CYBER HACKING BREACHES PREDICTION USING MACHINE LEARNING

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

NALLURI KARTHIK	20UECS0659	**VTU15337**
KOTA MANISH	20UECS0501	**VTU17731**

Under the guidance of
Dr. P.J Beslin pajila M.E ., Ph.D .,
ASSISTANT PROFESSOR - SENIOR GRADE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CYBER HACKING BREACHES PREDICTION USING MACHINE LEARNING

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

NALLURI KARTHIK	20UECS0659	**VTU15337**
KOTA MANISH	20UECS0501	**VTU17731**

Under the guidance of
Dr. P.J Beslin pajila M.E ., Ph.D .,
ASSISTANT PROFESSOR - SENIOR GRADE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report title “CYBER HACKING BREACHES PREDICTION USING MACHINE LEARNING” by “NALLURI KARTHIK 20UECS0659, KOTA MANISH 20UECS0501, ” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

Signature of Professor In-charge
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

NALLURI KARTHIK

Date: / /

KOTA MANISH

Date: / /

APPROVAL SHEET

This project report entitled “CYBER HACKING BREACHES PREDICTION USING MACHINE LEARNING” by NALLURI KARTHIK 20UECS0659, KOTA MANISH 20UECS0501 is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Dr. P.J.Beslin Pajila ,M.E., Ph.D.,
ASSISTANT PROFESSOR.

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering,Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our Internal Supervisor **Dr. P.J BESLIN PAJILA M.E ., Ph.D .,** for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

NALLURI KARTHIK **20UECS0659**
KOTA MANISH **20UECS0501**

ABSTRACT

The rapid evolution of cyber threats necessitates proactive measures to anticipate and mitigate potential breaches. In this study, we employ machine learning algorithms including Decision Tree, Random Forest, AdaBoost, Logistic Regression, KNN, and SVC to forecast cyber hacking breaches. By analyzing historical data and identifying patterns indicative of potential security breaches, these algorithms enable proactive decision-making and resource allocation for cybersecurity defenses. Through rigorous evaluation and comparison of these algorithms, we aim to determine the most effective approach for predicting cyber hacking breaches. This research contributes to enhancing cybersecurity strategies by providing insights into the predictive capabilities of various machine learning techniques. By leveraging predictive analytics, organizations can strengthen their defenses, detect vulnerabilities, and preemptively thwart cyber threats, ultimately safeguarding sensitive information and preserving operational integrity in an increasingly digital landscape.

Cyber-physical systems (cps) have made significant progress in many dynamic applications due to the integration between physical processes, computational resources, and communication capabilities. However, cyber-attacks are a major threat to these systems. Unlike faults that occurs by accidents cyber-physical systems, cyber-attacks occur intelligently and stealthy. Some of these attacks which are called deception attacks, inject false data from sensors or controllers, and also by compromising with some cyber components, corrupt data, or enter misinformation into the system. If the system is unaware of the existence of these attacks, it won't be able to detect them, and performance may be disrupted or disabled altogether. Therefore, it is necessary to adapt algorithms to identify these types of attacks in these systems. It should be noted that the data generated in these systems is produced in very large number, with so much variety, and high speed, so it is important to use machine learning algorithms to facilitate the analysis and evaluation of data and to identify hidden patterns.

Keywords: Decision Tree, Random Forest, AdaBoost, Logistic Regression, KNN, SVC.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

CAM	Comprehensive Assessment Mode
CSRM	Cyber Security Risk Management
CPS	Cyber Physical Systems
DOS	Denial Of Service
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Ac Countability Act
ISMS	International Standard for Information Security Management Systems
IDS	Intrusion Detection System
KNN	K-Nearest Neighbor
NCS	Networked Control System
ROC	RATE Of Investment
SOC	Security Operations Centers
SIEM	Security Information And Event Management
SVM	Support Vector Machine
TTP	Tactic, Technique and Procedure
VCBD	VERIS Community Dataset

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

The advent of the digital age has ushered in unprecedented opportunities for con nectivity and innovation. However, it has also given rise to a growing menace– cyber hacking breaches. In recent years, cyberattacks have become increasingly sophisti cated and devastating, posing a significant threat to individuals, businesses, and even nations.

This project aims to comprehensively investigate and analyze cyber hacking breach that occurred in the past year. We will delve into the methods, motivations, and im pacts of these breaches to gain a deeper understanding of the evolving landscape of cyber threats. By examining a range of high-profile cases, we intend to identify common vulnerabilities and attack vectors.

Cyber hacking breaches are driven by various motivations, including financial gain, cyber espionage, hacktivism, and personal curiosity. Criminals seek profit through data theft and fraud, while state actors engage in espionage. Hacktivists pursue political or social causes, and some hackers explore vulnerabilities for per sonal notoriety, highlighting diverse threats in cybersecurity. Additionally, we will explore the ethical, legal, and regulatory aspects surrounding cyberattacks and data breaches.

This research not only serves as a valuable resource for cybersecurity profession als but also contributes to raising awareness among individuals and organizations about the importance of robust digital security. By shedding light on the ever evolv- ing world of cyber hacking breaches, we aim to empower stakeholders to for tify their defenses and safeguard their digital assets in an increasingly interconnected world.

1.2 Aim of the Project

To illustrate the escalating pace of cyber threats necessitates preemptive measures to anticipate and counter potential breaches. This study aims to develop predic tive models using machine learning algorithms such as Decision Tree, Random For est, AdaBoost and KNN to forecast cyber hacking breaches. By leveraging histor ical data and identifying patterns indicative of security risks, the research seeks to enhance cybersecurity strategies and enable proactive defense mechanisms against evolving cyber threats.

1.3 Project Domain

Using actual cybercrime data, the initial phase is to predict a cybercrime strategy, and the accuracy results are then compared. The second is to examine if the infor mation at hand can be used to predict cybercrime perpetrators. It is employed to hide a system's data. Information theft is caused by sensitive and highly confidential data as well as poor management. The hackers' techniques might be found in two differ ent methods. One is to move through with legal action, get in touch with the victim, and let them know about the violations. The organizations should be aware of the sorts, trends, and patterns of assaults for the purpose of enabling them to monitor the system.

A study on the consequences of these kinds of attacks in an effort for managing the prevention of occurring the beaches.And provide comprehensive study of the breaches that have occurred by the various organizations and financial effect Because of improvements in information technology, prices for memory and stor age devices, and the expansion of the digital economy, businesses and governmental organizations now acquire more data every day.

Businesses and organizations have the threat of data attacks because of the collect ing of personal data on their computers. Computer networks are used in manufac turing, healthcare, research. This information is transferring every second through network. These attacks are used for profit and destroy the important information and use that for own need which rises the risk of data. Hybrid based detection is used to detect the high false positive rates and low false positive rates. Anomaly based detection analyse the behaviour of traffic, where signature based detection has the previous attacks records and able to detect the possibilities.

1.4 Scope of the Project

The project scope encompasses assessing existing cybersecurity measures, identi fying vulnerabilities, implementing security enhancements, conducting user aware ness training, and establishing continuous monitoring protocols. It aims to fortify digital systems and data protection against cyber hacking breaches. Develop a ro bust machine learning model to analyze historical cyber hacking incidents, aiming to predict and identify potential future breaches based on patterns and trends.

Enhance cybersecurity preparedness by leveraging predictive analytics, enabling organizations to proactively address vulnerabilities and mitigate risks associated with cyber hacking threats. Evaluate the effectiveness of the developed machine learning system in real-time breach prediction, providing a valuable tool for preemptive cybersecurity measures and reducing the impact of potential cyber breaches.

Chapter 2

LITERATURE REVIEW

[1] Kwon et al.(2013), explored the security issue in the state estimation problem is investi gated for a networked control system (NCS). The communication channels between the sensors and the remote estimator in the NCS are vulnerable to attacks from ma licious adversaries. The false data injection attacks are considered. The aim of this paper to find the so-called insecurity conditions under which the estimation system is insecure in the sense that there exist malicious attacks that can bypass the anomaly detector but still lead to unbounded estimation errors. In particular, a new necessary and sufficient condition for the insecurity is derived in the case that all communica tion channels are compromised by the adversary .

[2] Pajic et al.(2017), investigated the significant increase in the num ber of security-related incidents in control systems. These include high-profile at tacks in a wide range of application domains, from attacks on critical infrastructure, as in the case of the Maroochy Water breach , and industrial systems (such as the StuxNet virus attack on an industrial supervisory control and data acquisition sys tem , and the German Steel Mill cyberattack) , to attacks on modern vehicles . Even high-assurance military systems were shown to be vulnerable to attacks, as illustrated in the highly publicized downing of the RQ-170 Sentinel U.S. drone . These inci dents have greatly raised awareness of the need for security in cyberphysical systems (CPSs), which feature tight coupling of computation and communication substrates with sensing and actuation components.

[3] Sheng et al.(2012), explored the Embedded computational resources in autonomous robotic vehicles are becoming more abundant and have enabled improved opera tional effectiveness of cooperative robotic systems in civilian and military applica tions. Compared to autonomous robotic vehicles that operate single tasks, coopera tive teamwork has greater efficiency and operational capability. Multirobotic vehicle systems have many potential applications, such as platooning of vehicles in urban transportation, the operation of the multiple robots, autonomous underwater vehi

cles, and formation of aircrafts in military affairs . The study of group behaviors for multirobot systems is the main objective of the work. Group cooperative behavior signifies that individuals in the group share a common objective and action according to the interest of the whole group. Group cooperation can be efficient if individuals in the group coordinate their actions well.

[4] Zeng et al.(2014),discussed the problem of reaching a consensus among all the agents in the networked control systems (NCS) in the presence of misbehaving agents. A reputation-based resilient distributed control algorithm is first proposed for the leader-follower consensus network. The proposed algorithm embeds a resilience mechanism that includes four phases (detection, mitigation, identification, and up date), into the control process in a distributed manner. At each phase, every agent only uses local and one-hop neighbors' information to identify and isolate the mis behaving agents, and even compensate their effect on the system.

[5] Hongtao et al.(2017), focused on resilient control of networked control systems (NCSs) under the denial of service (DoS) attacks which is characterized by a Markov process. Firstly, the packets dropout are modeled as Markov process accord ing to the gamebetweenattack strategies and defense strategies. Then, an NCS under such game results is modeled as a Markovian jump linear system and four theorems are proved for the system stability analysis and controller design. Finally, a numer ical example is used to illustrative the application of these theorems. Networked control systems (NCSs) have received an increasing attention in the past decades. Now, NCSs have been widely applied in industrial processes, electric power net works, intelligent transportation and so on. With the growing of the NCSs, network, as a critical element in an NCS, is vulnerable to cyber threats which can menace the control systems.

[6] M. Eling et al.(2016), defined the literature by Eling and Schnell (2016) explores the intricate landscape of cyber risk and cyber risk insurance, providing valuable insights into this rapidly evolving field. The authors delve into the realization of cy ber risk, offering a comprehensive analysis that spans issues such as identification, assessment, and mitigation strategies. The study, featured in the Journal of Risk Fi nance, is particularly noteworthy for its emphasis on the dynamic nature of cyber threats and the corresponding challenges in developing effective insurance mecha

nisms. Eling and Schnell contribute significantly to the scholarly discourse by ad dressing key facets of cyber risk management, ultimately shedding light on the com plexities associated with safeguarding organizations against cyber threats.

[7] Mandal et al.(2020), focused considering the different aspects of social events, responses and their relations to further improve the classification of the so cial sentiment. The proposed method covers not only the response due to major so cial events but also predicting and generating alert for situations of significant social importance. The approach has made use of Twitter datasets and performed aspect based sentiment analysis on the obtained text data. It is shown to outperform the state-of-the-art methods.

[8] Poyraz et al. (2020) investigated various factors that can affect the monetary im pact of data breaches on companies. This paper introduces a model for the total cost of a mega data breach based on a data set created from multiple sources that categorises stolen data for U.S. residents as personally identifiable information (PII) and sensitive personally identifiable information (SPII). They use a rigorous step wise regression analysis that includes polynomial and factorial multilevel effects of the independent variables. There are three significant findings. First, our model finds a significant relation between total data breach cost and revenue, the total amount of PII and SPII, and class action lawsuits.

