CYBER HACKING BREACHING PREDICTION AND DETECTION USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

BALA VISHWATHARAN G	**913320104013**
AMEED HARSAI U	**913320104006**
SRI RAM MAHADEVAN S	**913320104049**
VISHVA J	**913320104055**

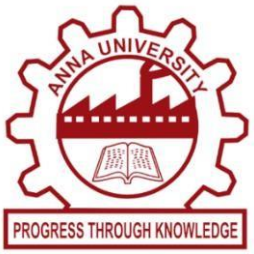
in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



VAIGAI COLLEGE OF ENGINEERING

ANNA UNIVERSITY:: CHENNAI 600025

MAY 2024

ANNA UNIVERSITY : : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"CYBER HACKING BREACHING PREDICTION AND DETECTION USING MACHINE LEARNING"** is the bona fide work of **"BALA VISHWATHARAN G 913320104013, AMEED HARSAI U913320104006, SRI RAM MAHADEVAN S 913320104049, VISHVA J 913320104055"** who carried out the work under my supervision.

SIGNATURE

Mrs.S.KAYALVIZHI,M.E., (Phd),
HEAD OF THE DEPARTMENT
ASSISTANT PROFESSOR
COMPUTER SCIENCE & ENGINNERING
VAIGAI COLLEGE OF ENGINEERING
THERKUTHERU, MADURAI-625 122

SIGNATURE

Mrs.J.VIJAYALAKSHMI,M.E
SUPERVISOR
ASSISTANT PROFESSOR
COMPUTER SCIENCE & ENGINEERING
VAIGAI COLLEGE OF ENGINEERING
THERKUTHERU, MADURAI-625 12

Project viva voice held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our gratefulness to our respectable management for having offered as the golden opportunity to do the project work in this prestigious institution.

We express our sincere thanks to our respected Principal Dr.R.Sivaranjani, Ph.D., for providing more facilities, to do this project work.

We sincerely thanks Mrs.S.Kayalvizhi, M.E., (Phd), Assistant Professor and Head of the Department of Computer Science and Engineering, who inspired us and gave us time to make this project to work a grand success.

We are deeply grateful to our Project coordinator Mrs. J.Vijayalakshmi, M.E., Assistant Professor, Department of Computer Science and Engineering for her valuable guidance and encouragement throughout the project.

We extend our heartfelt thanks and profound gratitude to all faculty members of our department for their kind help during our project work.

Finally we express our sincere thanks to our parents, who have constantly encouraged us and for being the source of encouraging spirits throughout our course.

ABSTRACT

The "Detection of Cyber Crime Breach using Machine Learning" project aims to develop a proactive solution for identifying and forecasting cybercrime breaches. By combining time series analysis with SARIMAX modeling, the project analyzes historical cybercrime data to forecast future breach occurrences. Innovative methodologies for data preprocessing, feature engineering, and model optimization enhance accuracy and reliability compared to existing systems. The project offers customizable solutions tailored to different organizational needs and provides cost-effective alternatives to proprietary systems. Rigorous evaluation tests and real-world case studies demonstrate superior performance in enhancing cyber security measures and empowering organizations to stay ahead of evolving threats.

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In today's interconnected world, every system is at risk from cyber-attacks. Malicious actors use powerful automated technologies, and new threats emerge constantly, making cybersecurity maintenance challenging. One of the significant threats is malware, software intentionally designed to damage computer systems or networks for espionage or financial gain. Modern malware is complex and can alter its code and behaviour to avoid detection. It targets embedded computational platforms like IoT devices, medical equipment, and E&I control systems. Given the vast range of malicious software, relying solely on signature-based protections is insufficient. Machine learning methodologies can use static and behavioural artifacts from malware families to represent the ever-changing structure of modern malware. This allows for the identification of complex malware attacks, including zero-day malware, which would otherwise evade signature-based detection methods. Deep learning algorithms can further enhance feature extraction and representation. Today's cybersecurity state requires an ongoing process of collecting and comparing millions of data points across your organization's people and infrastructure. Human efforts alone are insufficient; machine learning assistance is needed to identify patterns and anticipate potential security issues in large datasets. Human expertise remains crucial in cybersecurity operations, but the sheer volume and complexity of data make it impractical for humans alone to detect and respond to threats effectively.

1.2 OBJECTIVE

The objectives of the project is to analyze time series data related to data breaches, specifically focusing on breaches categorized as "HACK." This analysis likely involves exploring trends, seasonality, and patterns in the frequency and magnitude of data breaches over time.Modeling and Forecasting: The project involves developing predictive models, such as SARIMAX (Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors), to forecast future trends in data breaches. By fitting SARIMAX models to historical data and evaluating their performance, the project seeks to generate accurate forecasts of the number of breaches expected in the future.

Parameter Optimization: The project includes the optimization of model parameters, such as the order and seasonal order of the SARIMAX model. This optimization process likely involves conducting grid searches or other techniques to identify the most suitable parameter combinations that minimize information criteria like AIC (Akaike Information Criterion) or BIC (Bayesian Information Criterion).Evaluation and Visualization: The project aims to evaluate the performance of the developed models and visualize the results. This involves comparing the forecasts generated by the models with the observed data, assessing the accuracy of predictions, and presenting the findings through visualizations such as time series plots and forecasted vs. actual comparisons .The project's primary objectives revolve around analyzing historical data breaches, developing accurate predictive models, optimizing model parameters, and evaluating model performance to provide insights into future trends in data breaches and inform decision-making processes related to cybersecurity and risk management.

1.3 SCOPE OF THE PROJECT

The project involves the development of a robust time series analysis and forecasting system for cybersecurity risk management. This system utilizes advanced techniques such as SARIMAX modeling and parameter optimization to analyze historical data breaches, predict future trends, and mitigate associated risks effectively.

Key components of the project include data preprocessing, model development, evaluation, visualization, and documentation.

1. Cybersecurity Risk Management:

The developed system can be used by organizations to proactively manage cybersecurity risks by predicting future data breach occurrences and identifying potential vulnerabilities. This allows organizations to implement targeted security measures and allocate resources efficiently to mitigate risks.

2. Incident Response Planning:

The forecasting system can assist organizations in developing incident response plans by providing insights into potential future threats and enabling proactive measures to minimize the impact of data breaches. This ensures organizations are well-prepared to respond effectively to cybersecurity incidents.

3. Regulatory Compliance:

The system can help organizations comply with regulatory requirements related to data security and privacy, such as GDPR, HIPAA, or PCI DSS. By accurately forecasting data breach risks, organizations can implement necessary controls and safeguards to maintain compliance with regulatory standards.

4. Insurance and Risk Assessment:

Insurance companies can utilize the forecasting system to assess cybersecurity risks associated with potential policyholders. By analyzing historical data and predicting future trends, insurers can determine appropriate premiums and coverage levels for cyber insurance policies.

5. Strategic Planning and Decision Making:

The insights provided by the forecasting system can inform strategic planning and decision-making processes within organizations. Executives and stakeholders can use the forecasts to allocate resources, prioritize investments in cybersecurity, and develop long-term risk management strategies.

6. Enhanced Security Measures:

As the forecasting system evolves and becomes more sophisticated, it can contribute to the development of advanced security measures and technologies for preventing and detecting cyber threats. This includes the integration of machine learning and artificial intelligence algorithms for real-time threat detection and response.

7. Adaptation to Emerging Threats:

The forecasting system can adapt to emerging cybersecurity threats and evolving attack vectors by continuously analyzing new data and updating predictive models. This ensures organizations remain resilient in the face of emerging cyber risks and can respond effectively to new challenges.

8. Industry-wide Impact:

The insights and methodologies developed through this project can have a broader industry-wide impact by contributing to the advancement of cybersecurity practices and standards. Sharing findings and best practices with the cybersecurity community can help raise awareness and improve overall cybersecurity posture across various sectors.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

In recent years, the proliferation of digital technologies has revolutionized the way individuals and organizations interact, communicate, and conduct business. However, this digital transformation has also given rise to new and sophisticated forms of criminal activity known as cybercrime. Cybercriminals exploit vulnerabilities in computer systems, networks, and software to steal sensitive information, disrupt services, and cause financial harm. The increasing frequency and severity of cyberattacks have underscored the urgent need for effective strategies to detect, prevent, and mitigate cyber threats.

