)	Linear Model	Probabilistic	Instance-based	Distinction
Learning Style	Supervised	Supervised	Supervised	Supervised	Supervised
Use Case	Classification, Regression	Binary Classification Multiclass	Classification	Classification, Regression	Classification, Regression
Handling of Data	Ensemble of Decision trees	Linear combination of features	Assumes conditional independence	Proximity-based	Hyperplane-based
Nature of Output	Class Prediction	Probability	Class probabilities	Class Prediction	Class Prediction
Handling of Outliers and Noise	strong to overfitting handles missing data	Sensitive to outliers	handles missing data	Sensitive to noise and irrelevant features	Effective in high-dimensional spaces
Computational Complexity	Generally higher (multiple trees)	Generally lower (simple calculations)	Generally lower (simple calculations)	Computationally intensive (stores all training instances)	Computationally intensive (depends on kernel function)
Parameter Tuning	Moderate	Typically low	Typically low	Choice of 'k' and distance metric	Choice of kernel, C, gamma, etc.

PART 4

RESULTS

4.1. ACCURACY METRICS

As a result, we measured the scores of accuracy, precision, recall, F1-score, and ROC AUC score and plotted the Convolution matrix. The results showed that the highest accuracy in using the algorithms was in the RF algorithm, which reached 0.97% compared to the rest.

These performance metrics are usually used in classification tasks to evaluate the performance of a model. First, we have four terms:

- **True Positive (TP):** Instances that are actually positive and are correctly predicted as positive.
- **True Negative (TN):** Instances that are actually negative and are correctly predicted as negative.
- **False Positive (FP):** Instances that are actually negative but are incorrectly predicted as positive.
- **False Negative (FN):** Instances that are actually positive but are incorrectly predicted as negative.

To calculate the performance metrics, we own these equations [68] :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (4.4)$$

Accuracy measures the proportion of correctly classified instances (both positive and negative) out of total instances. Moreover it is better when there is a balanced dataset where classes are distributed equally, relying on True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) [69].

Precision measures the proportion of true positive predictions out of all positive predictions (correctness of positive predictions). Moreover it is better when focus is on reducing false positives, where each positive prediction needs to be accurate and relies on True Positives (TP), False Positives (FP) [69].

Recall measures the proportion of true positive predictions out of all actual positive instances (completeness of positive predictions), and it is better when the emphasis is on minimizing false negatives, ensuring that actual positives are correctly identified, relying on True Positives (TP), False Negatives (FN) [70].

F1 score is a harmonic mean of precision and recall, balancing between the two metrics. It is better when desire to balance precision and recall, especially in imbalanced datasets. and relies on True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN) [71].

Table 4 1. Result of performance metrics.

Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Roc Auc Score**
Random Forest	**0.97368**	**0.9737**	**0.9759**	**0.9738**	**0.9993**
Logistic Regression	0.8157	0.8158	0.8285	0.7987	0.9697
Naive Bayes	0.6843	0.6842	0.5417	0.5942	0.8825
K-Nearest Neighbors	0.8948	0.8947	0.9049	0.8965	0.9658
Support Vector Machines	0.9473	0.9474	0.9555	0.9479	0.9622

4.2. CONVOLUTION MATRIX

The convolution operation occurs when a kernel-based filter is moved over an input image and the method of element-wise multiplication and summation is computed at each position. In a 2D convolution, using I, K, and O, create a convolution matrix

equation for the input image, convolution kernel, and output feature map, respectively [68]. The mathematical representation for this convolution operation is as follows:

$$O(i,j) = \sum_m \sum_n I\,(i+m, j+n).K(m,n) \tag{4.5}$$

O(i,j) is the value of O(i,j) when (i,j) is located on the output feature map. Summation over the filter/kernel dimensions is reported using m and n as indices for summation in these dimensions.

Positioned in the input image, the pixel value in the input image is I(i+m,j+n) while n represents the position (i+m,j+n), Position (m,n) of the convolution kernel, K(m,n) is obtained by placing the operator on the vector plane and finding for K(m,n) at position n(i) in the convolution kernel.

The summation is carried out by using all valid positions of the filter on the input image. The output feature map assigns the resulting sum to the position on it.

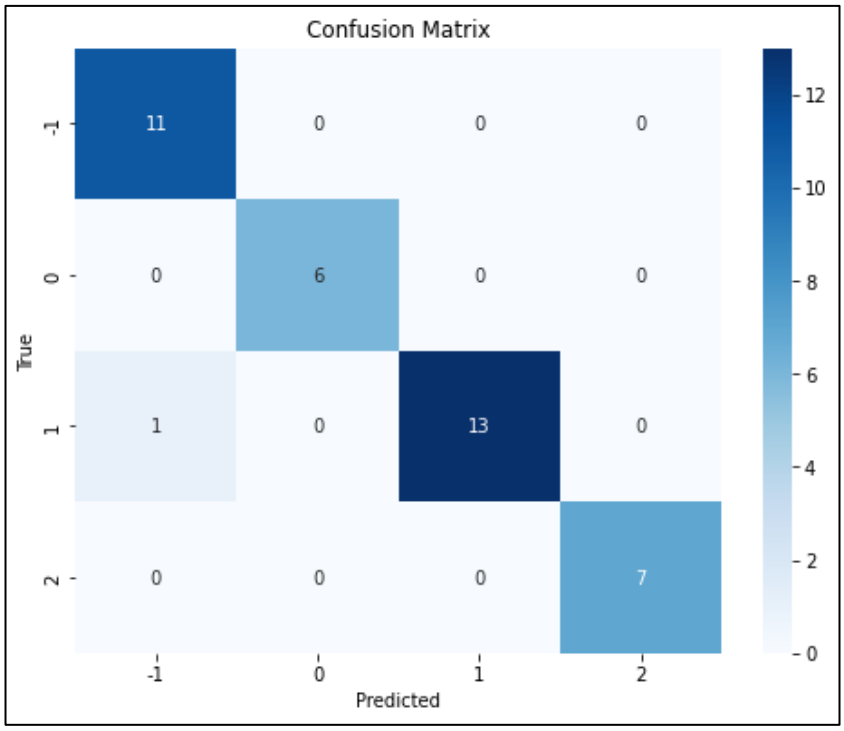

Figure 4 1. Confusion Matrix plan for Random Forest model.

The results of the matrix in our current work were as shown in figure 4.1, figure 4.2, figure 4.3, figure 4.4 and figure 4.5.