[9] Kure et al.(2021), presented an effective cybersecurity risk management (CSRM) practice using assets criticality, predication of risk types and evaluating the effec tiveness of existing controls. They follow a number of techniques for the proposed unified approach including fuzzy set theory for the asset criticality, machine learning classifiers for the risk predication and comprehensive assessment model (CAM) for evaluating the effectiveness of the existing controls. The proposed approach consid ers relevant CSRM concepts such as asset, threat actor, attack pattern, tactic, tech nique and procedure (TTP), and controls and maps these concepts with the VERIS community dataset (VCDB) features for the risk predication. The experimental re sults reveal that using the fuzzy set theory in assessing assets criticality supports stakeholder for an effective risk management practice ..

[10] R. R. Subramanian et al.(2021), designed a model by using machine learning to defend a website from security breaches. The primary aim of this research work is to create a machine learning model, which trains in Realtime and monitors the website or a system and trains from the state-of-art attacks. The proposed model has created a web application using Django, which takes the data from multiple sources such as Amazon, Flipkart, Snapdeal, and Shop clues, which shows the data that is safe to obtain from the website. Then, the data will be sorted on our page and then it will be made secured and illegal for the external people to access the data from our website and the proposed model will monitor the website 24/7. The model is trained daily and it generates predictions from the several of datasets available and from the pre vious state-of-the-art attacks. This model will be trained from the existing datasets and the history of attacks and breaches on our website.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

In the existing system, implementation of machine learning algorithms is bit complex to build due to the lack of information about the data visualization. Mathematical calculations are used in existing system for SVM and Logistic Regression model building this may take the lot of time and complexity. To overcome all this, And use machine learning packages available in the scikit-learn library.

The existing system collects data from various sources such as network logs, sys tem event logs, firewall logs, intrusion detection system (IDS) alerts, and other cyber security telemetry data sources. Data preprocessing techniques are applied to clean, normalize, and transform the raw data into a format suitable for machine learning analysis. This may involve handling missing values, removing noise, and encoding categorical variables.

3.1.1 Disadvantages

1. Limited Accuracy: Machine learning predictions for cyber hacking breaches may suffer from limited accuracy due to the evolving nature of hacking techniques and strategies.

2. Over-reliance on Historical Data: Dependence on historical data for machine learning models can lead to a disadvantage as it may not fully capture emerging and novel hacking patterns.

3. Adversarial Attacks: Machine learning models are vulnerable to adversarial attacks, where malicious actors intentionally manipulate input data to deceive prediction systems and compromise their effectiveness.

4. Resource Intensiveness: Implementing and maintaining machine learning solutions for cyber breach prediction can be resource-intensive, requiring significant computational power and expertise.

3.2 Proposed System

This study proposes a predictive system for cyber hacking breaches using machine learning algorithms such as Decision Tree, Random For est, AdaBoost and KNN. Leveraging historical data analysis and pattern recognition, the system aims to forecast potential security breaches, enabling proactive decision making and resource allocation for cybersecurity defenses. Through rigorous eval uation and comparison of these algorithms, the system seeks to identify the most effective approach for predicting cyber threats, contributing to enhanced cybersecu rity strategies and preemptive threat mitigation in digital environments.

Random Forest is a powerful machine learning algorithm that assembles multiple decision trees to make accurate predictions. Each tree is trained on a subset of data and votes on the final prediction, resulting in improved accuracy and robustness. It mitigates overfitting and handles complex relationships in data by averaging predictions from different trees. Random Forest is versatile, handling classification and regression tasks effectively.

3.2.1 Advantages

1. Early Detection: Machine learning enables early identification of potential cyber hacking breaches, allowing proactive measures to be implemented before significant damage occurs.

2. Adaptive Analysis: ML algorithms adapt to evolving cyber threats, providing continuous and dynamic analysis for more accurate prediction of hacking attempts.

3. Pattern Recognition: Machine learning excels in recognizing patterns within vast datasets, enhancing the ability to detect subtle indicators of potential cyber breaches.

4. Improved Incident Response: ML-driven predictions empower organizations with timely insights, facilitating faster and more effective incident response strategies.

5. Enhanced Risk Mitigation: By leveraging machine learning, organizations can enhance their risk mitigation efforts, preemptively addressing vulnerabilities and minimizing the impact of cyber hacking breaches.

3.3 Feasibility Study

Feasibility study is an important step in the development of any project, These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements. Examples of functional requirements:
1) Authentication of user whenever he/she logs into the system
2) System shutdown in case of a cyber-attack
3) A verification email is sent to user whenever he/she register for the first time on some software system.It involves evaluating the viability of the project in terms of its technical, economic, and social feasibility.

3.3.1 Economic Feasibility

An economic feasibility study involves a thorough assessment of its financial via bility and potential returns on investment (ROI). This evaluation begins by estimat ing both the initial investment required and the recurring expenses associated with data acquisition, infrastructure setup, and personnel. Subsequently, potential revenue streams are identified, including cost savings resulting from preventing cyber hack ing breaches and potential revenue generation from offering cybersecurity services or licensing the predictive model. By comparing the expected benefits to the costs and conducting ROI analysis using financial metrics like NPV, IRR, and payback period, the project's economic performance is evaluated over time. Additionally, a comprehensive risk analysis is conducted to identify and mitigate potential risks that could impact the project's economic feasibility. Based on these findings, conclusions and recommendations are made regarding whether to proceed with the project, mod ify its scope, or explore alternative approaches. Through this economic feasibility study, stakeholders can make well-informed decisions about investing in the project to predict cyber hacking breaches using machine learning.

3.3.2 Technical Feasibility

Achieving technical feasibility the project's practicality from a technologi cal standpoint. It begins by evaluating the availability and quality of relevant datasets containing historical cybersecurity incidents, network logs, and system event logs. Additionally, the technical requirements for infrastructure, hardware, and software needed for data storage, preprocessing, model training, and deployment are analyzed. Consideration is given to scalability, compatibility with existing systems, and the computational resources required. The feasibility of implementing machine learn ing algorithms for breach prediction, such as logistic regression, random forests, or neural networks, is also examined. Furthermore, the project's adherence to legal and ethical guidelines regarding data privacy, security, and bias mitigation is assessed. By conducting a comprehensive technical feasibility study, stakeholders can deter mine whether the necessary resources, technology, and expertise are available to successfully implement the project and achieve its objectives.

3.3.3 Social Feasibility

A social feasibility study involves evaluating its acceptance, impact, and implica tions within the broader social context. It begins by assessing the societal need for improved cybersecurity measures to protect individuals, organizations, and critical infrastructure from cyber threats. Stakeholder perspectives, including those of end users, cybersecurity experts, policymakers, and the general public, are considered to understand their attitudes, concerns, and expectations regarding the project. Addi tionally, potential social benefits, such as enhanced trust and confidence in online se curity, are identified, along with any potential risks or negative consequences, such as privacy concerns or unintended biases in the machine learning models. The project's alignment with societal values, ethical principles, and legal regulations is evaluated to ensure that it promotes fairness, transparency, and accountability in cybersecu rity practices. By engaging stakeholders in dialogue, addressing their needs and concerns, and fostering collaboration and trust, the project can contribute to a more resilient and secure digital environment that benefits society as a whole. Through this social feasibility study, stakeholders can assess the project's potential social impact and its ability to garner support and acceptance from the community.

3.4 System Specification

3.4.1 Hardware Specification

• Processor: I5 and above generation
• RAM: 4GB and above

3.4.2 Software Specification

• Operating System:Windows 7/8/10
• Server side Script:HTML, CSS, Bootstrap JS
• Programming Language:Python
• Libraries:Flask, Pandas, Mysql.connector, Os, Smtplib, Numpy
• IDE/Workbench:PyCharm
• Technology:Python 3.6+
• Server Deployment:Xampp Server
• Database:MySQL

3.4.3 Standards and Policies

Standards and policies to ensure ethical, legal, and responsible implementation. One critical aspect is compliance with data privacy regulations such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Ac countability Act (HIPAA), which govern the collection, processing, and protection of personal and sensitive data. Additionally, adherence to industry-specific stan dards like ISO/IEC 27001 for information security management systems is essential to ensure the confidentiality, integrity, and availability of data throughout the project lifecycle. Ethical guidelines for AI and machine learning, such as those outlined by organizations like the IEEE and the Partnership on AI, provide principles for fair ness, transparency, accountability, and privacy in algorithmic decision-making.
• Standards Organization: International standard for information security management systems (ISMS).
• Division Name: Security Management
• Section Name: Cybersecurity Framework
• Designator of Legally binding Document: ISO/IEC 27001 .
• Jupyter Notebook-Standards Used: ISO3166-1:2018
• Python-Standard Used: ISO6160:1979

Anaconda Prompt: Anaconda prompt is a type of command line interface which explicitly deals with the ML modules and the navigator is available in all Windows, Linux, and MacOS. The Anaconda prompt has many integrated development environments that make coding easier. The user interface can also be implemented in Python.
Standard Used: ISO/IEC 27001

Jupyter: It's like an open-source web application that allows us to share and create documents that contain live code, equations, visualizations, and narrative text. It can be used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, and machine learning.
Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 General Architecture

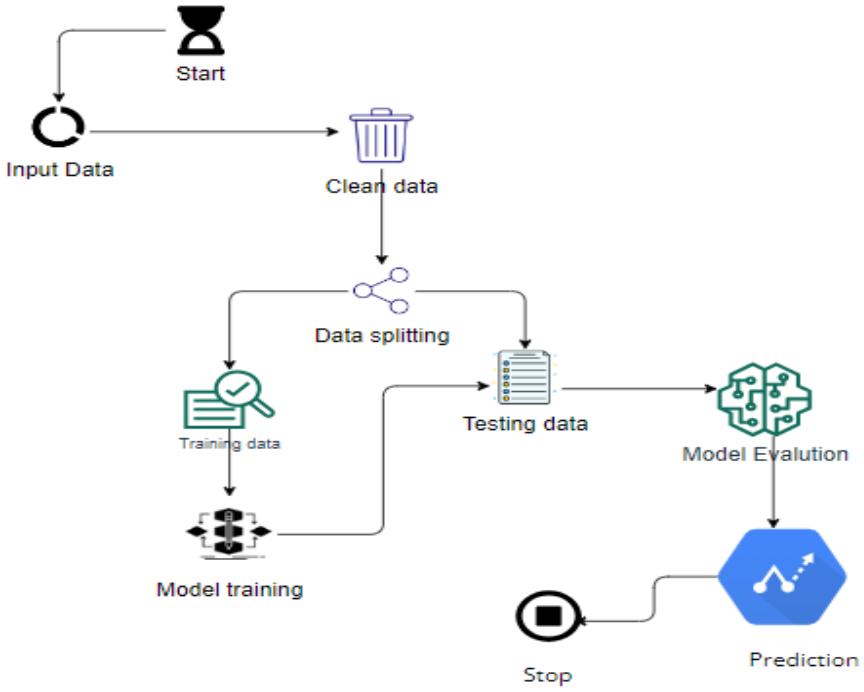

Figure 4.1: **Architecture Diagram**

Figure 4.1 outlines the system's overall workflow, starting with data acquisition illustrates the end-to-end process of predict ing cyber hacking breaches using machine learning, from data collection and prepro cessing to model deployment and continuous improvement. It emphasizes the inte gration of machine learning capabilities into existing cybersecurity infrastructure to enhance threat detection and response capabilities.