The field of cybersecurity encompasses a wide range of disciplines, methodologies, and technologies aimed at safeguarding digital assets and protecting against cyber threats. One area of particular interest is the use of machine learning techniques for cybercrime detection and prediction. Machine learning leverages algorithms and statistical models to analyze large volumes of data, identify patterns, and make predictions without explicit programming instructions. In the context of cybersecurity, machine learning offers a promising approach to enhance the capabilities of traditional security systems and address the evolving nature of cyber threats.

This literature review seeks to provide a comprehensive overview of existing research on cybercrime detection and prediction using machine learning. By synthesizing findings from previous studies and examining current methodologies and approaches, this review aims to identify key trends, challenges, and opportunities in the field. The review will cover various aspects of cybercrime detection and prediction, including the types of machine learning algorithms used, evaluation metrics and performance benchmarks, challenges and open research questions, and future directions for research and innovation.

2.1.1 Definition and Types of Cybercrimes

Cybercrime refers to criminal activities that are carried out using computers, networks, or digital technologies. These activities are typically aimed at exploiting vulnerabilities in digital systems to gain unauthorized access, steal sensitive information, disrupt services, or cause financial harm. Some of the examples of common types such as phishing, malware attacks, data breaches, etc.

2.1.2 Importance of Detection and Prediction

Early detection and prediction play a crucial role in mitigating cyber threats by allowing organizations and security professionals to proactively identify and respond to potential attacks before they can cause significant harm.

By spotting suspicious activities or anomalies in real-time, security teams can swiftly contain threats, minimizing data loss, downtime, and productivity disruptions. This proactive approach also contributes to improving cyber resilience by continuously monitoring for vulnerabilities and adapting defenses to evolving threats. Machine learning further amplifies these benefits by offering scalability, accuracy, automation, adaptability, and real-time detection capabilities.

2.2 Historical Perspective

2.2.1 Brief History of Research in Cybercrime Detection and Prediction

The history of research in cybercrime detection and prediction traces back to the early days of computing when the internet was in its infancy. In the 1980s and 1990s, as the internet became more widespread, cybercriminals began to exploit vulnerabilities in computer systems and networks for financial gain, espionage, and disruption. During this time, research efforts primarily focused on understanding the nature of cyber threats and developing rudimentary security measures such as firewalls, antivirus software, and intrusion detection systems (IDS).

2.2.2 Key Milestones and Breakthroughs

1990s: The emergence of viruses and worms, such as the Morris Worm in 1988, highlighted the need for more advanced cybersecurity solutions. Researchers began exploring the use of statistical analysis and machine learning techniques for detecting and mitigating cyber threats.

Early 2000s: The proliferation of broadband internet and the increasing interconnectedness of digital systems led to a surge in cybercrime activities, including phishing attacks, identity theft, and data breaches. Researchers started experimenting with machine learning algorithms for anomaly detection, pattern recognition, and predictive analytics in cybersecurity.

Mid to Late 2000s: Breakthroughs in machine learning, particularly in the field of artificial neural networks and deep learning, paved the way for more sophisticated cybercrime detection and prediction models. Researchers developed novel techniques for analyzing large-scale datasets, extracting meaningful features, and training complex neural networks to recognize subtle patterns indicative of cyber threats.

2010s to Present: The rise of big data and cloud computing has further accelerated research in cybercrime detection and prediction. Advanced machine learning algorithms, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and ensemble methods, have been applied to a wide range of cybersecurity applications, including malware detection, intrusion detection, fraud detection, and threat intelligence analysis.

2.3 Machine Learning Algorithms in Cybersecurity

2.3.1 Review of Existing Literature

A comprehensive survey of current literature reveals a diverse range of machine learning-based techniques for cybercrime detection and prediction. Researchers have explored various methodologies and algorithms, including supervised learning, unsupervised learning, and deep learning approaches, to address different aspects of cybersecurity challenges.

2.3.2 Methodologies and Algorithms

Supervised Learning: Supervised learning techniques involve training machine learning models on labeled datasets, where each data point is associated with a known outcome or class label. Common supervised learning algorithms used in cybercrime detection include decision trees, random forests, support vector machines (SVM), and naive Bayes classifiers.

Unsupervised Learning: Unsupervised learning techniques aim to identify patterns and anomalies in unlabeled datasets without prior knowledge of the classes or categories. Clustering algorithms such as k-means, hierarchical clustering, and density-based clustering are commonly used for anomaly detection and behavior profiling in cybersecurity.

Deep Learning: Deep learning techniques, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have gained prominence in cybercrime detection due to their ability to learn complex patterns and representations from raw data. Deep learning models have been applied to various cybersecurity tasks, including malware detection, network intrusion detection, phishing detection, and fraud detection.

2.4 Challenges and Open Research Questions

2.4.1 Data Quality and Availability

Cybercrime detection models heavily rely on the quality and availability of data for training and testing. However, several challenges impede the effectiveness of these models:

- Data Quality
- Imbalance
- Limited Availability of Labelled data

2.4.2 Adversarial Attacks and Evasion Techniques

Threat of Adversarial Attacks:

Adversarial attacks pose a significant threat to machine learning-based cybercrime detection systems by exploiting vulnerabilities in the model's decision boundaries. Adversaries can craft malicious inputs specifically
designed to evade detection or mislead the model into making incorrect prediction. Common adversarial attack techniques include such as Adversarial Perturbations, Model Evasion, Poisoning Attacks.

Defense Mechanisms Against Adversarial Attacks:

To enhance the robustness of cybercrime detection models against adversarial manipulation, several defense mechanisms such as Adversarial Training: Adversarial training involves augmenting the training data with adversarial examples and retraining the model to improve its resilience against adversarial attacks.

2.5 LITERATURE REVIEW TABLE

Author	Title	Advantages	Disadvantages
E. Gandotra, D. Bansal, and S. Sofat	Malware analysis and classification:A survey	Provides a comprehensive survey on malware analysis and classification,offering insights into various methodologies and techniques employed in the field.	It lacks recent advancements in machine learning techniques for malware detection due to its publication date.
W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li	A deep learning framework for intelligentmalware detection	Introduces EMBER, an opendataset for training static PEmalware models, facilitating research and development in the field of malware detection.	May not discuss the limitations or biases inherent in the EMBER dataset.
A. Shalaginov, S. Banin, A. Dehghantanha, and K. Franke	Machine learning aided static malware analysis: Asurvey and tutorial	Offers a comprehensive survey and tutorial on machine learning aided static malware analysis, providing valuable insights	Might lack practical implementation details and empirical evaluations of the techniques discussed.

		into leveraging machine learning for static analysis.	
R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi	Microsoft Malware Classification Challenge	Offers insights into the latest advancements in malware classification through participation and analysis of the Microsoft Malware Classification Challenge.	May not provide detailed methodologies or comparative analyses of different approaches.
M. Egele, T. Scholte, E. Kirda, and C. Kruegel	A Survey on Automated Dynamic Malware Analysis Techniques and Tools	Provides a comprehensive survey on automated dynamic malware analysis techniques and tools, offering a thorough understanding of dynamic analysis methodologies.	Might not focus specifically on machine learning techniques for dynamic malware analysis.

S. Anderson and P. Roth	EMBER: An Open Datasetfor Training Static PE Malware	Introduces EMBER, an opendataset for training static PEmalware models, facilitating research and development in the field of malware detection.	May not discuss the limitations or biases inherent in the EMBER dataset.
T. Morgenstern	Malware Terms for Non-TechiesCode Entropy	Offers insights into malware terminology and concepts like code entropy, providing valuable knowledge for both technical andnon-technical audiences.	Might lack in-depth technical discussions or empirical evaluations related to machine learning for malware detection.

CHAPTER 3

SYSTEM ANALYSIS

System analysis is "the process of studying a procedure or business to identify project goals and purposes and create systems and procedures that will efficiently achieve them".