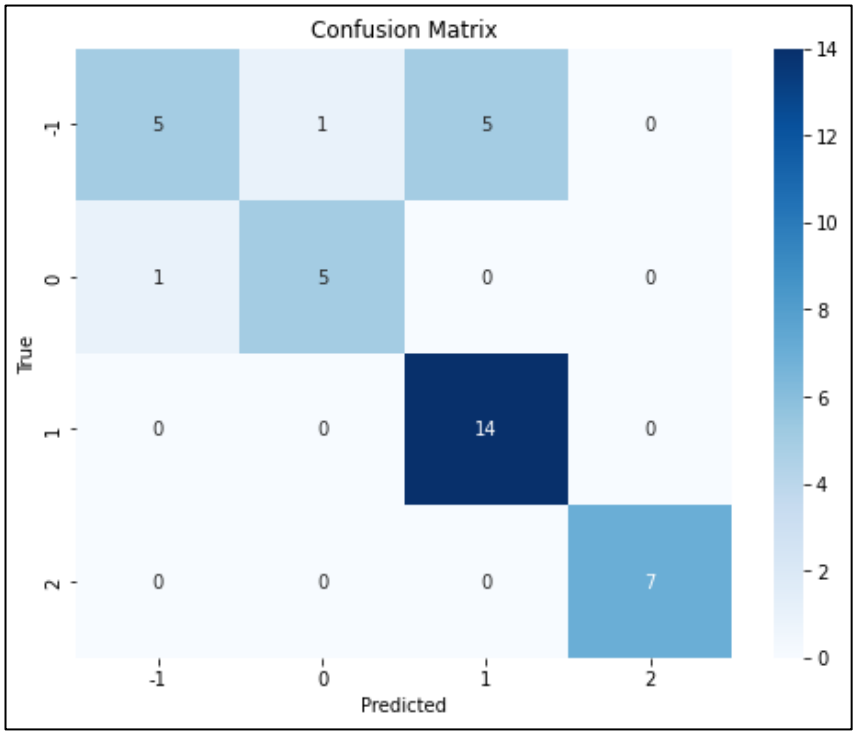

Figure 4 2. Confusion Matrix plan for Logistic Regression model.

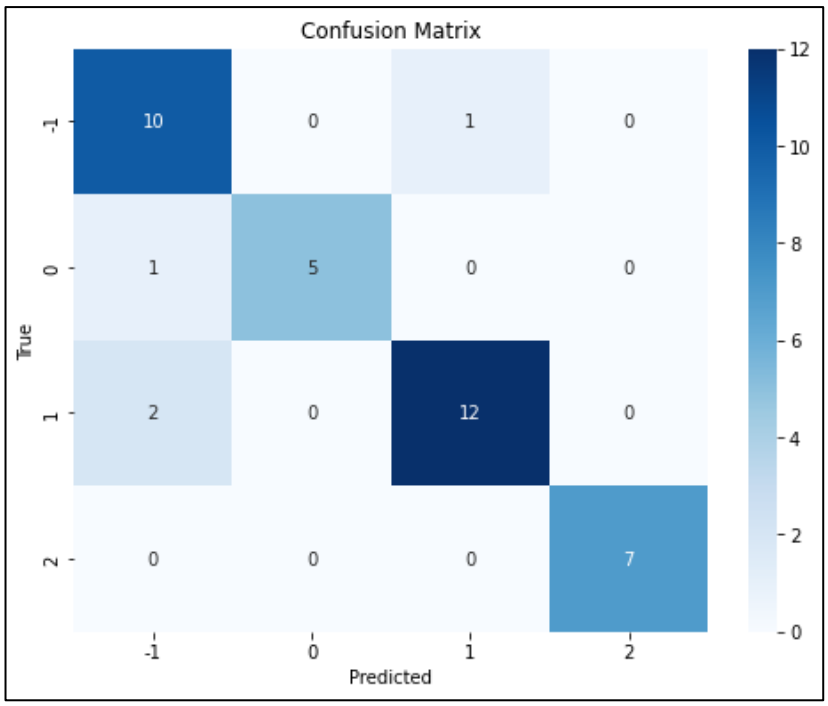

Figure 4 3. Confusion Matrix plan for K-Nearest Neighbors model.

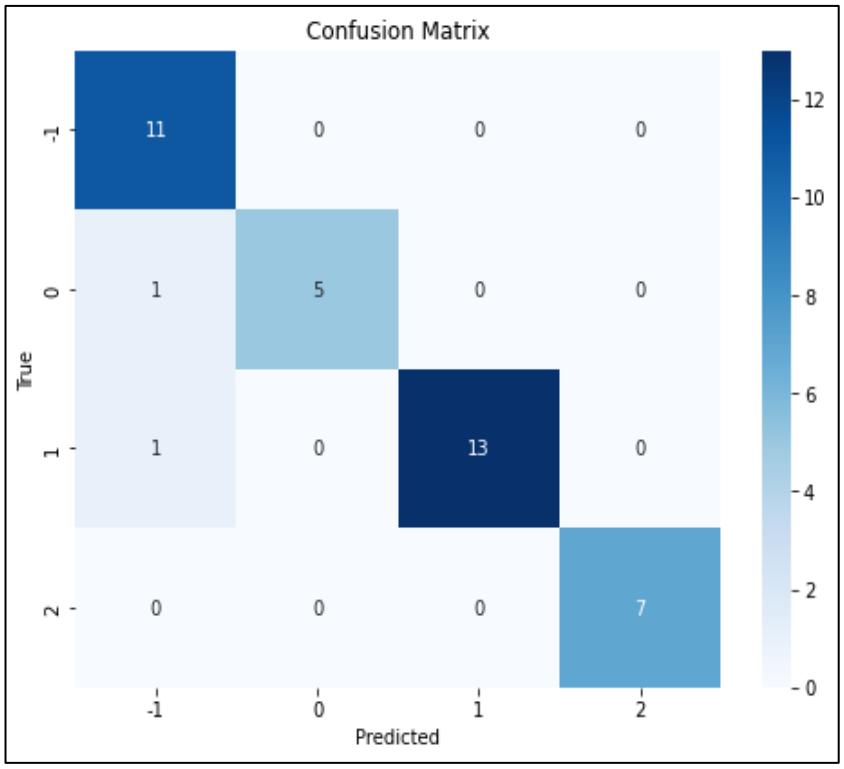

Figure 4 4. Confusion Matrix plan for Naive Bayes model.