4.2 Design Phase

4.2.1 Data Flow Diagram

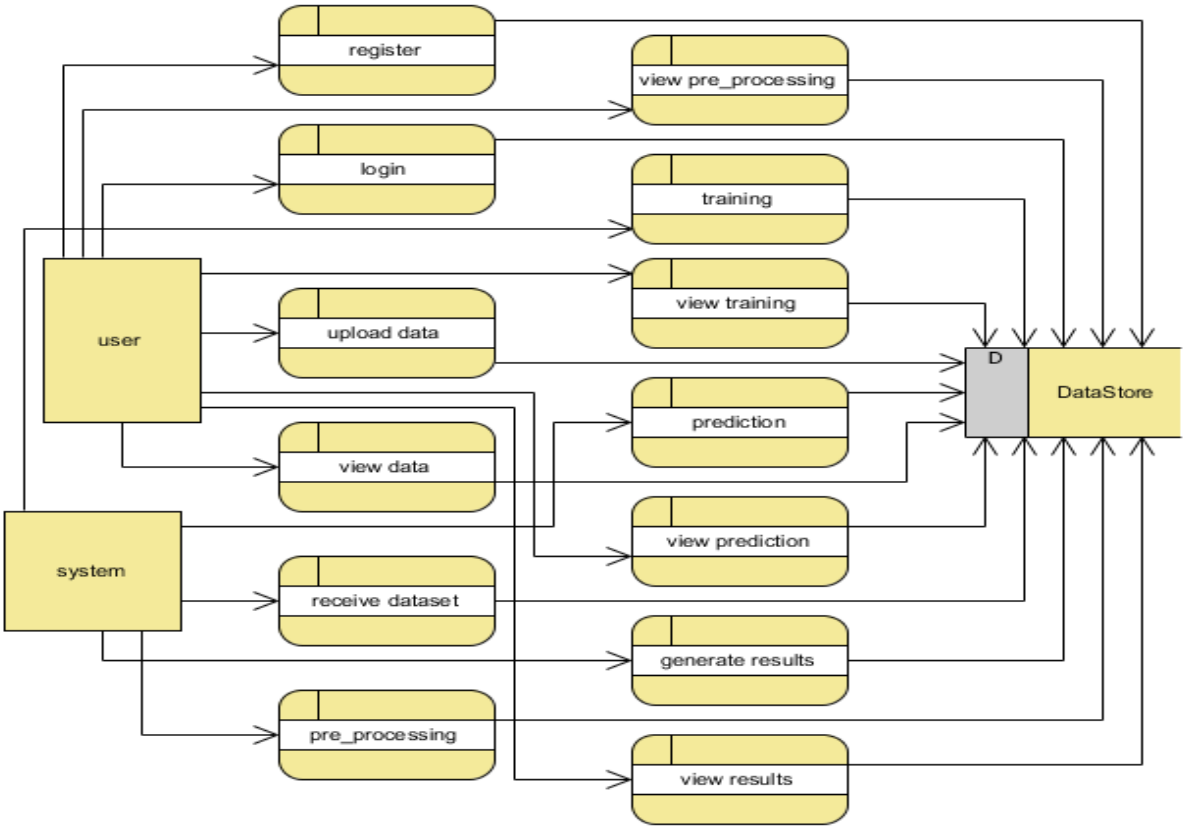

Figure 4.2: **Data Flow Diagram**

Figure 4.2 shows the flow of data, delineating several sequential steps. Initially, the system receives input in the form of a CSV file comprising data collection and preprocessing to model deployment and continuous improvement. It highlights the key stages and interac tions involved in predicting cyber hacking breaches using machine learning. Raw data from the sources undergo preprocessing to clean, transform, and prepare it for analysis. This step involves tasks such as handling missing values, removing noise, normalizing features, and encoding categorical variables.Feature engineering extracts relevant features from the preprocessed data to represent patterns indicative of potential hacking breaches. These features could include network traffic features, system activity features, user behavior features. Deployed models are integrated into the existing cybersecurity infrastructure for real-time or batch processing of data streams. Integration with security information and event management (SIEM) sys tems, security operations centers (SOCs), or threat intelligence platforms facilitates automated alerting and incident response based on model predictions.

4.2.2 Class Diagram

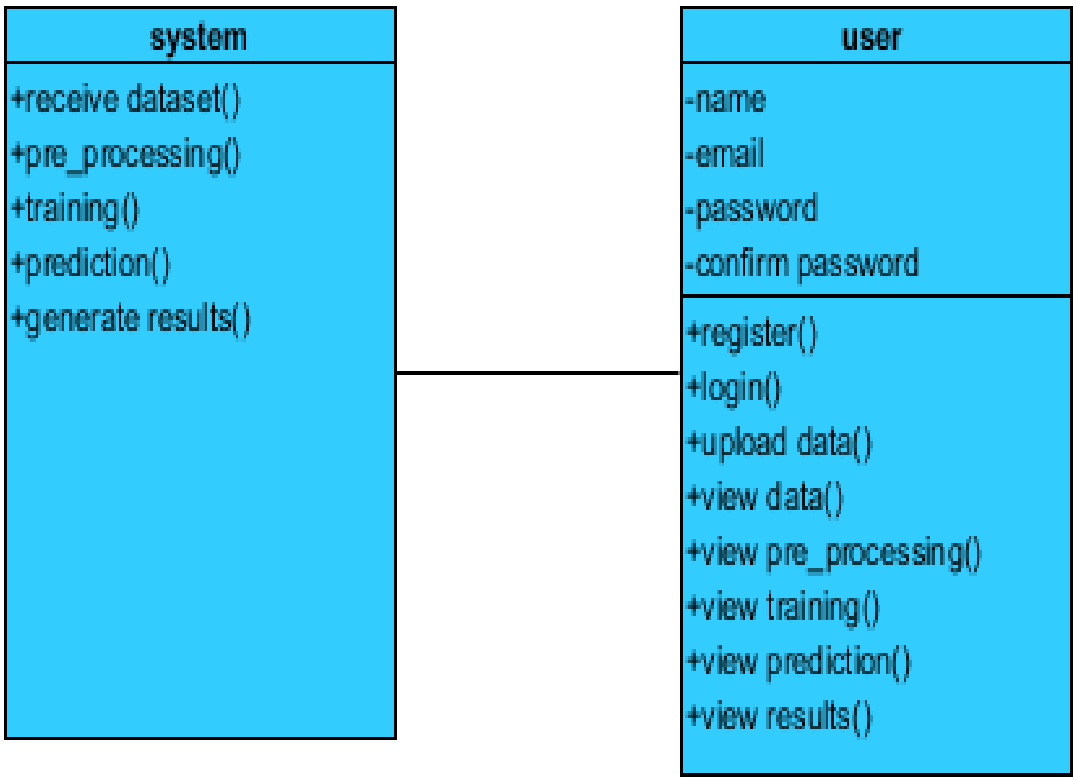

Figure 4.3: **Class Diagram**

Figure 4.3 illustrates the Class diagram a simplified view of the project's ar chitecture, focusing on the key components responsible for data processing, ma chine learning, evaluation, visualization, feedback loop management, integration, and compliance. Actual implementations may involve more classes and additional complexities based on specific requirements and design considerations. Static struc ture diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information DataProcessingSystem class represents the system responsible for preprocess ing data and extracting features from raw data sources. MachineLearningModel class encapsulates the machine learning model, including training, prediction, and model storage. ModelEvaluation class handles the evaluation of the machine learn ing model's performance using various metrics. DataVisualization class provides methods for visualizing the results of model predictions and evaluation. Feedback Loop class manages the feedback loop for monitoring model performance and up dating the model with new data.

4.2.3 Sequence Diagram

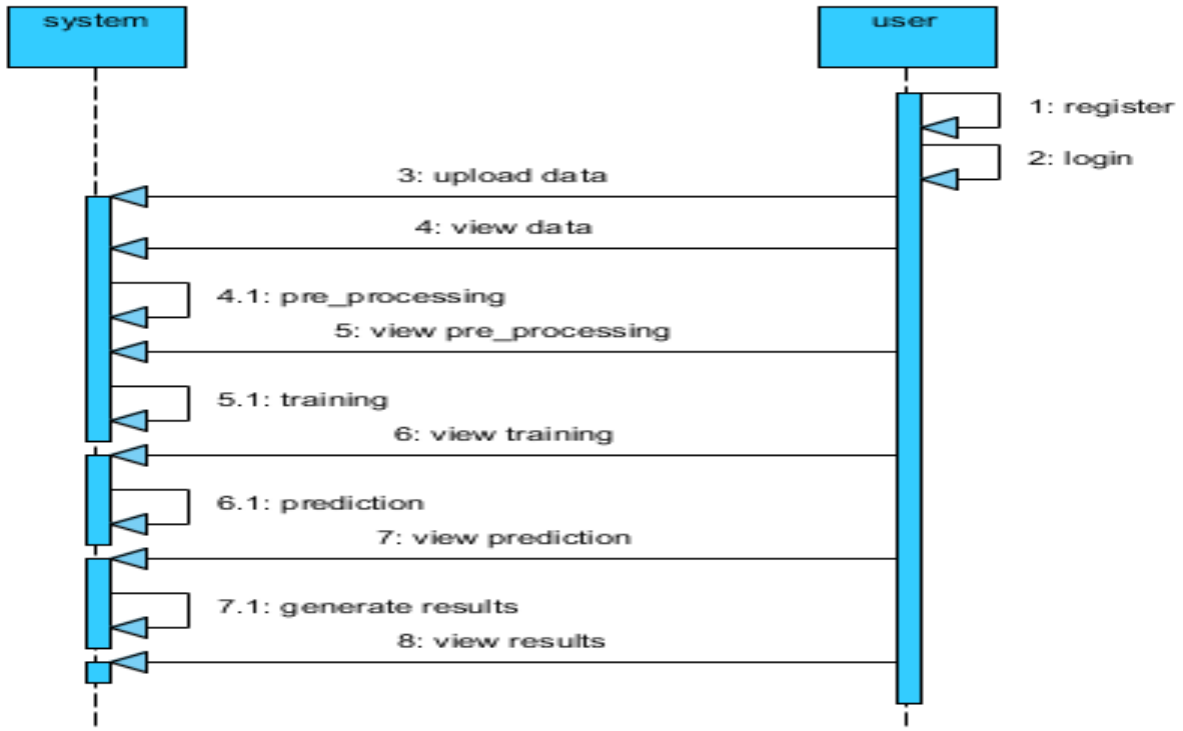

Figure 4.4: **Sequence Diagram**

Figure 4.4 illustrates the flow of interactions between different components of the project, including data retrieval, preprocessing, feature engineering, model prediction, evaluation, visualization, and feedback loop man agement. It highlights the key steps involved in predicting cyber hacking breaches using machine learning and the interactions between these steps. Actual implementa tions may involve more complex interactions and additional steps based on specific requirements and design considerations. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams

4.2.4 Activity Diagram

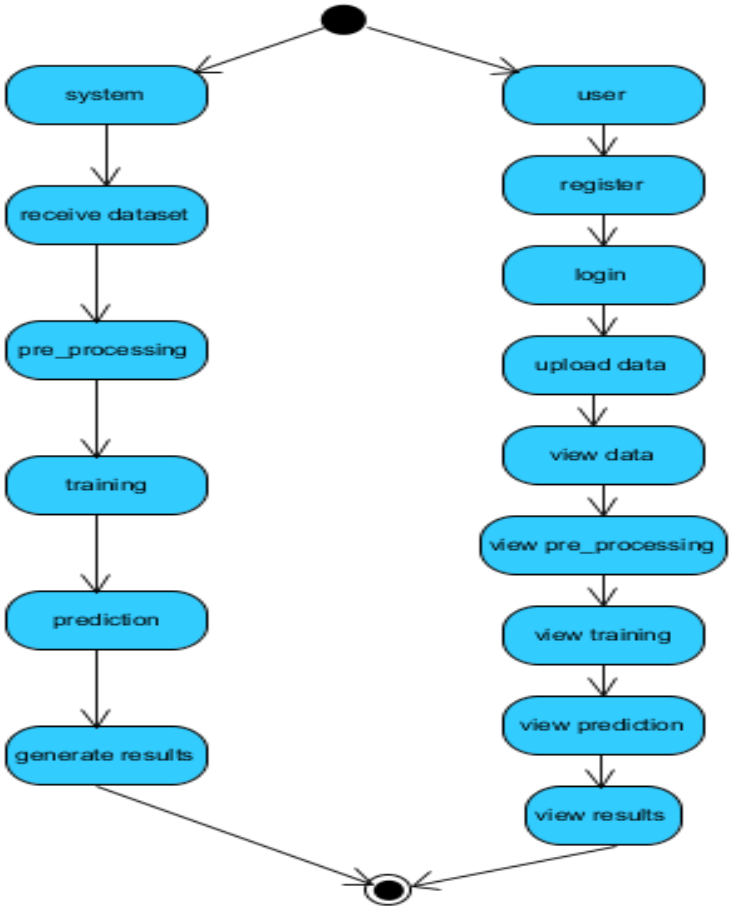

Figure 4.5: **Activity Diagram**

Figure 4.5 shows the overall flow of control initial phase involves gathering information about the target system or network. Machine learning may be utilized to analyze vast amounts of data quickly and identify potential vulnerabilities or weaknesses.hackers extract sensitive data from the compromised system or network. Machine learning techniques can help obfuscate data transfers, making it harder for security systems to detect and block the unauthorized transfer of data. to cover their tracks to evade detection and attribution. Machine learning can be employed to generate fake or misleading activity, modify logs, or employ other deception techniques