3.1 EXISTING SYSTEM

Intrusion Detection Systems (IDS):

Intrusion Detection Systems (IDS) are security tools designed to detect and alert about unauthorized access or suspicious activities within a network or system. IDS maintains a pool of information regarding user profiles, hosts, connections, protocols, and devices, and uses this information to identify malicious signatures or threat patterns. Popular IDS solutions include Snort, Suricata, and Bro IDS.

Firewalls and Monitors:

Firewalls and monitoring systems are essential components of network security infrastructure, responsible for blocking unauthorized access attempts and monitoring network traffic for potential threats. Firewalls prevent unauthorized access to a network by analyzing incoming and outgoing traffic and enforcing predetermined security rules. Common firewall solutions include Cisco ASA, Palo Alto Networks, and Check Point Firewall.

Machine Learning Algorithms:

Machine learning algorithms are increasingly being used to detect and identify patterns related to Cyber hacking breaches, enabling automated prediction and detection of security threats. Classification techniques such as logistic regression, decision tree learning, support vector machines, and neural networks are employed to identify masqueraders or unauthorized users based on their access patterns.

Security Algorithms:

Various encryption methods, firewall blocking, and security protocols like honeypots are employed to protect information from unauthorized access and hacking breaches. Encryption ensures data confidentiality, integrity, and authenticity, while firewalls and security protocols help in detecting and mitigating Cyber threats. Encryption algorithms like AES, RSA, and SHA are widely used to secure data transmission and storage, while honeypots are deployed to lure and track potential attackers.

3.2 PROPOSED SYSTEM

The proposed system aims to leverage machine learning techniques, specifically time series analysis with SARIMAX modelling, to detect and forecast Cyber Crime breaches. With the increasing threat of Cyber Attacks and breaches in today's digital landscape, there is a critical need for proactive measures to detect and mitigate Cyber threats.

The proposed system seeks to address this need by providing a predictive tool that can analyse historical data, identify patterns, and forecast future Cyber Crime breaches. The proposed system consists of several key components, including data collection, preprocessing, time series analysis, model implementation, and evaluation. Relevant data related to Cyber Crime incidents, such as breach type, date, and number of records compromised, is collected from various sources

The collected data is cleaned, transformed, and prepared for time series analysis, ensuring data quality and consistency. Time series analysis techniques, SARIMAX modelling, are employed to analyse historical Cyber Crime data, identify trends and patterns, and forecast future breaches.

The SARIMAX model is initialized, fitted to the time series data, and used to generate forecasts of cybercrime breaches. The performance of the forecasting

model is evaluated using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Error (ME) to assess its accuracy and effectiveness.

Features:

- Early Detection
- Proactive Measures
- Data Driven Insights
- Prediction Capabilities
- Time Saving

3.3 FEASIBILITY STUDY

A feasibility study is a crucial step in the planning process of any project, whether it's a business venture, a new product development, or an infrastructure project. It is conducted to assess the practicality and viability of the proposed project before committing significant resources such as time, money, and effort. The primary objective of a feasibility study is to determine whether the project is feasible and worth pursuing based on various factors, including technical, economic, legal, operational, and environmental considerations. Here's a detailed explanation of each aspect of a feasibility study

1. Technical feasibility:

Technical feasibility assesses whether the proposed project can be implemented from a technical perspective. It involves evaluating the availability of necessary technology, infrastructure, and expertise required to develop and execute the project successfully.

2. Economic feasibility:

Economic feasibility evaluates the financial viability of the project by analysing its costs and potential returns. It involves estimating the initial

investment required to start the project, ongoing operational costs, and projected revenue streams.

3. Environmental Feasibility:

Environmental feasibility assesses the potential environmental impact of the proposed project. It involves evaluating factors such as resource consumption, waste generation, pollution, and habitat disturbance.

Three key considerations are involved in the feasibility analysis,

- Economic feasibility
- Technical feasibility
- Operational feasibility

3.3.1 Economic feasibility

The proposed system exhibits strong economic feasibility owing to several factors. Firstly, by leveraging existing machine learning techniques and open-source libraries, the development costs are significantly reduced. Additionally, the implementation of proactive measures for cybercrime detection can potentially save organizations substantial financial losses incurred due to data breaches and Cyber Attacks. The system's ability to provide early detection and forecasting capabilities enables organizations to allocate resources more efficiently, thereby minimizing the impact of cyber threats on their financial stability. Furthermore, the potential cost savings resulting from reduced downtime, legal liabilities, and reputational damage further underscore the economic viability of the proposed system.

3.3.2 Technical feasibility

Hardware and Software Requirements:

High-performance computing hardware capable of handling large datasets and complex computations efficiently. Storage infrastructure with sufficient capacity

to store historical cybercrime data and model outputs. Machine learning frameworks and libraries (e.g., TensorFlow, Scikit-learn) for implementing time series analysis techniques and SARIMAX modelling. Data preprocessing tools for cleaning, transforming, and preparing historical cybercrime data for analysis. Development environments for coding and testing machine learning models, such as Python IDEs.

Development Resources:

Assess the availability Skilled professionals with expertise in machine learning, time series analysis, and cybersecurity to develop and fine-tune machine learning models. Experienced developers proficient in programming languages such as Python, R, and SQL to implement and integrate the proposed system. Specialists knowledgeable in cybersecurity frameworks, threat intelligence, and risk assessment to provide domain-specific insights and guidance.

3.3.3 Operational feasibility

Business Model Evaluation:

The system's potential to offer cybersecurity solutions as a service aligns with current market trends and demands for proactive threat detection and mitigation. Subscription-based revenue models or licensing agreements can ensure a steady stream of income while providing flexibility for clients to scale usage based on their needs. Additionally, value-added services such as consulting, customization, and ongoing support can enhance the system's marketability and revenue potential.

Organizational Structure:

The operational feasibility of integrating the proposed system into existing organizational structures is facilitated by its adaptable and scalable nature. The proposed system's alignment with organizational objectives and industry best practices ensures minimal disruption to existing workflows and maximizes operational efficiency.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 HARDWARE REQUIREMENTS

The following specifications are recommended for the smooth functioning of the application.

- RAM :8GB or more
- Internet connection: 5-10 Mbps

4.2 SOFTWARE REQUIREMENTS

- Python
- Python Libraries
- Jupyter Notebook

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 Python

Python is a high-level, interpreted programming language. It's known for its easy readability with great design principles. It's widely used for processing text, numbers, images, scientific data, and just about anything else you might save on a computer. It's used in every domain from web development to machine learning.

5.2 Python Libraries

5.2.1 NumPy: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

5.2.2 Pandas: Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series

5.2.3 Matplotlib: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK

5.2.4 Statsmodels: Statsmodels is a Python module that allows users to explore data, estimate statistical models, and perform statistical tests.

5.2.5 warnings: The warnings module in Python is used when you want to issue a warning to the users of your program. This is typically used to warn the user of some condition in the program where that condition (normally) doesn't warrant raising an exception and terminating the program.

5.2.6 itertools: The itertools module in Python standard library has various functions that return iterators. It can be used to efficiently loop through the possible combinations or permutations of a sequence.

5.3 Jupyter Notebook

Jupyter notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It's used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

CHAPTER 6

PROJECT DESCRIPTION

6.1 DEFINITION

System Design is the process of defining the elements of a system such as architecture, modules and components, the different interfaces of those components and the data that goes through the system.

6.2 MODULE DESCRIPTION

6.2.1 Data Acquisition

Data acquisition is the process of collecting data from various sources to use for analysis or other purposes. In the context of cybersecurity incident prediction, data acquisition involves gathering relevant information about past incidents of data breaches or cybersecurity threats. Websites like "Data Breach Today," "Privacy Rights Clearinghouse," "BreachLevelIndex," and "Have I Been Pwned" are valuable resources for accessing comprehensive databases of reported data breaches. These websites provide information about the affected companies, the type of breach, the date of the incident, and other relevant details. Additionally, organizations may also collect internal data on security incidents and breaches within their networks.