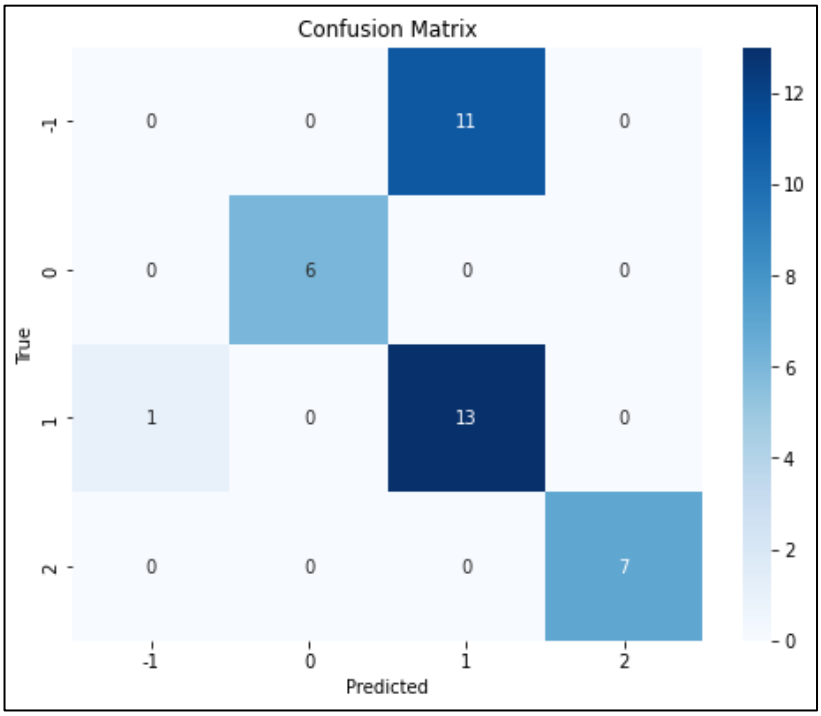

Figure 4 5. Confusion Matrix plan for SVM model.

4.3. ROC AUC

To begin, the receiver operating option ROC is generated by keeping track of the details and details for different gasoline blanks for the Naples values. This is done by creating a list of the different values of the test, as well as the post level and quality that confirm their value. Initially, ROC tactics are plotted in ROC models with the true (driver rate) on the y-axis and 1-specific (false driver rate) on the x-axis, where tabulated values are embodied [72]. Finally, the ROC colors are depicted as a graphical representation, using values tabulated for different taxa on the y-axis, and the ROC AUC results can be displayed in Figure 4.6, Figure 4.7, Figure 4.8, Figure 4.9, and Figure 4.10.

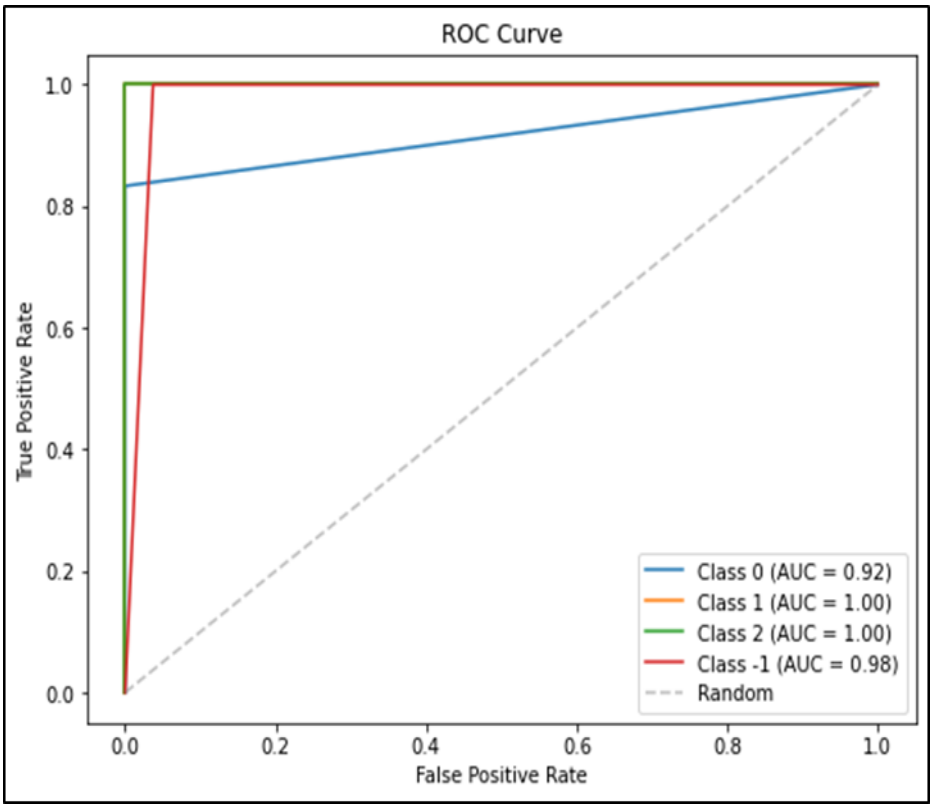

Figure 4 6. ROC AUC plan for Random Forest model.

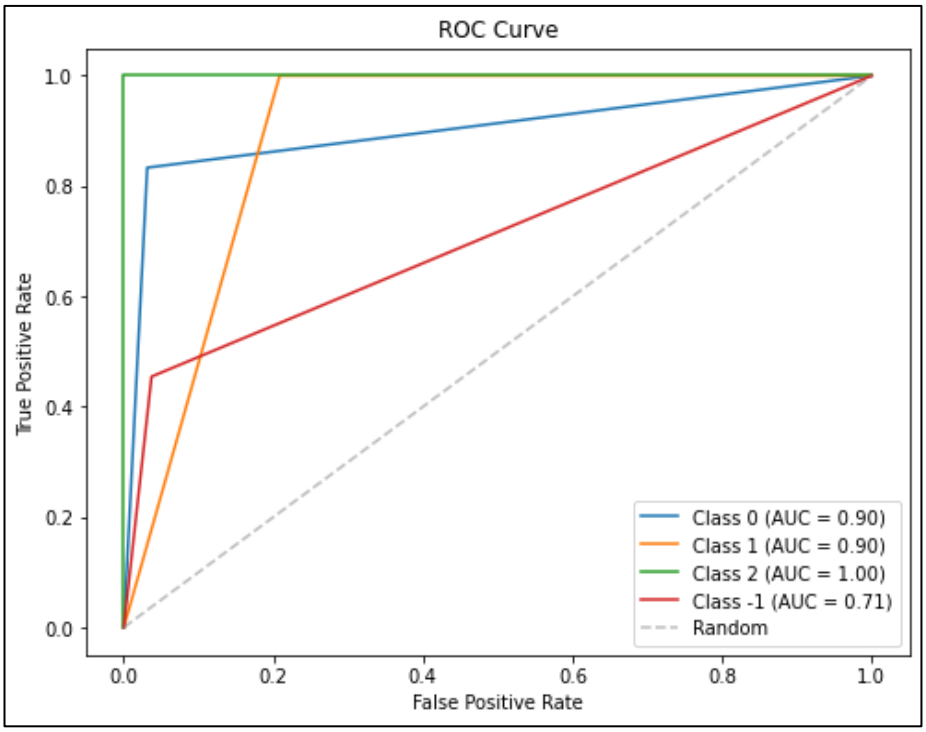

Figure 4 7. ROC AUC plan for Logistic Regression model.