4.2.5 Collabration Diagram

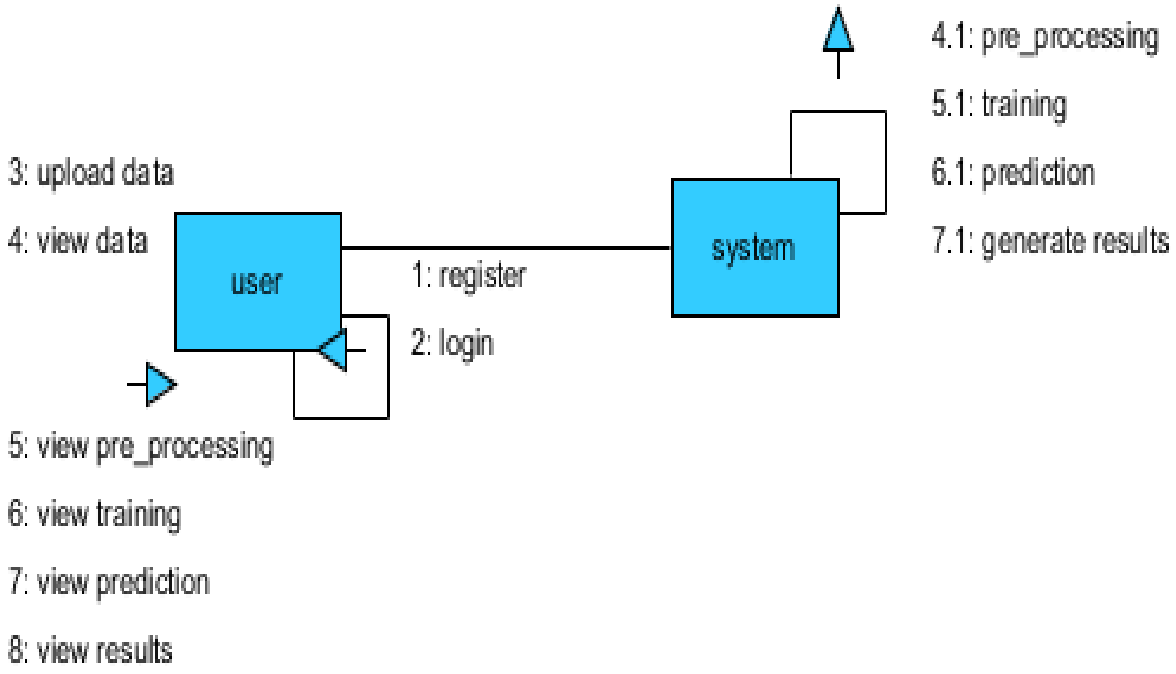

Figure 4.6: **Collabration Diagram**

Figure 4.6 shows that the method call sequence is indicated by some numbering technique as shown above. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

4.3 Algorithm & Pseudo Code

4.3.1 Random Forest Algorithm

1. Load the necessary libraries such as numpy, pandas, Tensorflow, Keras, sklearn, and matplotlib.
2. Collect historical data on cyber hacking incidents, including network logs, system event logs, firewall logs, intrusion detection system (IDS) alerts, etc.
3. Calculate technical indicators such as moving averages and exponential moving averages using the pandas rolling and ewm functions.
4. Determine the scope of the project, including the types of breaches to predict and the data available.
5. Ensure the data is representative of the problem and contains features that could be useful for prediction.
6. Perform dimensionality reduction techniques if needed (e.g., PCA) to reduce computational complexity.
7. Choose appropriate machine learning algorithms for the problem (e.g., classifi cation algorithms such as logistic regression, decision trees, random forests, or more advanced methods like neural networks).
8. Experiment with different algorithms and evaluate their performance using ap propriate metrics (e.g., accuracy, precision, recall, F1-score).
9. Tunehyperparameters using techniques like grid search, random search, or Bayesian optimization to improve model performance.
10. Evaluate the trained models on the validation dataset to assess their generaliza tion performance.
11. Deploy the trained model into production environment, making sure it can han dle real-time or batch prediction requests.

4.3.2 Pseudo Code

```
import random_forest_library

function cyberAttackUsingRandomForest(target):
    // Phase 1: Reconnaissance
    gatheredInfo = reconnaissance(target)

    // Phase 2: Exploitation
    vulnerabilities = assessVulnerabilities(gatheredInfo)
    exploit(vulnerabilities)

    // Phase 3: Exfiltration
    sensitiveData = exfiltrateData()

    // Phase 4: Evasion/Covering Tracks
    coverTracks()
    destroyEvidence()

    return sensitiveData

function reconnaissance(target):
    // Gather information about the target using various sources
    gatheredInfo = gatherInformation(target)
    // Apply Random Forest algorithm to analyze and prioritize gathered information
    analyzedInfo = random_forest_analyze(gatheredInfo)

    return analyzedInfo

function assessVulnerabilities(gatheredInfo):
    // Use Random Forest algorithm to identify potential vulnerabilities
    vulnerabilities = random_forest_identifyVulnerabilities(gatheredInfo)
    // Prioritize vulnerabilities based on potential impact and exploitability
    prioritizedVulnerabilities = prioritize(vulnerabilities)

    return prioritizedVulnerabilities

function exploit(vulnerabilities):
    // Exploit identified vulnerabilities to gain unauthorized access
    for each vulnerability in vulnerabilities:
        if exploitSuccessful(vulnerability):
            establishAccess()

function exfiltrateData():
    // Extract sensitive data from the compromised system or network
    data = extractData()
    // Use Random Forest algorithm to obfuscate data exfiltration
    obfuscateData(data)

    return data
```

```
49
50 function coverTracks():
51     // Employ Random Forest algorithm to generate fake or misleading activity
52     generateFakeActivity()
53     // Modify logs and timestamps to hide traces of the attack
54     modifyLogs()
55
56 function destroyEvidence():
57     // Remove any remaining evidence of the attack
58     eraseLogs()
59     // Use Random Forest algorithm to predict potential forensic analysis techniques and counter
           them
60     counterForensicAnalysis()
61     def reconnaissance(target):
62     # Gather information about the target using various sources
63     gathered_info = gather_information(target)
64     # Apply Random Forest algorithm to analyze and prioritize gathered information
65     analyzed_info = random_forest_library.analyze(gathered_info)
66
67     return analyzed_info
68
69 def assess_vulnerabilities(gathered_info):
70     # Use Random Forest algorithm to identify potential vulnerabilities
71     vulnerabilities = random_forest_library.identify_vulnerabilities(gathered_info)
72     # Prioritize vulnerabilities based on potential impact and exploitability
73     prioritized_vulnerabilities = prioritize(vulnerabilities)
74
75     return prioritized_vulnerabilities
76
77 def exploit(vulnerabilities):
78     # Exploit identified vulnerabilities to gain unauthorized access
79     for vulnerability in vulnerabilities:
80         if exploit_successful(vulnerability):
81             establish_access()
82
83 def exfiltrate_data():
84     # Extract sensitive data from the compromised system or network
85     data = extract_data()
86     # Use Random Forest algorithm to obfuscate data exfiltration
87     obfuscate_data(data)
88
89     return data
90
91 def cover_tracks():
92     # Employ Random Forest algorithm to generate fake or misleading activity
93     generate_fake_activity()
94     # Modify logs and timestamps to hide traces of the attack
95     modify_logs()
```

4.4 Module Description:

4.4.1 Data Preprocessing

This module provides a structured approach to processing the data required for training a machine learning model to predict cyber hacking breaches. Adjustments may be needed based on the specific characteristics of your dataset and project re quirements. Splits the preprocessed data into training and testing sets using train tests-split from scikit-learn.

CSV file dataset-cyberhacking.dataset

4.4.2 Feature Selection

This module provides a simple feature selection method using the chi-squared test. You can adjust the scoring function and feature selection method based on the characteristics of your data and specific project requirements.Uses the SelectKBest method from scikit-learn with the chi-squared test as the scoring function to select the top num-features features.

4.4.3 Model Training

In this segment, the trained Trains different machine learning models using the preprocessed data. Includes tasks such as hyperparameter tuning and model evalu ation using cross-validation.Choose a set of candidate machine learning algorithms suitable for the problem, such as logistic regression, decision trees, random forests, support vector machines (SVM), or neural networks.

4.4.4 Prediction

In this segment, the trained model(s) to make predictions on the preprocessed new data. Obtain predicted probabilities or class labels depending on the problem (e.g., binary classification of hacking breaches).Format the predictions into a suitable format for further analysis or presentation. Depending on the application, this may involve storing predictions in a database, generating reports, or displaying them in a user interface.

4.4.5 Monitoring and Maintenance

Monitors the deployed model's performance in the production environment. Han dles model retraining and updates to keep the model up-to-date with new data and evolving threats.Trigger model retraining based on predefined criteria such as a de crease in model performance or significant data drift. Retrain the model periodically with new data to keep it up-to-date and effective. Use techniques like incremental learning to update the model without retraining from scratch.

4.4.6 Visualization

This component generates visualizations to explore the data, understand feature importance, and present model evaluation results. Helps in communicating insights and findings to stakeholders effectively. Use techniques such as bar charts, heatmaps, or permutation feature importance to rank and display the importance of each feature in predicting cyber hacking breaches.Use techniques such as bar charts, heatmaps, or permutation feature importance to rank and display the importance of each feature in predicting cyber hacking breaches.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc. Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens.

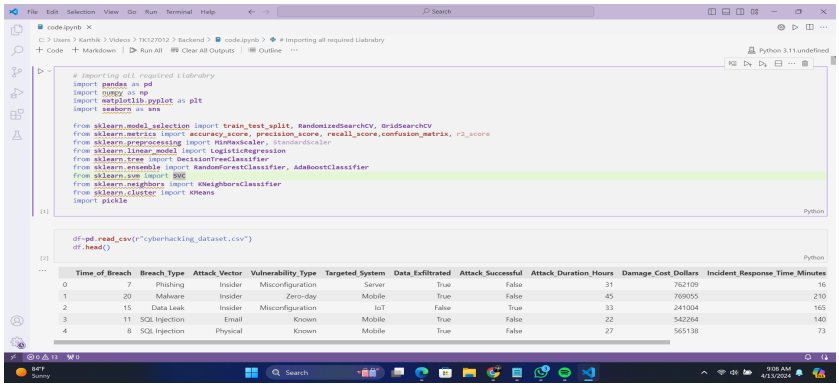

Figure 5.1: **Dataset Input Diagram**

Figure 5.1 shows that a comprehensive dataset comprising various features related to past breaches is essential. This dataset may include attributes such as the type of attack (e.g., malware, phishing), the target system or network, the methods used for infiltration and exfiltration, the duration of the breach, and the severity of the impact. Additionally, contextual information such as the industry sector, geographical location, and time of occurrence can provide valuable insights into the dynamics of cyber attacks. Gathering such data may involve collating information from incident reports, security logs, threat intelligence feeds, and other sources. Careful preprocessing and labeling of the dataset are crucial to ensure the quality and relevance of the data for training machine learning models.