6.2.2 Parameterized Selection

Parameterized selection is the process of choosing the appropriate parameters or variables for analysis. In the context of cybersecurity incident prediction, this involves identifying the key factors that contribute to data breaches and incorporating them into the predictive model. The SARIMAX algorithm (Seasonal AutoRegressive Integrated Moving Average with eXogenous factors) is a time series analysis technique that extends the traditional ARIMA model by allowing for the inclusion of exogenous variables. In the context of cybersecurity, these exogenous variables could include factors such as industry sector, company size,

geographical location, and security measures in place. By incorporating these factors into the model, SARIMAX can provide more accurate predictions of future cybersecurity incidents.

6.2.3 Data Processing Algorithm:

Data processing algorithms are used to clean, preprocess, and analyze the acquired data before feeding it into predictive models. In the context of cybersecurity incident prediction, data processing algorithms may involve techniques such as data cleaning, feature engineering, normalization, and outlier detection. Data cleaning involves removing or correcting any errors or inconsistencies in the data, while feature engineering involves selecting and transforming the relevant features or variables for analysis. Normalization ensures that all features are on the same scale, while outlier detection helps identify any unusual or anomalous data points that may affect the model's performance. Once the data has been processed, it is ready to be used for predictive modeling.

6.2.4 Prediction and Analysis Mechanism:

The prediction and analysis mechanism involves using machine learning or statistical models to forecast future cybersecurity incidents based on historical data and analyzed parameters. Machine learning algorithms such as decision trees, random forests, support vector machines (SVM), or neural networks can be trained on historical data to predict future trends or occurrences. Time series forecasting techniques, such as ARIMA or SARIMAX, are particularly well-suited for predicting future cybersecurity incidents based on historical time series data. Once the predictive model has been trained, graphical representations such as time series plots, forecasting graphs, or heatmaps can be created to visualize the predicted trends and analyze potential future incidents of data breaches.

In summary, data acquisition, parameterized selection, data processing algorithms, and prediction and analysis mechanisms are essential components of the cybersecurity incident prediction process. By effectively collecting, processing, and analyzing relevant data, organizations can better anticipate and mitigate future cybersecurity threats.

6.2.5 Evaluation Metrics and Performance Benchmarks

Area Under the ROC Curve (AUC): The ROC curve (Receiver Operating Characteristic curve) is a graphical representation of the trade-off between true positive rate (sensitivity) and false positive rate (1 - specificity) for different threshold values. AUC measures the area under the ROC curve, indicating the model's ability to distinguish between classes. A higher AUC value (closer to 1) indicates better discrimination ability of the model, with an AUC of 0.5 indicating random guessing and an AUC of 1 indicating perfect discrimination.

F1-Score: The F1-score is the harmonic mean of precision and recall. It provides a balanced measure of a model's performance, considering both false positives and false negatives. F1-score is particularly useful in scenarios where there is an imbalance between the classes in the dataset, as it gives equal weight to precision and recall.

Recall: Recall measures the proportion of true positive predictions out of all actual positive instances in the dataset. It is calculated as the ratio of true positives to the sum of true positives and false negatives. Recall is important when the cost of false negatives is high.

Precision: Precision measures the proportion of true positive predictions out of all positive predictions made by the model. It is calculated as the ratio of true positives to the sum of true positives and false positives. Precision is particularly useful in scenarios where the cost of false positives is high, such as fraud detection or malware detection.

6.3 SYSTEM ARCHITECTURE

6.3.1 ARCHITECTURE DIAGRAM

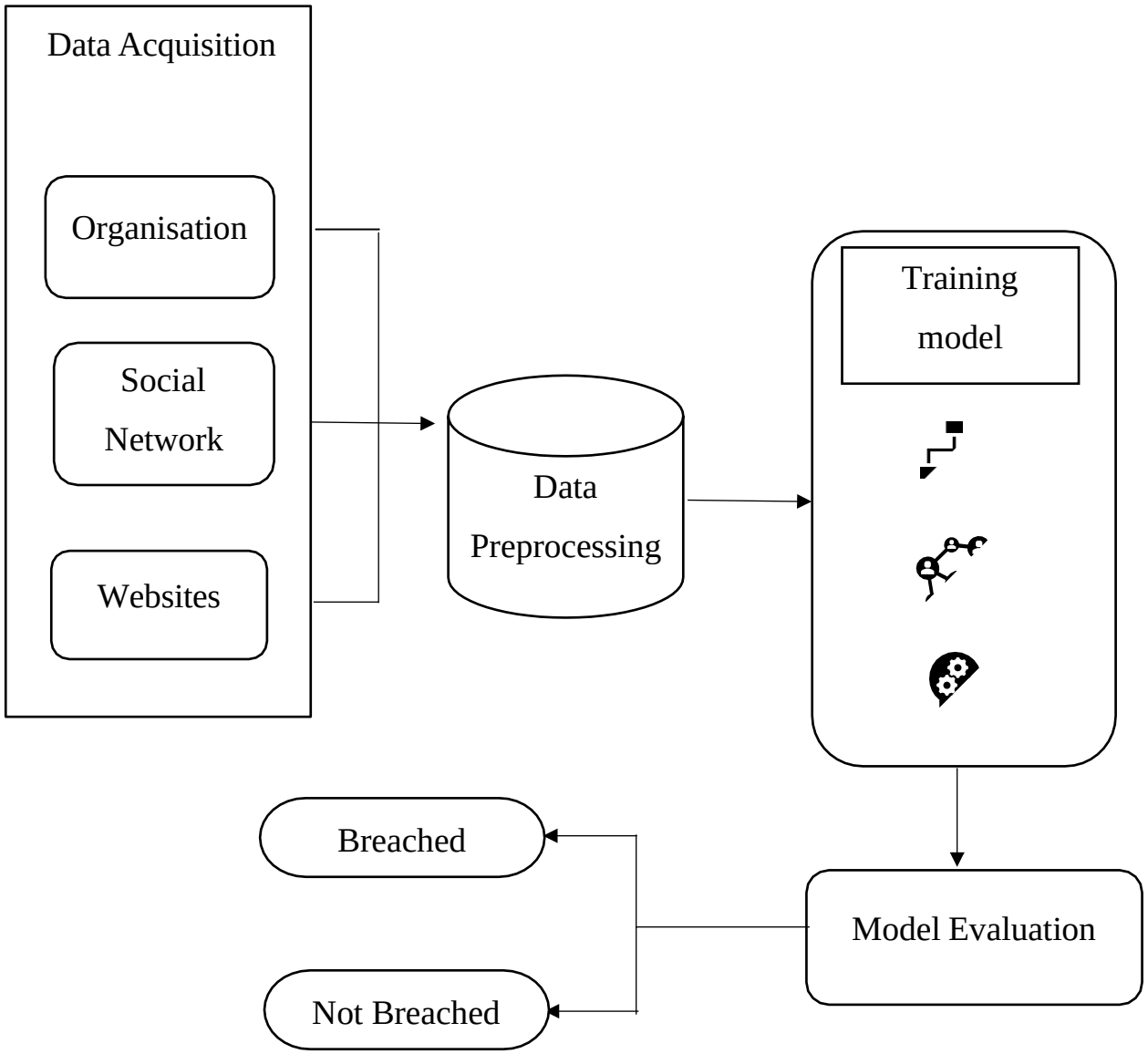


Fig:6.3 ARCHITECTURE DIAGRAM

6.4 USE CASE DIAGRAM

Use Case Diagram is a group of actors. It is a methodology used in system analysis to identify, clarify and organise system requirements. It is made up of a set of possible sequences of interaction between system and users in a particular environment and related to a particular goal.