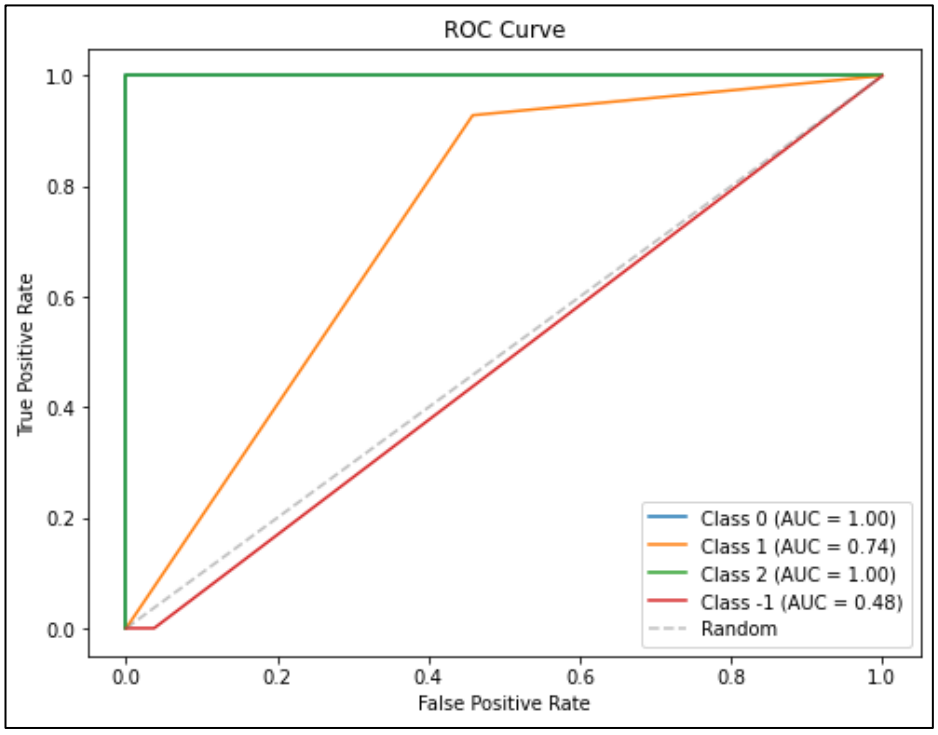

Figure 4 8. ROC AUC plan for Naive Bayes model.

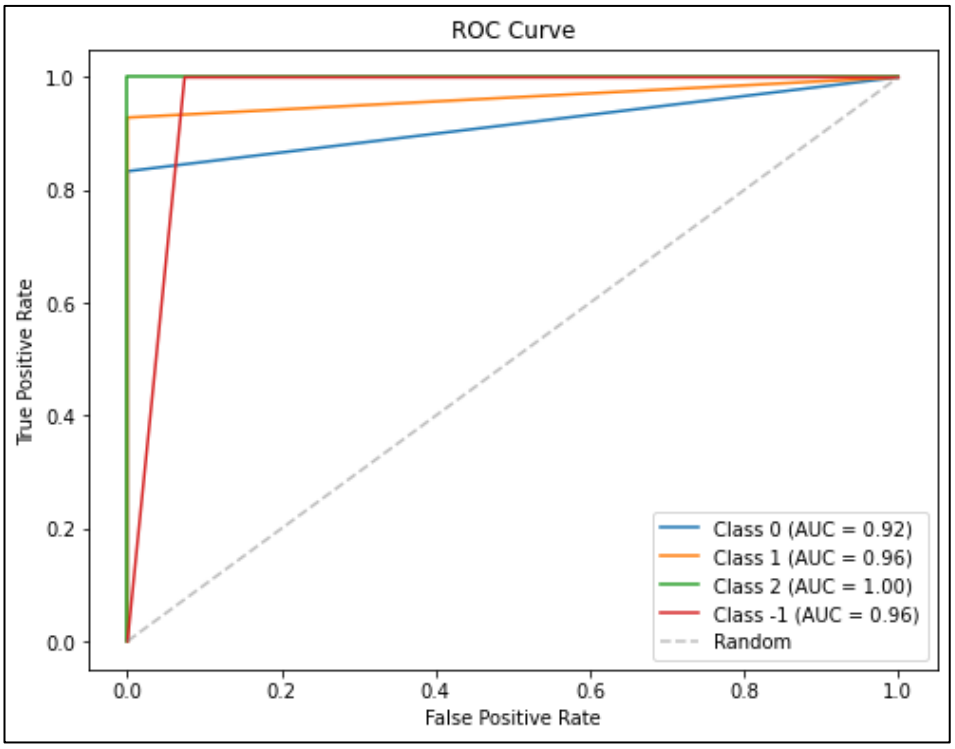

Figure 4 9. ROC AUC plan for K-Nearest Neighbors model.