5.1.2 Output Design

By incorporating these elements into the output design, you can create an effec tive and informative interface for presenting prediction results and insights from the cyber hacking breaches prediction . Adjustments and refinements may be neces sary based on the specific requirements and preferences of the stakeholders.Display the prediction results prominently on the dashboard interface. Present the predicted probability or class label for each observation in a tabular format or as a list.

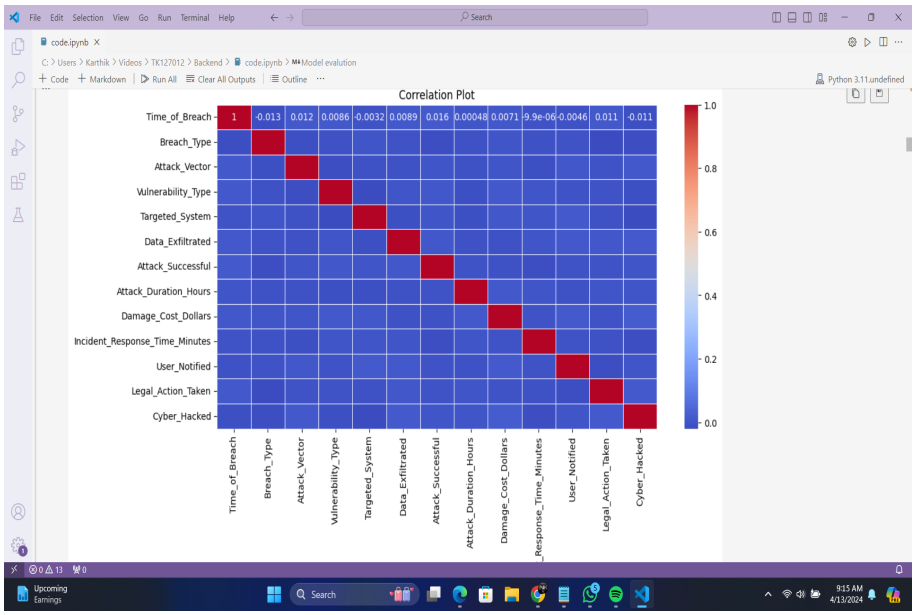

Figure 5.2: **Graph Representing Types of Breaches in Output Diagram**

Figure 5.2 shows that various types of attacks are observed, each with distinct characteristics and implications. Malware-based attacks involve the deployment of malicious software to compromise systems or networks, potentially leading to data theft or system disruption. Phishing attacks leverage deceptive tactics, such as fraudulent emails or websites, to trick individuals into revealing sensitive information or installing malware. Other common breach types include DDoS (Distributed Denial of Service) attacks, where malicious actors overwhelm targeted systems with excessive traffic, and SQL injection attacks, exploiting vulnerabilities in web applications to gain unauthorized access to databases. Machine learning techniques play a crucial role in detecting and mitigating these breaches by analyzing patterns and anomalies in data traffic, enhancing cybersecurity defenses against evolving threats.

5.2 Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement. Testing plays a crucial role in ensuring the reliabil ity, accuracy, and effectiveness of a machine learning-based cyber hacking breaches prediction.

5.3 Types of Testing

5.3.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal pro gram logic is functioning properly, and that program inputs produce valid outputs.All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an in dividual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive.

Unit tests perform basic tests at component level and test a specific business pro-cess, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

• **Define the input data**: Unit testing requires input data that can be used to test the functionality of each unit. In this case, the input data would include data about the breaches occured, the date, and other relevant features.

• **Define the output data**: The output data would be the predicted values of leaked data. This output data would be generated using random forest classifier.

Input

```
# Step 1: Import necessary libraries
import pandas as pd  # For data manipulation
from sklearn.model_selection import train_test_split  # For splitting data into train and test sets
from sklearn.ensemble import RandomForestClassifier  # For Random Forest classifier
from sklearn.metrics import accuracy_score  # For evaluating model accuracy

# Step 2: Load the dataset
dataset = pd.read_csv("cyber_hacking_dataset.csv")

# Step 3: Prepare the data
# Split the dataset into features (X) and target variable (y)
X = dataset.drop(columns=["breach_type"])  # Features (excluding the breach type)
y = dataset["breach_type"]  # Target variable (breach type)

# Step 4: Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 5: Initialize the Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)  # Using 100 decision
    trees

# Step 6: Train the Random Forest classifier
rf_classifier.fit(X_train, y_train)

# Step 7: Make predictions on the test set
predictions = rf_classifier.predict(X_test)

# Step 8: Evaluate the model accuracy
accuracy = accuracy_score(y_test, predictions)
print("Accuracy:", accuracy)
```

Test result

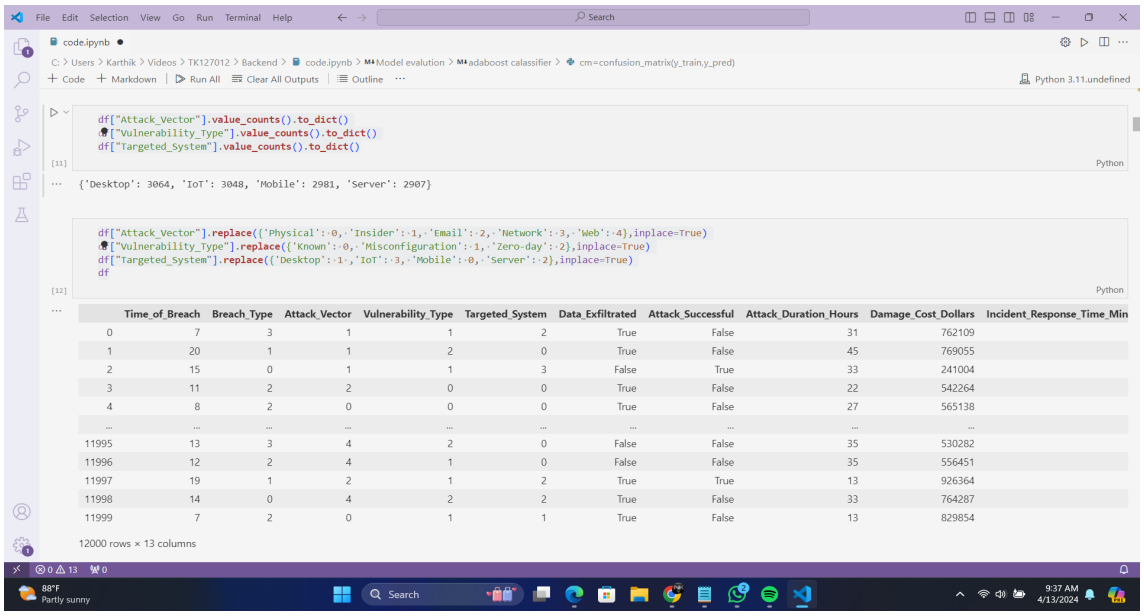

Figure 5.3: **Input for Unit Testing**

Figure 5.3 shows that the data preprocessing steps, such as handling missing values, encoding categorical variables, and scaling numerical features, are executed correctly. The processed data meets expected standards and is suitable for model training.Measure the computational efficiency and resource utilization of the model during training and inference. Model Training Unit T

5.3.2 Black Box Testing

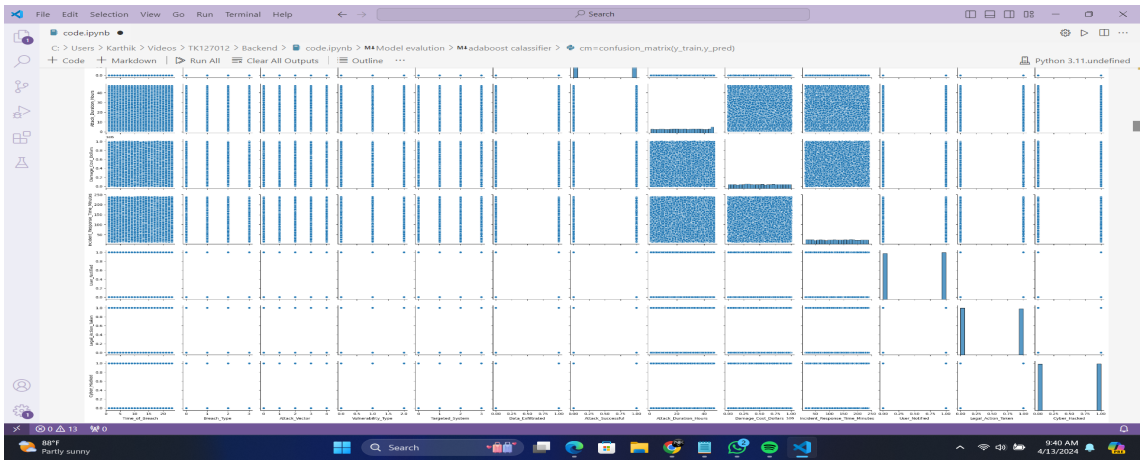

Figure 5.4: **Result of Black Box Testing**

Figure 5.4 illustrates the significance of black-box testing in evaluating the performance testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works

5.3.3 Test Result

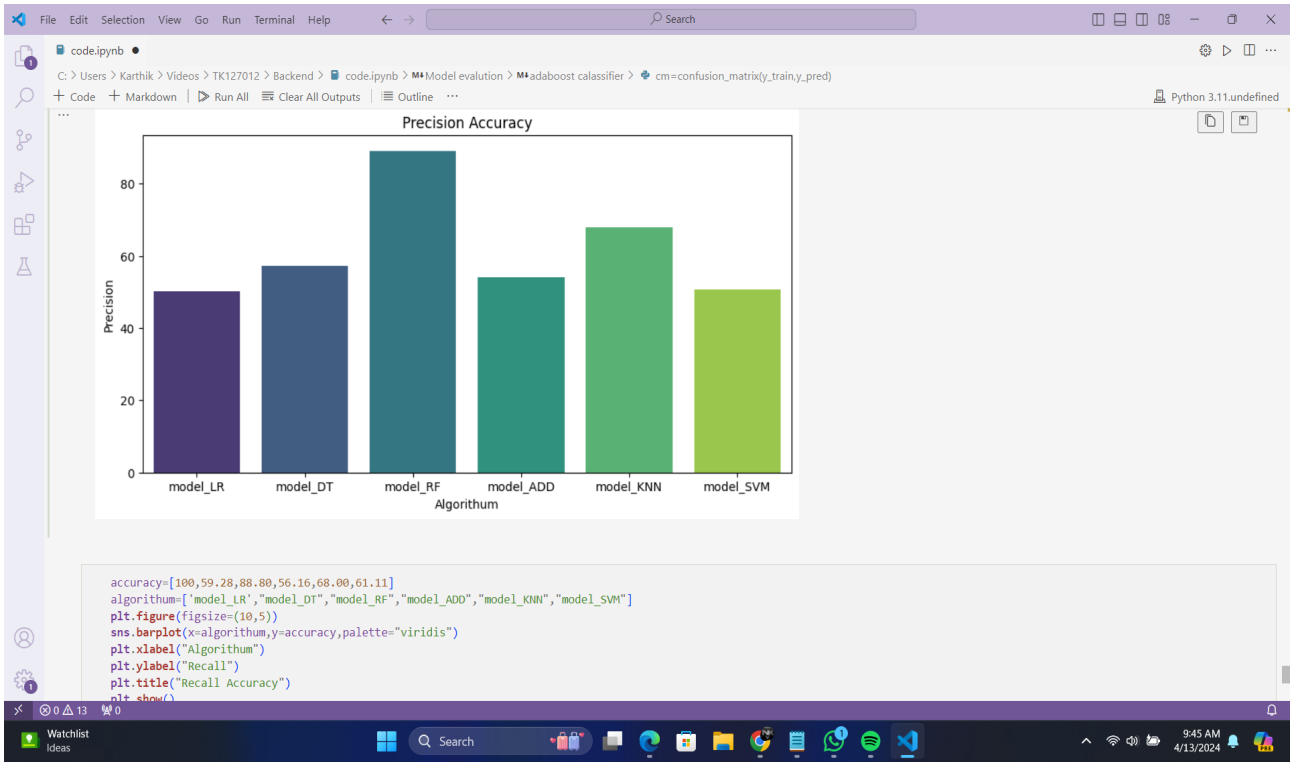

Figure 5.5: **Test Image**

Figure 5.5 summary of the model's performance on the test set, including the overall accuracy.Compare the predicted labels with the actual labels in the testing dataset to calculate accuracy. Accuracy = (Number of correct predictions) / (Total number of predictions) Analyze the accuracy score to understand how well the model performs. A higher accuracy score indicates that the model is making more correct predictions, while a lower accuracy score suggests that the model may need improvement.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system leverages machine learning algorithms, such as Random Forest, to analyze patterns and anomalies in data traffic, enabling the detection of potential cyber hacking breaches. The complexity of the machine learning model influences both computational requirements and predictive performance. Balancing model complexity with computational constraints is crucial for achieving a trade-off between accuracy and efficiency. Techniques such as model pruning, regularization, and algorithm optimization can help streamline model complexity without sacrificing predictive power.