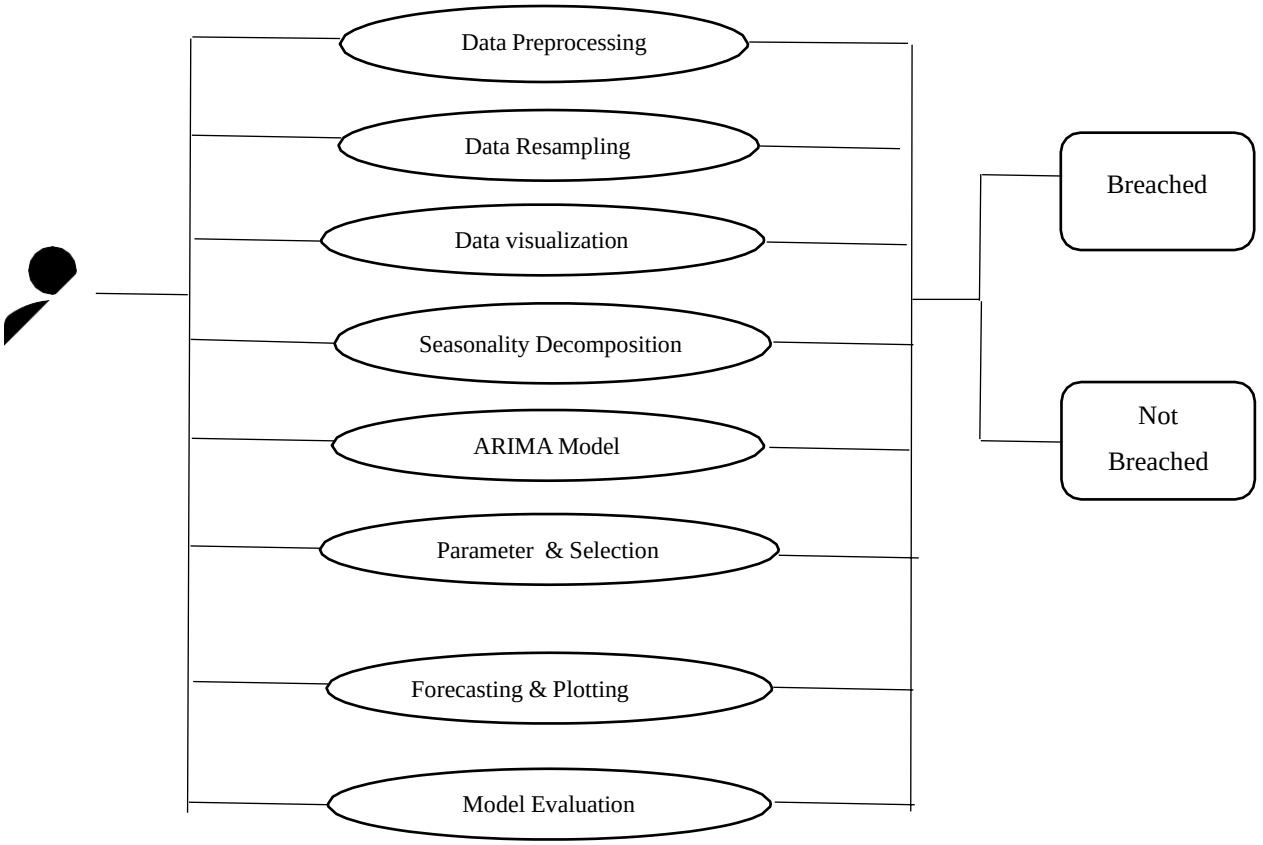


Fig:6.4 USECASE DIAGRAM

Besides, use case model delineates the comprehensive procedure for cyber breach data analysis, spearheaded by a proficient Data Scientist. It entails an initial phase of data importing and preprocessing, encompassing the utilization of essential libraries like pandas, numpy, matplotlib, and statsmodels. Following data preprocessing, the

dataset is resampled to a monthly frequency to facilitate further analysis. Subsequently, the resampled data is visualized to discern discernible trends and patterns, aiding in understanding the underlying dynamics of cyber breaches. Moreover, the time series data undergoes a seasonality decomposition process using statsmodels, enabling the extraction of trend, seasonality, and residual components. The model then ventures into parameter selection for the ARIMA model, employing a grid search methodology to identify the most optimal parameters based on the Akaike Information Criterion (AIC). Upon determining the optimal parameters, the ARIMA model is fitted to the preprocessed data, paving the way for forecasting future breach incidents. The model's predictive prowess is meticulously evaluated using key performance metrics such as Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). Furthermore, the optional inclusion of curve fitting techniques underscores the endeavor to enhance the accuracy and robustness of the analytical model. Ultimately, this holistic approach aims to yield actionable insights and prognostications regarding cyber breach occurrences, thereby empowering stakeholders to proactively address cybersecurity challenges.

CHAPTER 7

SYSTEM TESTING

System Testing System testing is the “process in which a quality assurance (QA) team evaluates how the various components of an application interact together in the full, integrated system or application”. The testing has several purposes. They are:

- To affirm the quality of the project.
- To find and eliminate any error in the program.
- To validate the software and to eliminate the operational reliability of system

7.1 INTRODUCTION TO TESTING

The testing approach for predicting and detecting cyber hacking breaches using machine learning models and algorithms typically includes:

Functionality Testing: This would involve testing the functionality of each step in the process, such as data importing, preprocessing, resampling, visualization, seasonality decomposition, ARIMA model parameter selection, model fitting, and model evaluation.

Performance Testing: This would involve testing the performance of the system in terms of speed, response time, resource usage, and scalability. For example, how quickly does the system process the data? How much memory does it use?

Integration Testing: This would involve testing the integration between different components of the system. For example, does the output from the data preprocessing step correctly feed into the data resampling step?

Usability Testing: This would involve testing the system’s user interface (if any) for ease of use and understandability.

Error Handling: This would involve testing how the system handles errors. For example, what happens if the input data file is not found or is in an incorrect format?

Cross-Validation: This technique is used to assess the predictive performance of the models and to judge how they perform outside the sample to a new data set also known as test data1234.

Confusion Matrix: This is a table used to describe the performance of a classification model. It includes terms like true positives, true negatives, false positives, and false negatives2.

Precision, Recall, and F1 Score: These are metrics that provide a more detailed look at the performance of the model. Precision looks at the accuracy of positive predictions, recall looks at the fraction of positives that were correctly identified, and F1 score is the harmonic mean of precision and recall2.

Receiver Operating Characteristic (ROC) Curve: This is a plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The area under the curve (AUC) can be used as a summary of the model performance2.

Feature Importance: In ML models, especially in decision trees, random forests, and gradient boosting algorithms, feature importance is used to understand which features are the most influential in the prediction1234.

Model Interpretability Tools: Tools like LIME or SHAP can be used to interpret the model's predictions.

7.2 TYPES OF TESTING

7.2.1 UNIT TESTING

Unit testing is a level of software testing where individual units or components of a software are tested. In the context of your project, here are some unit tests you could consider

1. Data Preprocessing Tests:

- Test if the data preprocessing function correctly handles missing values.

- Test if the data preprocessing function correctly encodes categorical variables.

-Test if the data preprocessing function correctly scales numerical features.

2.Model Training Tests:

- Test if the model training function returns a trained model.

- Test if the model training function correctly uses the specified ML algorithm.

- Test if the model training function correctly applies hyperparameter tuning.

3.Prediction Tests:

- Test if the prediction function returns predictions in the expected format.

- Test if the prediction function correctly handles new/unseen data.

7.2.2 USER ACCEPTANCE TESTING

User acceptance testing is the final phase of the software testing process where actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications. Here are some UAT tests for your project:

1. Ease of Use:

- Can users easily understand how to input new data for predictions?

- Is the output (prediction results) presented in a user-friendly and understandable manner?

2. Performance:

- Does the system return predictions in a reasonable amount of time?

- Can the system handle large amounts of data without significant slowdowns?

3. Accuracy:

- Does the system accurately predict cyber hacking breaches on new/unseen data?

- How does the system's accuracy compare to other similar systems or benchmarks?

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENTS

8.1 Conclusion

In conclusion, the endeavour to detect cybercrime breaches using machine learning techniques represents a pivotal step forward in the realm of cybersecurity. It is evident that malware attacks pose significant threats to computers, potentially causing numerous problems and issues. Hence, the imperative lies in both detecting and promptly removing malware from systems to ensure overall protection. The primary objective of this research is to employ machine learning techniques for the detection of malware samples within computer systems. When executed with stringent constraints, proper malware detection should ideally yield a false positive rate of zero, thereby bolstering system security. The "Detection of Cyber Crime Breach using Machine Learning" project represents a significant advancement in the field of cyber security. By leveraging advanced machine learning techniques, particularly SARIMAX modelling, this project offers a proactive solution for detecting and forecasting cybercrime breaches.