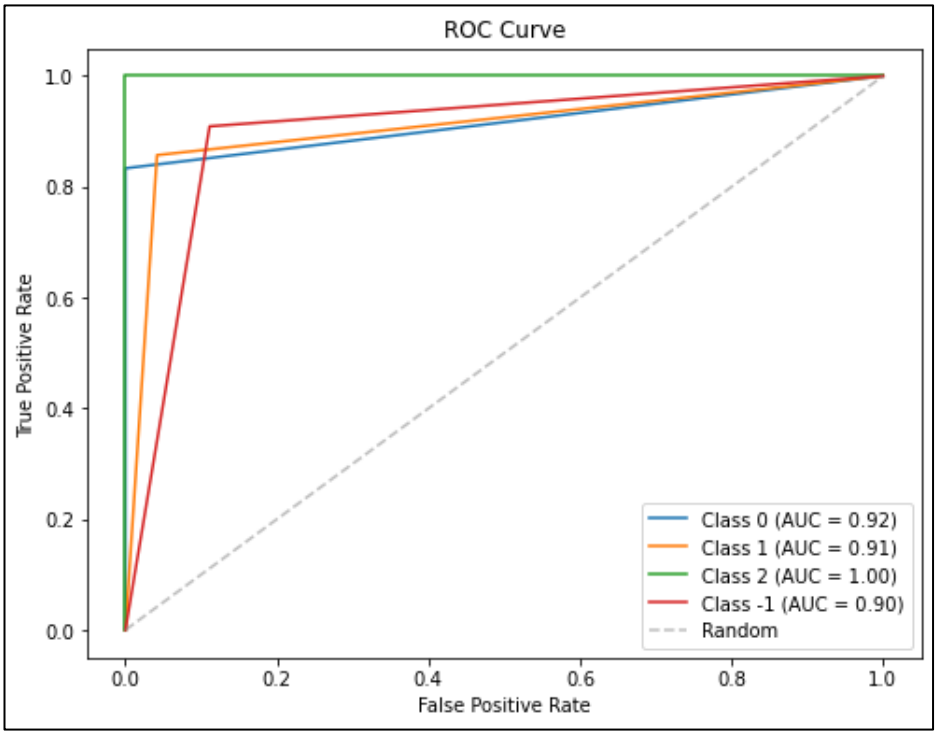

Figure 4 10. ROC AUC plan for SVM model.

PART 5

DISCUSSION

Operating systems are among the most important things that the user deals with, so it is important to ensure protection for them and their access to a security area. This is done by classifying weak points, which contributes to identifying and solving security problems before they cause more damage if hackers are able to access them. In this context, there are advanced algorithms in the field of learning with high success in partner classification; the performance of the Random Forest algorithm stands out as the most outstanding.

Table 5.1 shows that the Random Forest algorithm achieved an outstanding superiority of 97%, indicating the ability to recognize effective over weak ones. While other algorithms also perform well, the accuracy of Random Forest makes it the best choice in this context.

The system's overall performance on the security data classification set is represented in Figure 5.1. Random Forest generally outperformed everyone evaluated, including precision, positive precision, recall, f1-score, and ROC AUC, reaching 100%. This clearly shows the effectiveness of this technology in enhancing cybersecurity by allocating and detecting vulnerabilities in operating systems.

The results obtained from our study demonstrate the importance of using ML in improving Windows security and vulnerability classification. Let us review more details about the performance of each of the algorithms used:

Random Forest (97%): Accuracy: 97%, Precision: 97%, Recall: 98%, f1-score: 97%, ROC AUC point: 100%, Random Forest is a very effective tool in identifying vulnerabilities, achieving outstanding performance in all benchmarks, and significantly enhancing the ability to detect vulnerabilities in Windows.

Logistic Regression (82%): Accuracy: 82%, Precision: 82%, Recall: 83%, f1-score: 80%, ROC AUC point: 97%

Logistic Regression shows good performance, a good ability to classify vulnerabilities, but it is not as accurate and effective as Random Forest.
Naive Bayes (68%): Accuracy: 68% , Precision: 68%, Recall: 54% ,f1-score: 59%, ROC AUC point: 88%

Naive Bayes appears less effective in identifying vulnerabilities, especially regarding recall and f1-score.

K-Nearest Neighbors (89%): Accuracy: 89% , Precision: 89% , Recall: 90% , f1-score: 90% , ROC AUC point: 97%

K-Nearest Neighbors shows strong performance in vulnerability classification, identifying where there may be vulnerabilities in the operating system.
Support Vector Machines (95%): Accuracy: 95%, Precision: 95%, Recall: 96%, f1-score: 95%, ROC AUC point: 96%

The Support Vector Machines algorithm also shows strong performance in vulnerability classification, making it an important tool in enhancing operating system security.

Table 5 1. Class-based performance indicators are shown.

Method	Accuracy	Precision	Recall	F1-score	Roc Auc Score
Random Forest	**97%**	**97%**	**98%**	**97%**	**100%**
Logistic Regression	82%	82%	83%	80%	97%
Naive Bayes	68%	68%	54%	59%	88%
K-Nearest Neighbors	89%	89%	90%	90%	97%
Support Vector Machines	95%	95%	96%	95%	96%

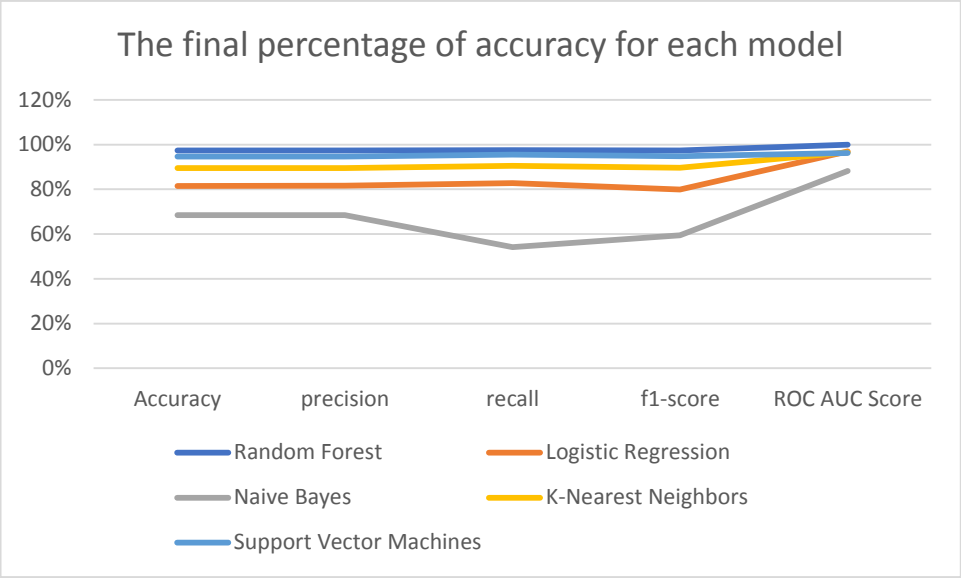

Figure 5 1. Class-based performance indicators are shown.

There are several reasons why the random forest algorithm performed better than the rest, especially if the data was classified based on a single feature:

- Diversity of models: The performance of a random forest depends on creating a large number of (multiple) decision trees and pooling their results. This gives great versatility to the models, increasing their ability to handle a various data.
- Avoid bias and noise: Random forest uses a random process to select the data samples and characteristics used in each decision tree, reducing the possibility of bias or noise in the model.
- Dealing with Imbalanced Data: In the case of imbalanced data (the balance of classes is different), Random Forest can handle this situation effectively by introducing voting through a large set of decision trees.

Now, let's discuss the workings of each of the five algorithms that were used:

Random forest is characterized by its ability to handle large data sets and multiple variables efficiently and is considered resistant to noise and bias. This algorithm can identify complex data models excellently, in part because of the diversity of the

models it creates. However, random forest may require relatively large computational resources for training, which can be a disadvantage in some cases.