The deployment architecture of the system influences its scalability, reliability, and operational efficiency. Cloud-based infrastructure, containerization, and microservices architecture enable flexible deployment and efficient resource allocation, supporting dynamic workloads and demand fluctuations. Additionally, leveraging distributed computing frameworks and parallel processing techniques can accelerate model training and inference, enhancing overall system efficiency.Performance metrics, system logs, and feedback loops facilitate real-time monitoring of model performance and resource utilization, enabling proactive identification of bottlenecks and inefficiencies. Adaptive algorithms, auto-scaling mechanisms, and periodic model retraining ensure that the system remains responsive and adaptive to evolving cyber threats and operational requirements.

Efficiency Metrics of Random Forest:

1. **Accuracy:** 0.9148888888888889
2. **Precision Score:** 0.926962457337884
3. **Recall Score:** 0.9015268864793096

6.2 Comparison of Existing and Proposed System

The existing system for detecting cyber hacking breaches may rely on traditional rule-based approaches, signature-based detection methods, or basic machine learning algorithms. While these systems may offer some level of protection, they often struggle to keep pace with the rapidly evolving nature of cyber threats.As the size and diversity of datasets increase, these systems may experience performance degradation and resource constraints, hindering their ability to provide timely and accurate threat detection.

The proposed system leverages advanced machine learning algorithms, such as Random Forest, to analyze vast amounts of data and adapt dynamically to evolving cyber threats. By learning from historical data and detecting subtle patterns and anomalies, the system can identify new and emerging attack vectors with greater accuracy and efficiency. Machine learning algorithms, trained on labeled datasets and validated against ground truth labels, can significantly reduce false positive rates compared to traditional rule-based systems. By leveraging advanced feature engineering, anomaly detection, and ensemble learning techniques, the proposed system can improve the accuracy and reliability of threat detection while minimizing false positives.System offers enhanced predictive capabilities, enabling the early detection of cyber threats and proactive mitigation of risks. By leveraging advanced algorithms such as Random Forest, deep learning, and anomaly detection, the system can identify subtle deviations from normal behavior and detect previously unseen attack patterns, enhancing cybersecurity posture and resilience.

Evaluation Metrics	Existing System(DT)	Proposed System (RF)
Accuracy	0.5607777777777778	0.9148888888888889
Precision Score	0.5568273092369478	0.926962457337884
Recall Score	0.6136313343660101	0.9015268864793096

Table 6.1: **Comparing Efficiency Metrics of Existing System and Proposed System**