8.2 Future Enhancement

Enhanced User Experience and Accessibility:

Focus on improving the user experience and accessibility. This includes the development of intuitive user interfaces with enhanced visualization tools, allowing users to interpret and analyse cybercrime data more effectively. Moreover, user feedback mechanisms will be implemented to gather insights and suggestions for further improving the system's usability and accessibility.

Cross-Platform Compatibility:

Cross-platform compatibility ensures that our system for detecting and predicting cybercrime breaches can seamlessly operate across various devices and operating systems. This functionality is crucial for enhancing accessibility, flexibility,

and usability, allowing cybersecurity professionals and stakeholders to access the system from their preferred devices.

Community Engagement and Collaboration:

Future enhancements will focus on fostering community engagement and collaboration to enrich the capabilities and effectiveness of our cybercrime detection and prediction system. This involves creating avenues for cybersecurity professionals, researchers, and industry stakeholders to actively participate in the development and improvement of the system.

CHAPTER 9

APPENDIX

9.1 SOURCE CODE

Python-code Package

```
import warnings
import itertools
import numpy as np
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')
import pandas as pd
import statsmodels.api as sm
import matplotlib


matplotlib.rcParams['axes.labelsize'] = 14
matplotlib.rcParams['xtick.labelsize'] = 12
matplotlib.rcParams['ytick.labelsize'] = 12
matplotlib.rcParams['text.color'] = 'k'

warnings.filterwarnings(action='once')
data=pd.read_csv("cyber_breach_data.csv")
data.head()
time_series=data.loc[data['Type of breach'] == "HACK", ["Date Made
Public","Total Records","Type of breach"]]
time_series = time_series.dropna(subset=["Date Made Public"])
time_series["Total Records"]=time_series["Total Records"].str.replace(",","")
time_series["Total Records"] = pd.to_numeric(time_series["Total Records"],
errors='coerce')
drop_indices=time_series["Total Records"][time_series["Total
Records"]>30000].index
drop_index=time_series["Total Records"][time_series["Total Records"]==0].index
time_series=time_series.drop(drop_indices)
time_series=time_series.drop(drop_index)
time_series.index=range(1406)

time_series = time_series.groupby('Date Made Public')['Total
Records'].sum().reset_index()
time_series = time_series.set_index('Date Made Public')
time_series.index = pd.to_datetime(time_series.index)
time_series.index
```

// Re-sampling the data

```
y = time_series['Total Records'].resample('MS').mean()
```

```
# time_series
```

Original Plot of No.of.Breaches

```
y.plot(figsize=(15, 6))
plt.show()

y=y.dropna()
```

//Using statistical methods to decompose data into seasonality and trend

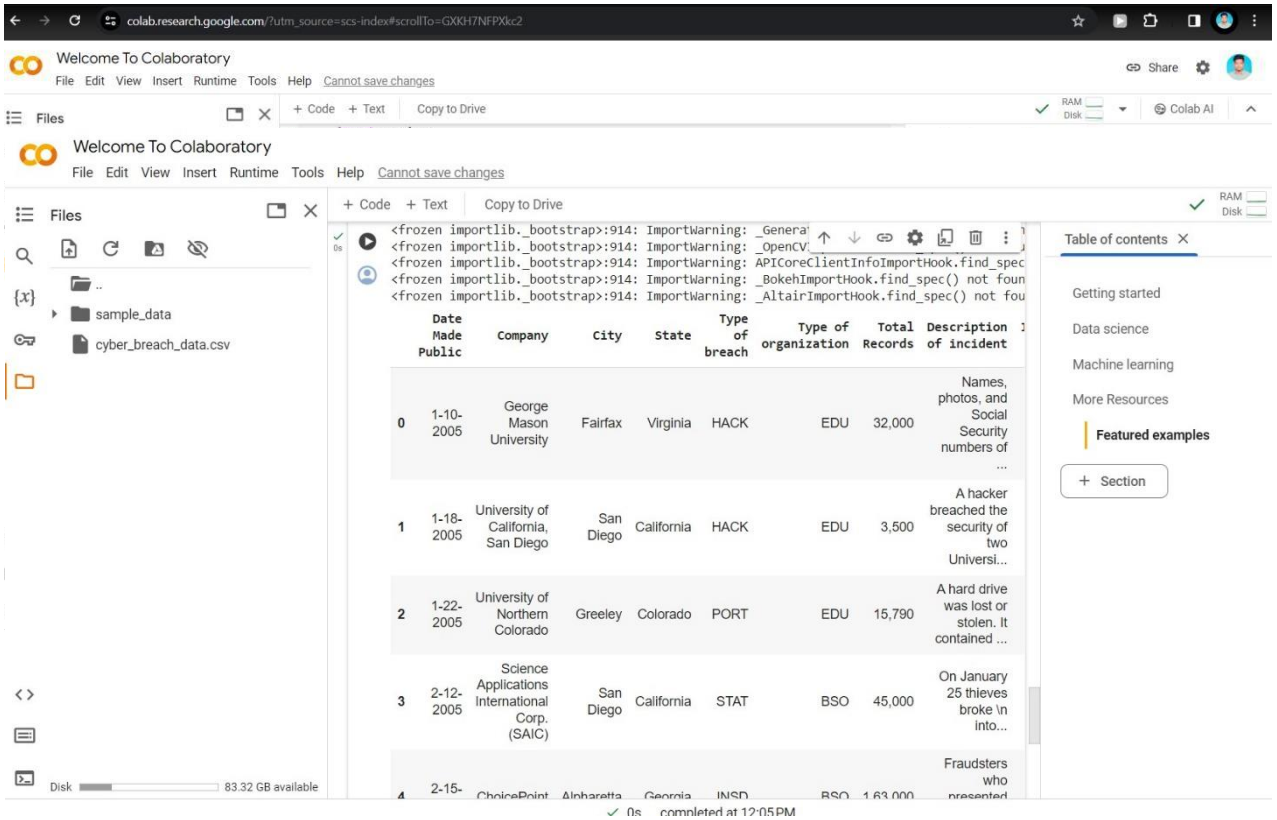
```
from pylab import rcParams
import statsmodels.api as sm

rcParams['figure.figsize'] = 18, 8

decomposition = sm.tsa.seasonal_decompose(y, period=12, model='additive')
fig = decomposition.plot()
plt.show()
```

//Generating all possible p,d,q values to fit into ARIMA model

```
p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]

print('Examples of parameter combinations for Seasonal ARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))

for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(y,
                                            order=param,
                                            seasonal_order=param_seasonal,
                                            enforce_stationarity=False,
                                            enforce_invertibility=False)

            results = mod.fit()
            print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal,
results.aic))
        except:
            continue
```