As for logistic regression, the model is simple and easy to interpret, which makes it suitable for data whose relationship is linear. However, it can be ineffective with non-linear data or in the case of large discrepancies in the data.

As for the naive Bayes algorithm is characterized by the simplicity of the model and the speed of training, and it works well with a large set of data. However, it relies on the assumption of conditional independence between variables, which may not always be accurate.

K-nearest neighbors work well with non-linear data and random variables, and the model is simple and easy to interpret. However, it can be slow to predict with large amounts of data.

Support vector machines effectively separate classes in a large, high-dimensional space and work well on non-linear data. However, it may be sensitive to parameter settings and may be slow to train with large amounts of data.

Based on this, it can be seen that random forest has greater model diversity and data handling capacity, leading to better performance, especially when the data is dependent on a single property.

PART 6

CONCLUSION

6.1. CONCLUSION

In conclusion, the rapid progress in systems and technology presents many challenges data security and information protection. The cybersecurity discipline develops strategies and procedures to ensure data protection. In our study we used ML methodologies to enhance the security of the Windows operating system; and to classify Windows vulnerabilities. We relied on five ML techniques: Support Vector Machine (SVM), Random Forest, Logistic Regression, Native Bayes, and Exploit-deb. The dataset was obtained from the National Institute of Standards and Technology (NIST).

Performance metrics were evaluated, which included precision, recall, roc auc score, and f1 score. Based on the results, it is clear that Random Forest excels at identifying vulnerabilities, as we achieved a commendable 97% accuracy rate in classifying Windows vulnerabilities using five different ML techniques. Both SVM and Random Forest algorithms showed great performance potential. Therefore, using machine learning to enhance cybersecurity is a valuable and promising task.

This study demonstrated the efficiency of machine learning techniques in enhancing cybersecurity, as the use of machine learning in the field of data security can contribute to enhancing the ability of systems to discover and address security vulnerabilities effectively and efficiently, which reflects the importance of adopting advanced technologies to enhance information security in light of technological developments. This study considers the importance of using machine learning to protect data.

6.2. FUTURE WORK

To develop this work in the future, we aim to enhance the ability to detect threats in real time and with continuous data updating by developing an antivirus program based on machine learning techniques. In addition, it includes directing efforts towards:

- Improving accuracy and efficiency: Work continues to improve the accuracy of ML models to ensure effective threat detection with high accuracy.
- AI integration: Explore opportunities to integrate other AI technologies, such as neural networks and Deep ML, to enhance antivirus software performance.
- Expanding detection: Complete research to expand threat detection to include a wider range of potential attacks and scenarios.
- Continuous updates: Providing effective mechanisms to update and develop the program periodically with technological developments and the emergence of new threats.
- Collaboration and engagement: Foster collaboration with the cybersecurity community and researchers to share knowledge and develop collective strategies to address future challenges.

These plans allow strengthening readiness to face future challenges and continuously improving the effectiveness of cybersecurity programs.

REFERENCES

1. B. Chernis and R. Verma, “Machine learning methods for software vulnerability detection,” in IWSPA 2018 - Proceedings of the 4th ACM International Workshop on Security and Privacy Analytics, Co-located with CODASPY 2018, Association for Computing Machinery, Inc, Mar. 2018, pp. 31–39. doi: 10.1145/3180445.3180453.

2. J. P. Bharadiya, “Machine Learning in Cybersecurity: Techniques and Challenges.” [Online]. Available: www.ajpojournals.org

3. L. Kim, “Cybersecurity: Ensuring confidentiality, integrity, and availability of information,” in *Nursing Informatics: A Health Informatics, Interprofessional and Global Perspective*, Springer, 2022, pp. 391–410.

4. S. Nadler, “Heretics!,” Nadler Princeton University Press, 2017.

5. M. Vanamala, K. Bryant, and A. Caravella, “Attribution (CC-BY) 4.0 license,” *Journal of Computer Science*, 2023, doi: 10.3844/jcssp.2023.

6. J. Moon, S. Kim, P. Jangyong, J. Lee, K. Kim, and J. Song, “MalDC: Malicious Software Detection and Classification using Machine Learning,” *KSII Transactions on Internet and Information Systems*, vol. 16, no. 5, pp. 1466–1488, May 2022, doi: 10.3837/TIIS.2022.05.004.

7. G. Apruzzese *et al.*, “The Role of Machine Learning in Cybersecurity,” *Digital Threats: Research and Practice*, vol. 4, no. 1, Mar. 2023, doi: 10.1145/3545574.

8. K. Sharifani and M. Amini, “Machine Learning and Deep Learning: A Review of Methods and Applications,” *World Information Technology and Engineering Journal*, vol. 10, no. 07, pp. 3897–3904, 2023.

9. Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, “A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions,” *Electronics (Basel)*, vol. 12, no. 6, p. 1333, 2023.

10. D. Mandal and İ. KÖsesoy, “Prediction of Software Security Vulnerabilities from Source Code Using Machine Learning Methods,” in *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, 2023, pp. 1–6.

11. B. Chernis and R. Verma, “Machine learning methods for software vulnerability detection,” in *Proceedings of the fourth ACM international workshop on security and privacy analytics*, 2018, pp. 31–39.

12. U. Adamu and I. Awan, “Ransomware prediction using supervised learning algorithms,” in *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, IEEE, 2019, pp. 57–63.

13. Y. Xue, “Machine Learning: Research on Detection of Network Security Vulnerabilities by Extracting and Matching Features,” *Journal of Cyber Security and Mobility*, pp. 697–710, 2023.

14. K. Munonye and M. Péter, “Machine learning approach to vulnerability detection in OAuth 2.0 authentication and authorization flow,” *Int J Inf Secur*, vol. 21, no. 2, pp. 223–237, 2022.

15. Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, “A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions,” *Electronics (Switzerland)*, vol. 12, no. 6. MDPI, Mar. 01, 2023. doi: 10.3390/electronics12061333.

16. “List of security hacking incidents - Wikipedia.” Accessed: Jan. 14, 2024. [Online]. Available: https://en.wikipedia.org/wiki/List_of_security_hacking_incidents

17. “The History of Cybersecurity | Avast.” Accessed: Jan. 14, 2024. [Online]. Available: https://blog.avast.com/history-of-cybersecurity-avast

18. D. of Defense, “Trusted Computer System Evaluation Criteria (TCSEC),” 1985.

19. R. Lehtinen and G. T. Gangemi Sr, *Computer security basics: computer security*. “ O’Reilly Media, Inc.,” 2006.

20. G. Sharma, A. Kumar, and V. Sharma, “Windows operating system vulnerabilities,” *International Journal of Computing and Corporate Research*, vol. 1, no. 3, 2011.