6.3 Sample Code

```
# Importing all required Liabrabry
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, RandomizedSearchCV, GridSearchCV
from sklearn.metrics import accuracy_score, precision_score, recall_score,confusion_matrix, r2_score
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.cluster import KMeans
import pickle

df=pd.read_csv(r"cyberhacking_dataset.csv")
df.head()

Time_of_Breach                  0
Breach_Type                     0
Attack_Vector                   0
Vulnerability_Type              0
Targeted_System                 0
Data_Exfiltrated                0
Attack_Successful               0
Attack_Duration_Hours           0
Damage_Cost_Dollars             0
Incident_Response_Time_Minutes  0
User_Notified                   0
Legal_Action_Taken              0
Cyber_Hacked                    0
dtype: int64
Index(['Time_of_Breach', 'Breach_Type', 'Attack_Vector', 'Vulnerability_Type',
       'Targeted_System', 'Data_Exfiltrated', 'Attack_Successful',
       'Attack_Duration_Hours', 'Damage_Cost_Dollars',
       'Incident_Response_Time_Minutes', 'User_Notified', 'Legal_Action_Taken',
       'Cyber_Hacked'],
      dtype='object')
      df["Breach_Type"].value_counts().to_dict()
      df["Breach_Type"].replace({'Data Leak': 0, 'Malware': 1, 'SQL Injection': 2, 'Phishing': 3},
          inplace=True)
df
df["Attack_Vector"].value_counts().to_dict()
df["Vulnerability_Type"].value_counts().to_dict()
df["Targeted_System"].value_counts().to_dict()
```

```
dfdf["Attack_Vector"].replace({'Physical': 0, 'Insider': 1, 'Email': 2, 'Network': 3, 'Web': 4},
    inplace=True)
df["Vulnerability_Type"].replace({'Known': 0, 'Misconfiguration': 1, 'Zero-day': 2},inplace=True)
df["Ta
df["Data_Exfiltrated"].value_counts().to_dict()
df["Data_Exfiltrated"].replace({True: 1, False: 0},inplace=True)
df["Attack_Successful"].replace({True: 1, False: 0},inplace=True)
df["User_Notified"].replace({True: 1, False: 0},inplace=True)
df["Legal_Action_Taken"].replace({True: 1, False: 0},inplace=True)
df["Cyber_Hacked"].replace({True: 1, False: 0},inplace=True)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12000 entries, 0 to 11999
Data columns (total 13 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Time_of_Breach                  12000 non-null  int64
 1   Breach_Type                     12000 non-null  int64
 2   Attack_Vector                   12000 non-null  int64
 3   Vulnerability_Type              12000 non-null  int64
 4   Targeted_System                 12000 non-null  int64
 5   Data_Exfiltrated                12000 non-null  int64
 6   Attack_Successful               12000 non-null  int64
 7   Attack_Duration_Hours           12000 non-null  int64
 8   Damage_Cost_Dollars             12000 non-null  int64
 9   Incident_Response_Time_Minutes  12000 non-null  int64
 10  User_Notified                   12000 non-null  int64
 11  Legal_Action_Taken              12000 non-null  int64
 12  Cyber_Hacked                    12000 non-null  int64
dtypes: int64(13)

corr=df.corr()
plt.figure(figsize=(12, 6))
sns.heatmap(corr, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Plot')
plt.show()

plt.figure(figsize=(10,10))
sns.pairplot(df)
plt.show()

x=df.drop("Cyber_Hacked",axis=1)
y=df["Cyber_Hacked"]        ..
11995    0
11996    1
11997    1
11998    0
11999    1
Name: Cyber_Hacked, Length: 12000, dtype: int64
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=1,stratify=y)
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Output

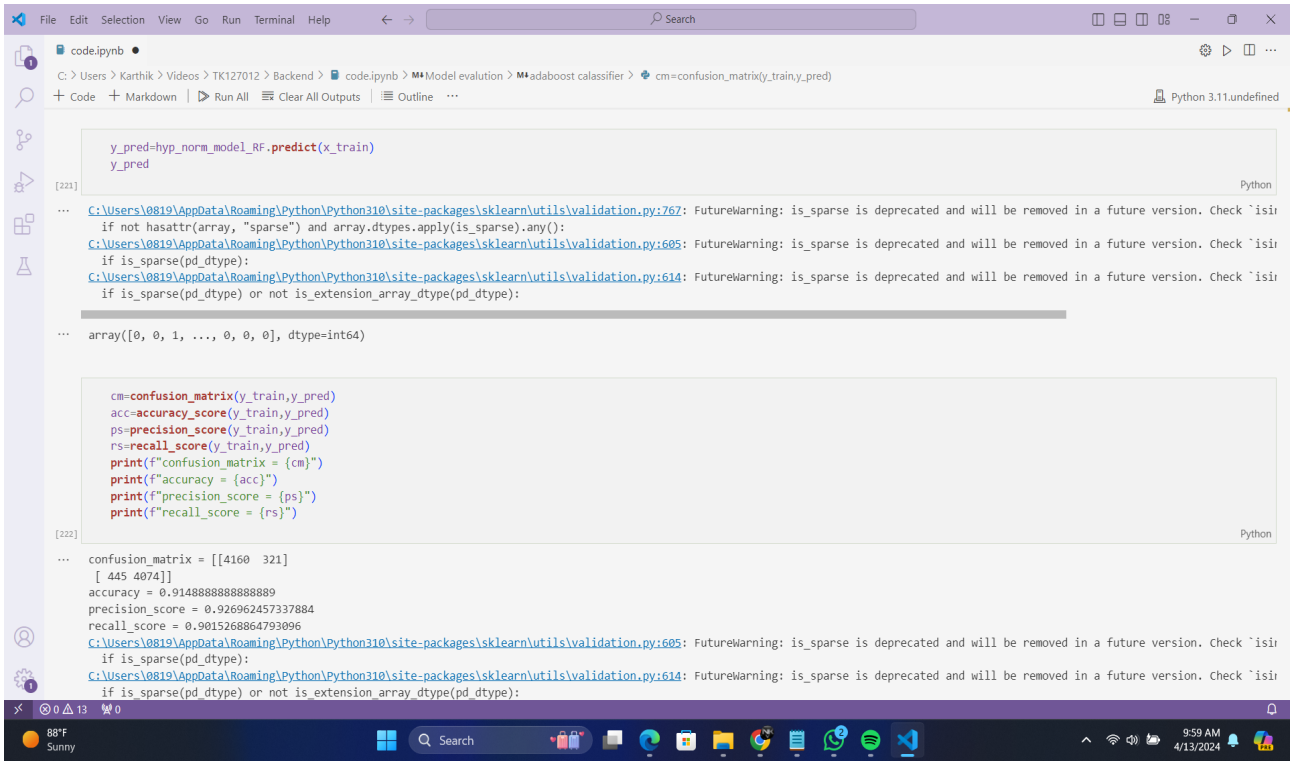

Figure 6.1: **Random Forest output Metrics**

Figure 6.1 shows that the model's performance on the test set, including the overall accu racy. A confusion matrix that shows how many images were correctly classified and misclassified for each class in the CIFAR-1 dataset. The diagonal elements of the matrix represent the number of correctly classified images, while the off-diagonal elements represent the number of misclassified images. A classification report that provides metrics such as precision, recall, and F1-score for each class in the CIFAR 10 dataset. These metrics give a more detailed understanding of the model's perfor mance in each class. A grid of images from the test set, along with their predicted labels and probabilities. The predicted labels are shown in bold, while the true labels are shown in normal font. The probabilities represent the confidence of the model in its predictions. A set of images that the model misclassified, along with their predicted labels and true labels. These images can help to identify areas where the model needs to improve.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The conclusion of the proposed approach for cyber hacking breaches prediction lies in its innovative utilization of machine learning algorithms to proactively iden tify and forecast potential cyber threats. Unlike traditional methods, this framework leverages advanced data analytics to discern subtle patterns and anomalies within vast datasets, enabling early detection of potential hacking breaches. The integration of machine learning models empowers the system to adapt and evolve with emerg ing cyber threats, enhancing its predictive capabilities. This novel approach not only contributes to the field of cybersecurity but also establishes a proactive and dynamic paradigm for anticipating and mitigating cyber risks, thus fortifying the resilience of digital ecosystems.The application of machine learning techniques for detecting cyber hacking breaches represents a significant advancement in cybersecurity capabilities. By leveraging the power of advanced algorithms such as Random Forest, deep learning, and anomaly detection, organizations can enhance their ability to detect, prevent, and mitigate cyber threats effectively.

The theoretical analysis highlights the key advantages of the proposed system, including its adaptability to evolving threats, scalability to handle large-scale datasets, and ability to reduce false positive rates while enhancing predictive capabilities. By transitioning from static rule-based systems to dynamic machine learning-based solutions, organizations can stay ahead of cyber adversaries and strengthen their cybersecurity posture

7.2 Future Enhancements

Future enhancements could involve integrating more advanced machine learning models, such as deep learning algorithms, to further improve the accuracy and ro bustness of cyber hacking breach predictions. Additionally, incorporating real-time data streams and leveraging anomaly detection techniques could enhance the timeli ness and effectiveness of threat detection. Furthermore, exploring ensemble learning methods to combine the strengths of multiple models could yield even more reli- able predictions. Moreover, developing automated response systems that can dynami cally adjust security measures based on predictive insights would bolster proactive defense strategies..

•**Integration of Advanced Algorithms:**Explore the integration of more advanced machine learning algorithms such as deep learning, ensemble methods, and gradi ent boosting algorithms. These algorithms may capture more complex patterns and relationships in the data, potentially leading to improved prediction accuracy.
•**Model Interpretability:** Enhance the interpretability of the machine learning mod- els to provide insights into how predictions are made. Techniques such as SHAP (SHapley Additive exPlanations) values, LIME (Local Interpretable Model agnos- tic Explanations), and model-agnostic interpretation methods can help explain the model's predictions to stakeholders.
Real-time Monitoring: Implement real-time monitoring capabilities to detect cyber hacking breaches as they occur or shortly thereafter. This involves integrat ing the prediction model into existing security systems and continuously analyzing incom- ing data streams for suspicious activity.

Chapter 8

INDUSTRY DETAILS

8.1 Industry name

ENCORA PVT.LTD

8.1.1 Duration of Internship (From Date - To Date)

17/01/2024 - 05/2024

8.1.2 Duration of Internship in months

6 months

8.1.3 Industry Address

Pallavaram-Thoraipakkam Road,Chennai.Tamil Nadu -600097

8.2 Internship Offer Letter

Dear ,

Nalluri Karthik

Reg ID- VTU_15337

Greetings from **Encora Innovation Pvt,Ltd**

Encora is Driven by a motto Digital Engineering for innovating at scale and enabling enterprise Modernization helps to provide companies and enterprises with complete IT Solutions to help improve their operations. Our expertise in IT Solutions helps us provide end-to-end support for IT Solutions, which includes planning, deploying, and maintaining the systems. Our ability to address customers' needs and concerns with diverse expectations and help them with new, refreshing ideas help us set ourselves apart. With the best blend of energy and expertise.the proffered partner for digitally native and enterprise clients

In reference to your application we would like to congratulate you on being selected for an internship for **"Associate Trainee"** with Encora Innovation Pvt Ltd. Your internship is scheduled to start effectively from **17-01-2024** for a Period of **120** days

The project details and technical platform will be shared with you on or before commencement of training.

Again, congratulations and we look forward to working with you.

You should report for training at the following address:

Reporting Office Address:

Pallavaram -Thoraipakkam Road, Chennai,Tamil Nadu- 600097

Contact Person: AnuhyaReddy

Contact Number- 9885488485

Dear ,

Kota Manish

Reg ID- VTU_17731

Greetings from **Encora Innovation Pvt,Ltd**

Encora is Driven by a motto Digital Engineering for innovating at scale and enabling enterprise Modernization helps to provide companies and enterprises with complete IT Solutions to help improve their operations. Our expertise in IT Solutions helps us provide end-to-end support for IT Solutions, which includes planning, deploying, and maintaining the systems. Our ability to address customers' needs and concerns with diverse expectations and help them with new, refreshing ideas help us set ourselves apart. With the best blend of energy and expertise.the proffered partner for digitally native and enterprise clients

In reference to your application we would like to congratulate you on being selected for an internship for **"Associate Trainee"** with Encora Innovation Pvt Ltd. Your internship is scheduled to start effectively from **17-01-2024** for a Period of **120** days

The project details and technical platform will be shared with you on or before commencement of training.

Again, congratulations and we look forward to working with you.

You should report for training at the following address:

Reporting Office Address:

Pallavaram -Thoraipakkam Road, Chennai,Tamil Nadu- 600097

Contact Person: AnuhyaReddy

Contact Number- 9885488485

Project Commencement Form

Name of the Industry: ENCORA

Address: Pallavaram -Thoraipakkam Road, Chennai,Tamil Nadu- 600097

Team Details:

S.No	ID No	Student Name	Degree & Branch
1.	VTU15337	NALLURI KARTHIK	B.TECH/CSE
2.	VTU17731	KOTA MANISH	

Date of reporting for project work: 10/02/2024

Name of the Industry Supervisor : Kuppam vasanthi

Department : IT

Designation : Internship Manager

Contact Number :9885488485

Email ID : Kuppamvasanthi@gmai.com

Name of the Internal Supervisor : Anuhya Reddy

Contact No. :9885488485

Email ID : encoraanuhya@gmail.com

Tentative Project Title / Project domain: Cyber hacking breaches using machine learning

Commencement Form

Chapter 9

PLAGIARISM REPORT

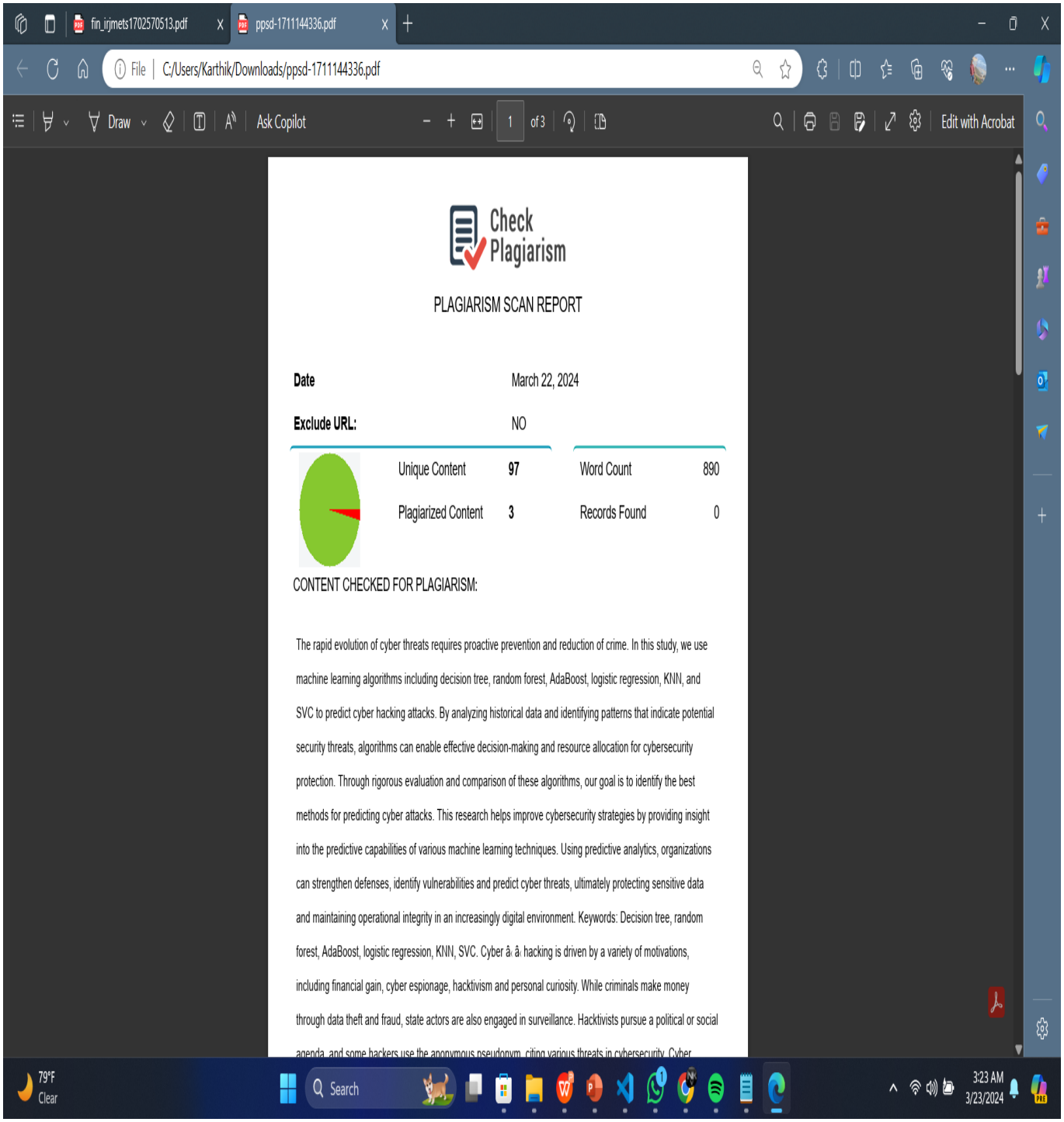

Figure 9.1: **Plagarism Report**

Chapter 10

SOURCE CODE & POSTER PRESENTATION

10.1 Source Code

```
from flask import Flask, render_template, request, url_for, flash, redirect,session
import pandas as pd
import numpy as np
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
import mysql.connector
db=mysql.connector.connect(user="root",password="",port='3306',database='cyber_attack')
cur=db.cursor()

app=Flask(__name__)
app.secret_key="CBJcb786874wrf78chdchsdcv"

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/about')
def about():
    return render_template('about.html')


@app.route('/login',methods=['POST','GET'])
def login():
    if request.method=='POST':
        useremail=request.form['useremail']
        session['useremail']=useremail
        userpassword=request.form['userpassword']
        sql="select * from user where Email='%s' and Password='%s'"%(useremail,userpassword)
```

```
        cur.execute(sql)
        data=cur.fetchall()
        db.commit()
        if data ==[]:
            msg="user Credentials Are not valid"
            return render_template("login.html",name=msg)
        else:
            return render_template("load.html",myname=data[0][1])
    return render_template('login.html')

@app.route('/registration',methods=["POST","GET"])
def registration():
    if request.method=='POST':
        username=request.form['username']
        useremail = request.form['useremail']
        userpassword = request.form['userpassword']
        conpassword = request.form['conpassword']
        Age = request.form['Age']
        address = request.form['address']
        contact = request.form['contact']
        if userpassword == conpassword:
            sql="select * from user where Email='%s' and Password='%s'"%(useremail,userpassword)
            cur.execute(sql)
            data=cur.fetchall()
            db.commit()
            print(data)
            if data==[]:

                sql = "insert into user(Name,Email,Password,Age,Address,Contact)values(%s,%s,%s,%s,%
                    s,%s)"
                val=(username,useremail,userpassword,Age,address,contact)
                cur.execute(sql,val)
                db.commit()
                flash("Registered successfully","success")
                return render_template("login.html")
            else:
                flash("Details are invalid","warning")
                return render_template("registration.html")
        else:
            msg = "Password doesn't match"
            return render_template("registration.html",msg=msg)
    return render_template('registration.html')

@app.route('/load',methods=["GET","POST"])
def load():
    global df, dataset
    if request.method == "POST":
        data = request.files['data']
        df = pd.read_csv(data)
        dataset = df.head(100)
```

```
        msg = 'Data Loaded Successfully'
        return render_template('load.html', msg=msg)
    return render_template('load.html')


@app.route('/view')
def view():
    print(dataset)
    print(dataset.head(2))
    print(dataset.columns)
    return render_template('view.html', columns=dataset.columns.values, rows=dataset.values.tolist()
        )