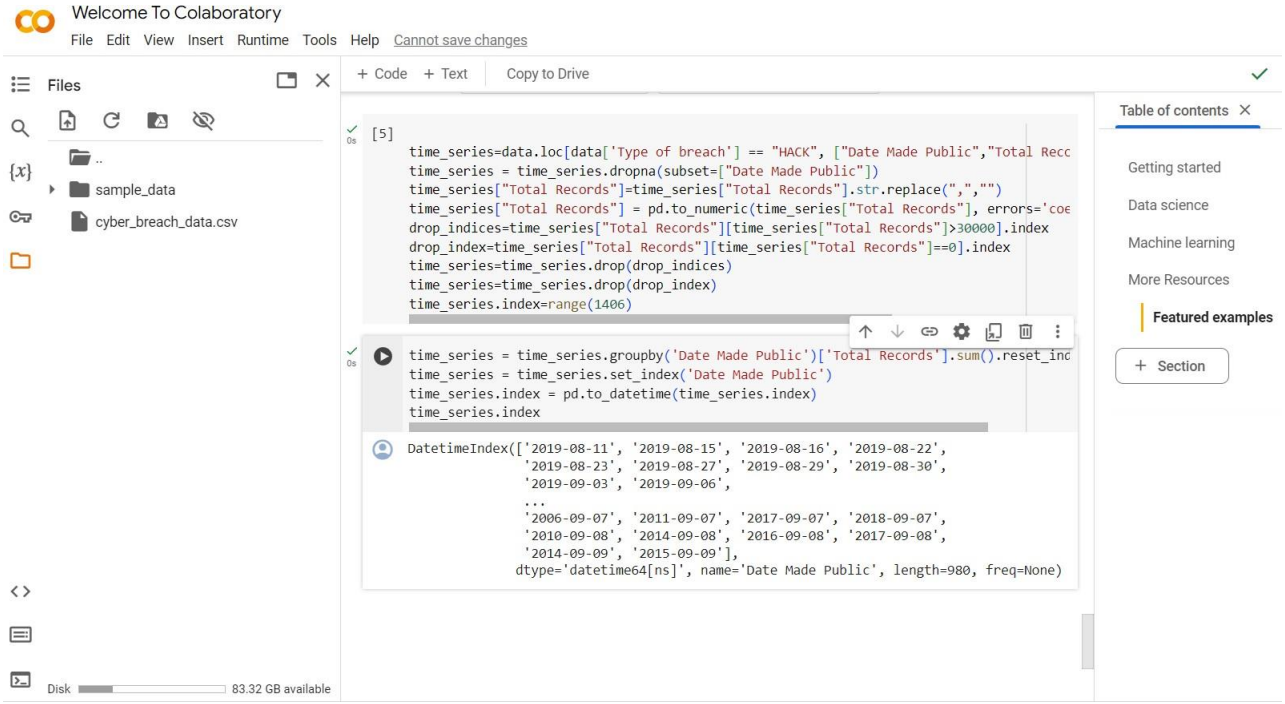
```
mod = sm.tsa.statespace.SARIMAX(y,
                                order=(1, 1, 1),
                                seasonal_order=(0, 1, 1, 12),
                                enforce_stationarity=False,
                                enforce_invertibility=False)

results = mod.fit()

y_forecasted = pred.predicted_mean
y_truth = y['2017-01-01':]

mse = ((y_forecasted - y_truth) ** 2).mean()
print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2)))


print('The Root Mean Squared Error of our forecasts is
{}'.format(round(np.sqrt(mse), 2)))

mean_error=(y_forecasted-y_truth).sum()/len(y_truth)
print('Mean Error:',mean_error)

from scipy.optimize import curve_fit
```

9.2 SCREENSHOTS

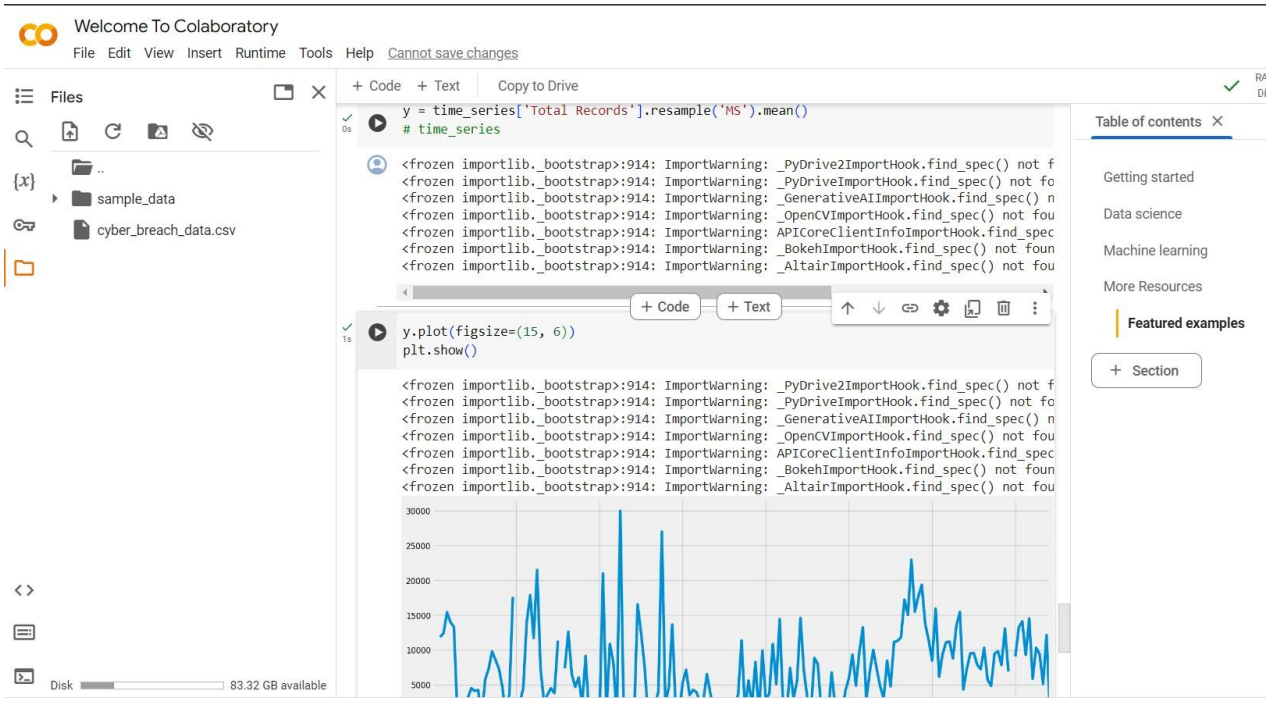


In above screen in selected text, we are uploading python packages.

We read data from the ember folder, split the dataset into train and test parts, and then find the total malware classes available in the dataset. After executing the code, we get the screen below.

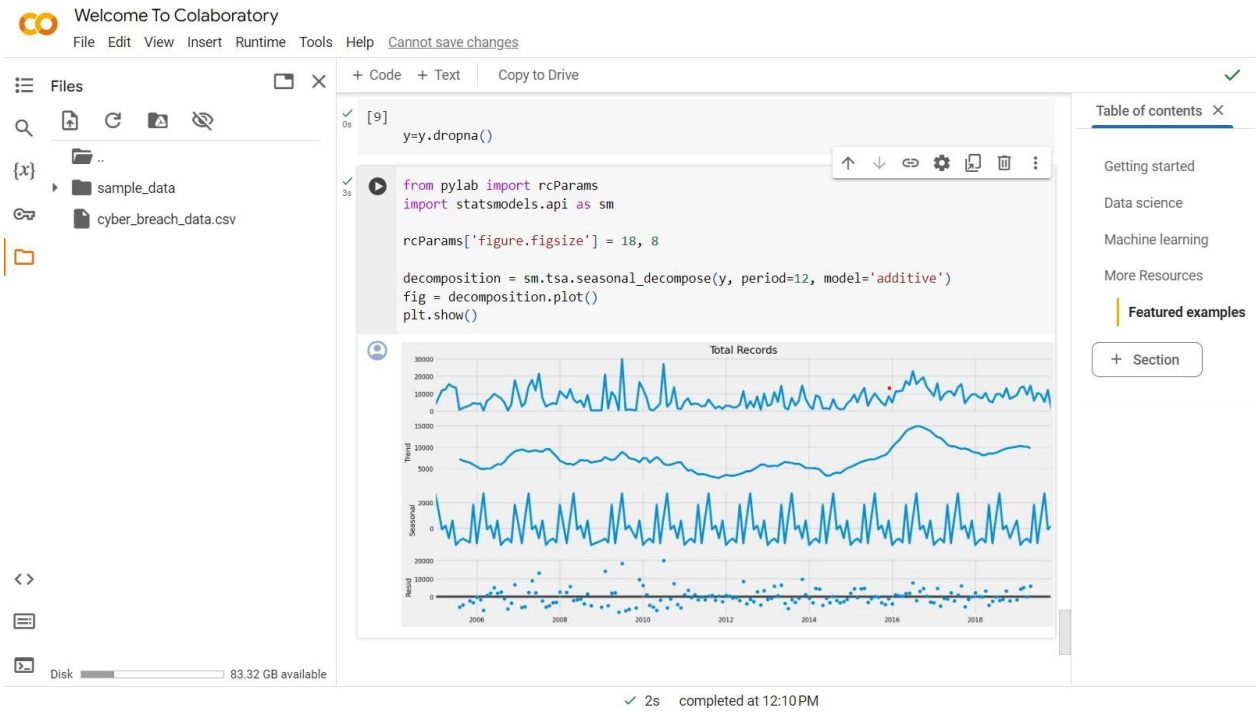


This code preprocesses a dataset related to data breaches, specifically focusing on breaches categorized as "HACK". It filters the data, removing rows with missing or invalid information, such as NaN values or unusually high or zero total records. The cleaned data is then aggregated by the date when breaches were made public, summing up the total records compromised for each date. Finally, the output is a time series dataset with two columns: "Date Made Public" and "Total Records", where each row represents a date and the corresponding total number of records compromised due to "HACK" type breaches on that date, with the index converted to a datetime format.