21. “NVD - Search and Statistics.” Accessed: Jan. 14, 2024. [Online]. Available: https://nvd.nist.gov/vuln/search

22. Mahāwitthayālai Khǭn Kǣn, IEEE Thailand Section., and Institute of Electrical and Electronics Engineers, *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE) : 13-15 July 2016.*

23. “Vulnerability Trend Dashboard - SC Dashboard | Tenable®.” Accessed: Jan. 14, 2024. [Online]. Available: https://www.tenable.com/sc-dashboards/vulnerability-trend-dashboard

24. “Vulnerability Database Catalog.” Accessed: Jan. 14, 2024. [Online]. Available: https://www.first.org/global/sigs/vrdx/vdb-catalog

25. V. Yosifova, A. Tasheva, and R. Trifonov, “Predicting Vulnerability Type in Common Vulnerabilities and Exposures (CVE) Database with Machine

Learning Classifiers," in *12th National Conference with International Participation, ELECTRONICA 2021 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., May 2021. doi: 10.1109/ELECTRONICA52725.2021.9513723.

26. F. A. Alshaya, S. S. Alqahtani, and Y. A. Alsamel, "VrT: A CWE-Based Vulnerability Report Tagger : Machine Learning Driven Cybersecurity Tool for Vulnerability Classification," in *Proceedings - 2023 IEEE/ACM 1st International Workshop on Software Vulnerability, SVM 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 10–13. doi: 10.1109/SVM59160.2023.00007.

27. "CVE - CVE." Accessed: Jan. 14, 2024. [Online]. Available: https://cve.mitre.org/

28. "NVD - CVSS v2 Calculator." Accessed: Jan. 14, 2024. [Online]. Available: https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator

29. "NVD - CVSS v3 Calculator." Accessed: Jan. 14, 2024. [Online]. Available: https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator

30. "NVD - Vulnerability Metrics." Accessed: Jan. 14, 2024. [Online]. Available: https://nvd.nist.gov/vuln-metrics/cvss

31. S. Badillo *et al.*, "An introduction to machine learning," *Clin Pharmacol Ther*, vol. 107, no. 4, pp. 871–885, 2020.

32. A. Cocho-Bermejo and M. Vogiatzaki, "Phenotype Variability Mimicking as a Process for the Test and Optimization of Dynamic Facade Systems," *Biomimetics*, vol. 7, no. 3, Sep. 2022, doi: 10.3390/biomimetics7030085.

33. D. Dasgupta, Z. Akhtar, and S. Sen, "Machine learning in cybersecurity: a comprehensive survey," *Journal of Defense Modeling and Simulation*, vol. 19, no. 1, pp. 57–106, Jan. 2022, doi: 10.1177/1548512920951275.

34. K. A. Al-Enezi, I. F. Al-Shaikhli, A. R. Al-Kandari, and L. Z. Al-Tayyar, "A survey of intrusion detection system using case study Kuwait Governments entities," in *2014 3rd International Conference on Advanced Computer Science Applications and Technologies*, IEEE, 2014, pp. 37–43.

35. Y. Wang, Y. Wang, J. Liu, Z. Huang, and P. Xie, "A survey of game theoretic methods for cyber security," in *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*, IEEE, 2016, pp. 631–636.

36. J. Martínez Torres, C. Iglesias Comesaña, and P. J. García-Nieto, "Review: machine learning techniques applied to cybersecurity," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 10, pp. 2823–2836, Oct. 2019, doi: 10.1007/s13042-018-00906-1.

37. S. Silvestri, S. Islam, S. Papastergiou, C. Tzagkarakis, and M. Ciampi, “A machine learning approach for the NLP-based analysis of cyber threats and vulnerabilities of the healthcare ecosystem,” *Sensors*, vol. 23, no. 2, p. 651, 2023.

38. S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, “A comparison of machine learning techniques for phishing detection,” in *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, 2007, pp. 60–69.

39. E. N. Ceesay, *Mitigating phishing attacks: a detection, response and evaluation framework*. University of California, Davis, 2008.

40. S. Purkait, “Phishing counter measures and their effectiveness–literature review,” *Information Management & Computer Security*, vol. 20, no. 5, pp. 382–420, 2012.

41. K. Rezaee, S. M. Rezakhani, M. R. Khosravi, and M. K. Moghimi, “A survey on deep learning-based real-time crowd anomaly detection for secure distributed video surveillance,” *Pers Ubiquitous Comput*, pp. 1–17, 2021.

42. R. Canzanese, M. Kam, and S. Mancoridis, “Toward an automatic, online behavioral malware classification system,” in *2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems*, IEEE, 2013, pp. 111–120.

43. T. S. Guzella and W. M. Caminhas, “A review of machine learning approaches to spam filtering,” *Expert Syst Appl*, vol. 36, no. 7, pp. 10206–10222, 2009.

44. Clusters,” *Commun ACM*, vol. 51, no. 1, pp. 107–113, 2008.

45. A. Klimburg, *National cyber security framework manual*. NATO Cooperative Cyber Defense Center of Excellence, 2012.

46. A. Almomani, B. B. Gupta, S. Atawneh, A. Meulenberg, and E. Almomani, “A survey of phishing email filtering techniques,” *IEEE communications surveys & tutorials*, vol. 15, no. 4, pp. 2070–2090, 2013.

47. A. M. Y. Koay, R. K. L. Ko, H. Hettema, and K. Radke, “Machine learning in industrial control system (ICS) security: current landscape, opportunities and challenges,” *Journal of Intelligent Information Systems*, vol. 60, no. 2. Springer, pp. 377–405, Apr. 01, 2023. doi: 10.1007/s10844-022-00753-1.

48. J. Wang, Y. Miao, A. Khamis, F. Karray, and J. Liang, “Adaptation approaches in unsupervised learning: a survey of the state-of-the-art and future directions,” in *Image Analysis and Recognition: 13th International Conference, ICIAR 2016, in Memory of Mohamed Kamel, Póvoa de Varzim, Portugal, July 13-15, 2016, Proceedings 13*, Springer, 2016, pp. 3–11.

49. E. Anthi, L. Williams, M. Rhode, P. Burnap, and A. Wedgbury, “Adversarial attacks on machine learning cybersecurity defences in industrial control systems,” *Journal of Information Security and Applications*, vol. 58, p. 102717, 2021.

50. M. Pawlicki, M. Choraś, and R. Kozik, “Defending network intrusion detection systems against adversarial evasion attacks,” *Future Generation Computer Systems*, vol. 110, pp. 148–154, 2020.