@app.route('/preprocess', methods=['POST', 'GET'])
def preprocess():
    global x, y, x_train, x_test, y_train, y_test,  hvectorizer,df
    if request.method == "POST":
        size = int(request.form['split'])
        size = size / 100
        dataset["Breach_Type"].replace({'Data Leak': 0, 'Malware': 1, 'SQL Injection': 2, 'Phishing'
            : 3},inplace=True)
        dataset["Attack_Vector"].replace({'Physical': 0, 'Insider': 1, 'Email': 2, 'Network': 3, '
            Web': 4},inplace=True)
        dataset["Vulnerability_Type"].replace({'Known': 0, 'Misconfiguration': 1, 'Zero-day': 2},
            inplace=True)
        dataset["Targeted_System"].replace({'Desktop': 1 ,'IoT': 3, 'Mobile': 0, 'Server': 2},
            inplace=True)
        dataset["Data_Exfiltrated"].replace({True: 1, False: 0},inplace=True)
        dataset["Attack_Successful"].replace({True: 1, False: 0},inplace=True)
        dataset["User_Notified"].replace({True: 1, False: 0},inplace=True)
        dataset["Legal_Action_Taken"].replace({True: 1, False: 0},inplace=True)
        dataset["Cyber_Hacked"].replace({True: 1, False: 0},inplace=True)

       # Assigning the value of x and y
        x=dataset.drop("Cyber_Hacked",axis=1)
        y=dataset["Cyber_Hacked"]
        x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=size,random_state=1,stratify=y)

        # describes info about train and test set
        print("Number transactions X_train dataset: ", x_train.shape)
        print("Number transactions y_train dataset: ", y_train.shape)
        print("Number transactions X_test dataset: ", x_test.shape)
        print("Number transactions y_test dataset: ", y_test.shape)


        print(x_train,x_test)

        return render_template('preprocess.html', msg='Data Preprocessed and It Splits Successfully'
            )
```

```
    return render_template('preprocess.html')

@app.route('/model', methods=['POST', 'GET'])
def model():
    if request.method == "POST":
        global model
        print('ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc')
        s = int(request.form['algo'])
        if s == 0:
            return render_template('model.html', msg='Please Choose an Algorithm to Train')
        elif s == 1:
            print('aaaaaaaaaaaaaaaaaaaaabbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb')
            from sklearn.ensemble import AdaBoostClassifier
            ad = AdaBoostClassifier()
            ad.fit(x_train,y_train)
            y_pred = ad.predict(x_train)
            ac_ad = accuracy_score(y_train, y_pred)
            ac_ad = ac_ad * 100
            print('aaaaaaaaaaaaaaaaaaaaaaaa')
            msg = 'The accuracy obtained by AdaBoost Classifier is  ' + str(ac_ad) + str('%')
            return render_template('model.html', msg=msg)
        elif s == 2:
            classifier = DecisionTreeClassifier(max_leaf_nodes=39, random_state=0)
            classifier.fit(x_train, y_train)
            y_pred  =  classifier.predict(x_train)

            ac_dt = accuracy_score(y_train, y_pred)
            ac_dt = ac_dt * 100
            msg = 'The accuracy obtained by Decision Tree Classifier is ' + str(ac_dt) + str('%')
            return render_template('model.html', msg=msg)
        elif s == 3:
            from sklearn.ensemble import RandomForestClassifier
            rf=RandomForestClassifier(n_estimators = 50,
                                      min_samples_split = 3,
                                      min_samples_leaf = 2,
                                      max_features = 'log2',
                                      max_depth = 10,
                                      bootstrap = True)
            rf.fit(x_train,y_train)
            rf=rf.fit(x_train,y_train)
            y_pred  =  rf.predict(x_train)

            ac_rf = accuracy_score(y_train, y_pred)
            ac_rf = ac_rf * 100
            msg = 'The accuracy obtained by random Forest Classifier is ' + str(ac_rf) + str('%')
            return render_template('model.html', msg=msg)
        elif s == 4:
            from sklearn.neighbors import KNeighborsClassifier
            knn = KNeighborsClassifier(n_neighbors=12)
            knn.fit(x_train, y_train)
```

```
            y_pred  =  knn.predict(x_train)

            ac_knn = accuracy_score(y_train, y_pred)
            ac_knn = ac_knn * 100
            msg = 'The accuracy obtained by K-Nearest Neighbour is ' + str(ac_knn) + str('%')
            return render_template('model.html', msg=msg)
        elif s == 5:
            svc = SVC()
            svc.fit(x_train, y_train)
            y_pred  =  svc.predict(x_train)

            ac_svc = accuracy_score(y_train, y_pred)
            ac_svc = ac_svc * 100
            msg = 'The accuracy obtained by support vector Classifier is ' + str(ac_svc) + str('%')
            return render_template('model.html', msg=msg)
        elif s == 6:
            from sklearn.linear_model import LogisticRegression

            lr = LogisticRegression()
            lr.fit(x_train, y_train)
            y_pred  =  lr.predict(x_train)

            ac_lr = accuracy_score(y_train, y_pred)
            ac_lr = ac_lr * 100
            msg = 'The accuracy obtained by Logistic Regression is ' + str(ac_lr) + str('%')
            return render_template('model.html', msg=msg)
    return render_template('model.html')

@app.route('/prediction', methods=["GET", "POST"])
def prediction():
    if request.method == "POST":
        # f1=int(request.form['city'])
        f1 = float(request.form['Time_of_Breach'])
        f2 = float(request.form['Breach_Type'])
        f3 = float(request.form['Attack_Vector'])
        f4 = float(request.form['Vulnerability_Type'])
        f5 = float(request.form['Targeted_System'])
        f6 = float(request.form['Data_Exfiltrated'])
        f7 = float(request.form['Attack_Successful'])
        f8 = float(request.form['Attack_Duration_Hours'])
        f9 = float(request.form['Damage_Cost_Dollars'])
        f10 = float(request.form['Incident_Response_Time_Minutes'])
        f11 = float(request.form['User_Notified'])
        f12 = float(request.form['Legal_Action_Taken'])


        print(f2)
        print(type(f2))

        li = [f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12]
```

```
        print(li)

        # model.fit(X_transformed, y_train)

        # print(f2)
        # import pickle
        # filename = 'randomforest.sav'
        # model = pickle.load(open(filename, 'rb'))
        # result = model.predict([li])
        from sklearn.ensemble import RandomForestClassifier
        rf=RandomForestClassifier(n_estimators = 50,
                                  min_samples_split = 3,
                                  min_samples_leaf = 2,
                                  max_features = 'log2',
                                  max_depth = 10,
                                  bootstrap = True)
        rf.fit(x_train,y_train)
        rf=rf.fit(x_train,y_train)
        result =  rf.predict([li])
        print(result)
        print('result is ',result)
        # (Normal  = 0,   Cyber_Hacked  = 1 )
        if result == 0:
            msg = 'There is  No-Cyber Hacked'
            return render_template('prediction.html', msg=msg)
        else:
            msg = 'There is Cyber Hacked'
            return render_template('prediction.html', msg=msg)
    return render_template('prediction.html')









if __name__=='__main__':
    app.run(debug=True)
```

10.2 Poster Presentation

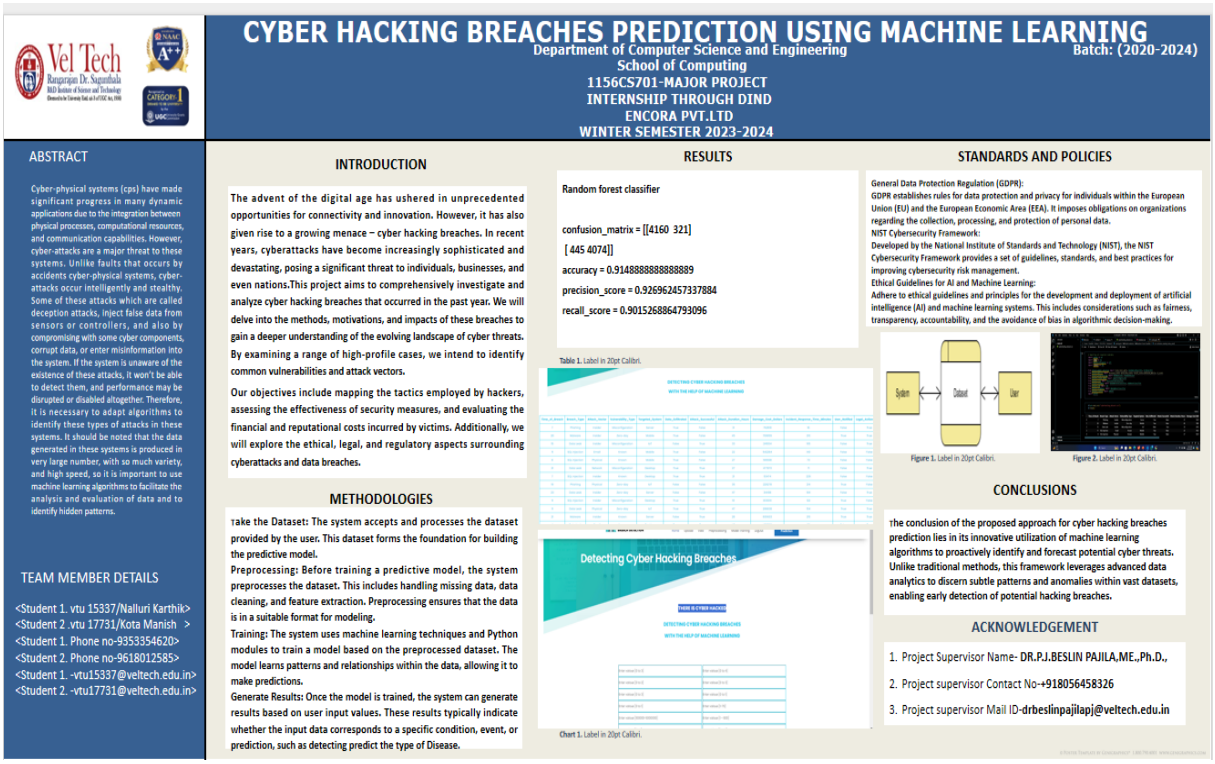

Figure 10.1: **Poster Presentation**

References

[1] Kure, H.I., Islam, S., Ghazanfar, M. et al. Asset criticality and risk prediction for an effective cybersecurity risk management of cyber-physical system. Neu 47 ral Comput Applic 34, 493–514 (2022). PP. 1-6, 2021.

[2] Kwon, Cheolhyeon, Weiyi Liu, and Inseok Hwang. ”Security analysis for cyber physical systems against stealthy deception attacks.” In 2013 American control conference, IEEE (2013): 3344-3349,2013

[3] Mandal, S., Saha, B., Nag, R. (2020). Exploiting Aspect-Classified Sentiments for CyberCrime Analysis and Hack Prediction. In: Kar, N., Saha, A., Deb, S. (eds) Trends in Computational Intelligence, Security and Internet of Things. IC CISIoT 2020. Communications in Computer and Information Science, vol 1358.PP. 1542-1552, 2020.

[4] M. Eling and W. Schnell, “What can we realize cyber risk and cyber risk insur ance?” J. Risk Finance, vol. 17, no. 5, pp. 474–491, 2016.

[5] Pajic, Miroslav, James Weimer, Nicola Bezzo, Oleg Sokolsky, George J. Pappas, and Insup Lee. ”Designandimplementationof attack-resilient cyberphysical sys tems: With a focus on attack-resilient state estimators.” IEEE Control Systems Magazine 37, no. 2 (2017): 66-81.

[6] Poyraz, O.I., Canan, M., McShane, M. et al. Cyber assets at risk: monetary impact of U.S. personally identifiable information mega data breaches. Geneva Pap Risk Insur Issues Pract 45, 616–638 (2020).

[7] R. R.Subramanian, R.Avula, P. S. SuryaandB.Pranay, ”ModelingandPredict ing Cyber Hacking Breaches,” 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 288-293

[8] Sun, Hongtao, Chen Peng, Taicheng Yang, Hao Zhang, and Wangli He. “Re silient control of networked control systems with stochastic denial of service attacks.” Neurocomputing 270 (2017): 170-177.

[9] Sheng, Long, Ya-Jun Pan, and Xiang Gong. ”Consensus formation control for a class of networked multiple mobile robot systems.” Journal of Control Science and Engineering 2012

[10] Zeng, Wente, and Mo-Yuen Chow. “Resilient distributed control in the presence of misbehaving agents in networked control systems.” IEEE transactions on cy bernetics 44, no. 11 (2014): 2038-2049.