51. A. Agarwal, R. Singh, M. Vatsa, and N. Ratha, “Image transformation-based defense against adversarial perturbation on deep learning models,” *IEEE Trans Dependable Secure Comput*, vol. 18, no. 5, pp. 2106–2121, 2020.

52. E. Wåreus, A. Duppils, M. Tullberg, and M. Hell, “Security Issue Classification for Vulnerability Management with Semi-supervised Learning.,” in *ICISSP*, 2022, pp. 84–95.

53. “Exploit Database - Exploits for Penetration Testers, Researchers, and Ethical Hackers.” Accessed: Jan. 16, 2024. [Online]. Available: https://www.exploit-db.com/

54. P. Cerda, G. Varoquaux, and B. Kégl, “Similarity encoding for learning with dirty categorical variables,” *Mach Learn*, vol. 107, no. 8–10, pp. 1477–1494, 2018.

55. M. J. Davis, “Contrast coding in multiple regression analysis: Strengths, weaknesses, and utility of popular coding structures,” *Journal of data science*, vol. 8, no. 1, pp. 61–73, 2010.

56. R. Bitton, N. Maman, I. Singh, S. Momiyama, Y. Elovici, and A. Shabtai, “Evaluating the Cybersecurity Risk of Real World, Machine Learning Production Systems,” Jul. 2021, [Online]. Available: http://arxiv.org/abs/2107.01806

57. V. Jackins, S. Vimal, M. Kaliappan, and M. Y. Lee, “AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes,” *Journal of Supercomputing*, vol. 77, no. 5, pp. 5198–5219, May 2021, doi: 10.1007/s11227-020-03481-x.

58. S. Joshi, Japanpreet, L. Rani, P. K. Sarangi, and V. P. Dubey, “Strengthening Cybersecurity: A Comparative Study of KNN and Random Forest for Spam Detection,” in *International Conference on Recent Developments in Cyber Security*, Springer, 2023, pp. 337–350.

59. A. Suresh, R. Udendhran, and S. Vimal, “Deep neural networks for multimodal imaging and biomedical applications,” *Deep Neural Networks for Multimodal Imaging and Biomedical Applications*, pp. 1–294, Jun. 2020, doi: 10.4018/978-1-7998-3591-2.

60. Q. H. Hoang, L. M. Duong, P. L. L. Bui, A. V. Tran, T. A. Nguyen, and V. D. Nguyen, “A Comparative Study of Machine Learning Algorithms for Breast Cancer Classification,” in *2023 International Conference on Advanced Technologies for Communications (ATC)*, IEEE, Oct. 2023, pp. 409–414. doi: 10.1109/ATC58710.2023.10318887.

61. X. L. D. Huang, R. F. Kunz, and X. I. A. Yang, “Linear logistic regression with weight thresholding for flow regime classification of a stratified wake,” *Theoretical and Applied Mechanics Letters*, vol. 13, no. 2, Mar. 2023, doi: 10.1016/j.taml.2022.100414.

62. X. I. A. Yang, P. E. S. Chen, R. Hu, and M. Abkar, “Logarithmic-Linear Law of the Streamwise Velocity Variance in Stably Stratified Boundary Layers,” *Boundary Layer Meteorol*, vol. 183, no. 2, pp. 199–213, May 2022, doi: 10.1007/S10546-021-00683-5/METRICS.

63. Y. Wei, M. Gao, J. Xiao, C. Liu, Y. Tian, and Y. He, “Research and implementation of cancer gene data classification based on deep learning,” *Journal of Software Engineering and Applications*, vol. 16, no. 6, pp. 155–169, 2023.

64. B. Bai *et al.*, “Naive Bayes classification-based surface water gap-filling from partially contaminated optical remote sensing image,” *J Hydrol (Amst)*, vol. 616, Jan. 2023, doi: 10.1016/j.jhydrol.2022.128791.

65. P. Dini, A. Elhanashi, A. Begni, S. Saponara, Q. Zheng, and K. Gasmi, “Overview on Intrusion Detection Systems Design Exploiting Machine Learning for Networking Cybersecurity,” *Applied Sciences*, vol. 13, no. 13, p. 7507, 2023.

66. A. X. Wang, S. S. Chukova, and B. P. Nguyen, “Ensemble k-nearest neighbors based on centroid displacement,” *Inf Sci (N Y)*, vol. 629, pp. 313–323, 2023.

67. W. Zhang, “Machine Learning Approaches to Predicting Company Bankruptcy,” *Journal of Financial Risk Management*, vol. 06, no. 04, pp. 364–374, 2017, doi: 10.4236/jfrm.2017.64026.

68. P. Darveau, “Support Vector Machines: Modeling The Dual Cognitive Processes of an SVM,” 2023.

69. A. Churcher *et al.*, “An experimental analysis of attack classification using machine learning in IoT networks,” *Sensors (Switzerland)*, vol. 21, no. 2, pp. 1–32, Jan. 2021, doi: 10.3390/s21020446.

70. P. Thölke *et al.*, “Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data,” *Neuroimage*, vol. 277, p. 120253, 2023.

71. R. Thukral, A. K. Aggarwal, A. S. Arora, T. Dora, and S. Sancheti, “Artificial intelligence-based prediction of oral mucositis in patients with head-and-neck

cancer: A prospective observational study utilizing a thermographic approach," *Cancer Research, Statistics, and Treatment*, vol. 6, no. 2, pp. 181–190, 2023.

72. T. R. Noviandy, G. M. Idroes, A. Maulana, I. Hardi, E. S. Ringga, and R. Idroes, "Credit Card Fraud Detection for Contemporary Financial Management Using XGBoost-Driven Machine Learning and Data Augmentation Techniques," *Indatu Journal of Management and Accounting*, vol. 1, no. 1, pp. 29–35, 2023.

73. Z. H. Hoo, J. Candlish, and D. Teare, "What is an ROC curve?," *Emergency Medicine Journal*, vol. 34, no. 6, pp. 357–359, Jun. 2017, doi: 10.1136/emermed-2017-206735.

RESUME

Nooralhuda AL-SARRAY began my academic journey at Halimah al-Sadiyah primary school. I completed my secondary education at AL-Hay Preparatory school in the 2013-2014 academic year. Subsequently, I pursued my undergraduate studies at Wasit University College of Education for Pure Sciences, graduating in 2017-2018. I graduated among the top students and worked as a teaching assistant at the university from 2019 to 2022. After that in 2022, I moved to Karabük, Turkey, to undertake postgraduate studies. I enrolled in a Master of Computer Engineering program at Karabük University.