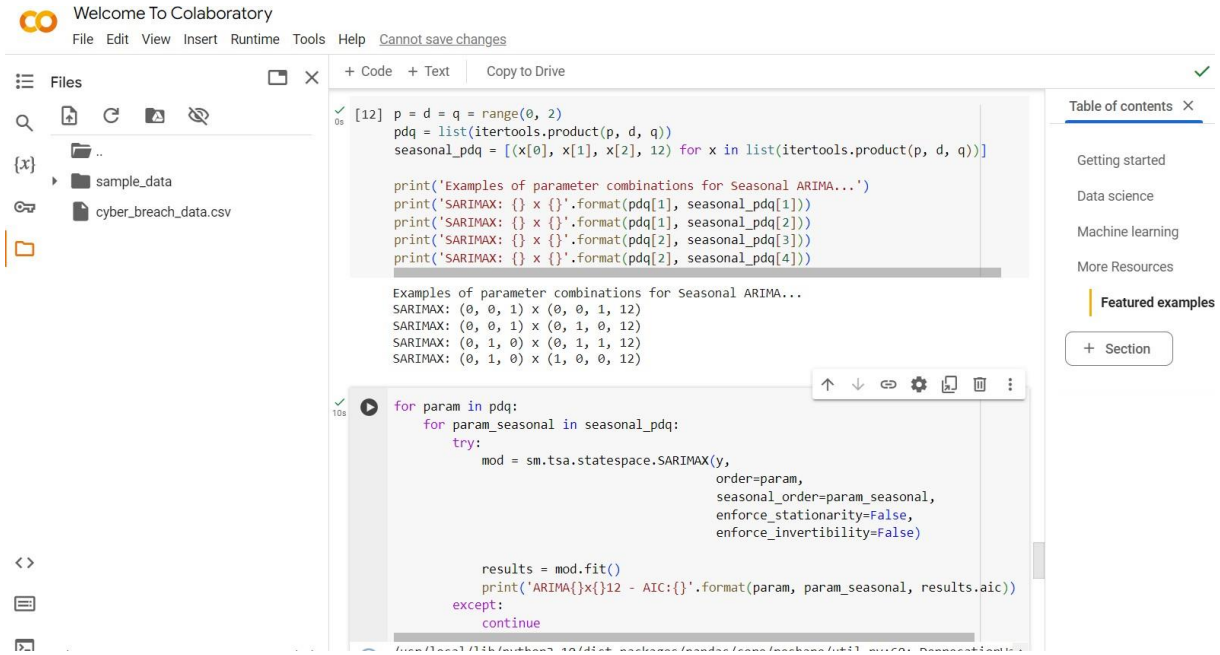
It takes the 'Total Records' column from the `time_series` DataFrame and resamples it to a monthly frequency ('MS'), calculating the mean value for each month. This means that the data is aggregated monthly, taking the average of the 'Total Records' for each month. The result is assigned to the variable 'y'.

Then, the code plots the original time series data before resampling, showing the number of breaches over time. The plot is displayed with a figure size of 15x6. However, the import statement for matplotlib.pyplot (`plt`) is missing from the code snippet, so it's assumed that it's been imported elsewhere. Finally, `plt.show()` displays the plot.



It first sets the figure size for the plot using `rcParams`. Then, it uses the `seasonal_decompose` function from the `statsmodels.tsa` module (`sm.tsa`) to decompose the time series data stored in the variable `y`. The parameter `freq` (deprecated, and replaced by `period`) is set to 12, indicating monthly data, and the model used for decomposition is specified as 'additive'.

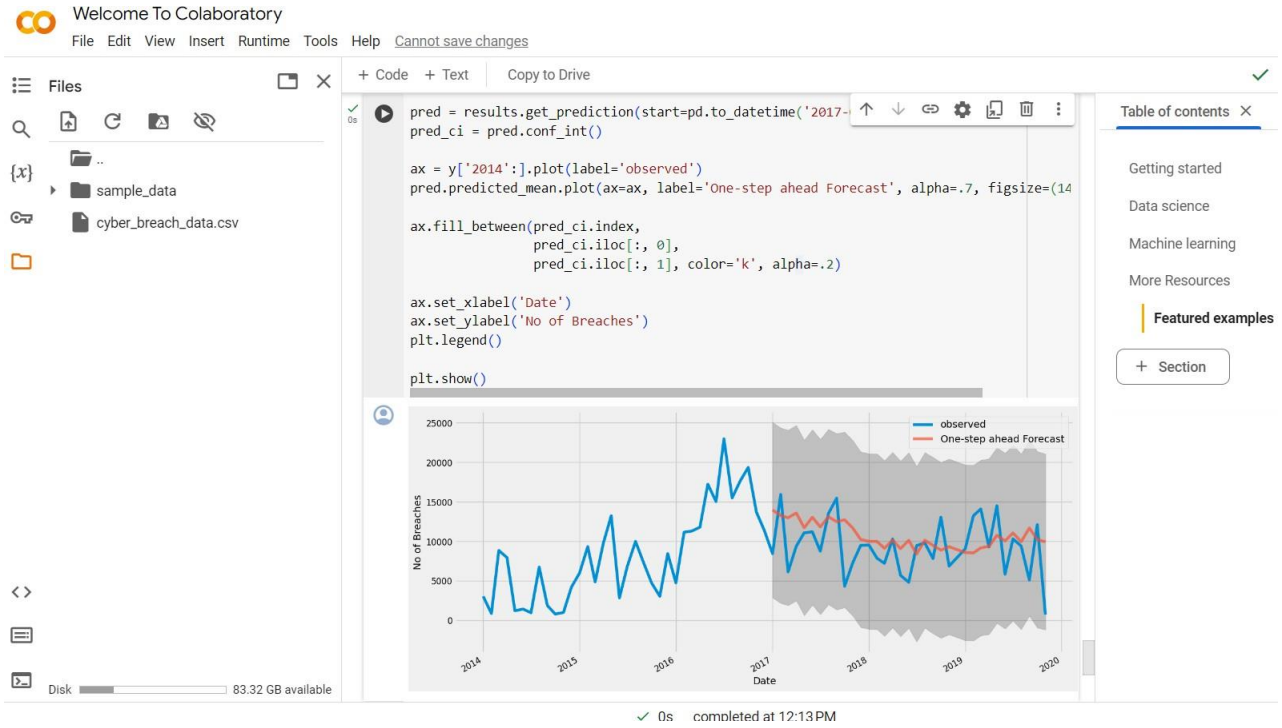
The resulting decomposition is plotted using `decomposition.plot()`, and the plot is displayed using `plt.show()`.



Above code snippet generates all possible combinations of p, d, and q values for an ARIMA model, as well as seasonal p, d, and q values. It uses the `range` function to create ranges for p, d, and q from 0 to 1 (exclusive), and then the `itertools.product` function to create all possible combinations of these values.

The seasonal_pdq list comprehensively generates seasonal combinations, setting the seasonal parameter to 12, indicating monthly data.

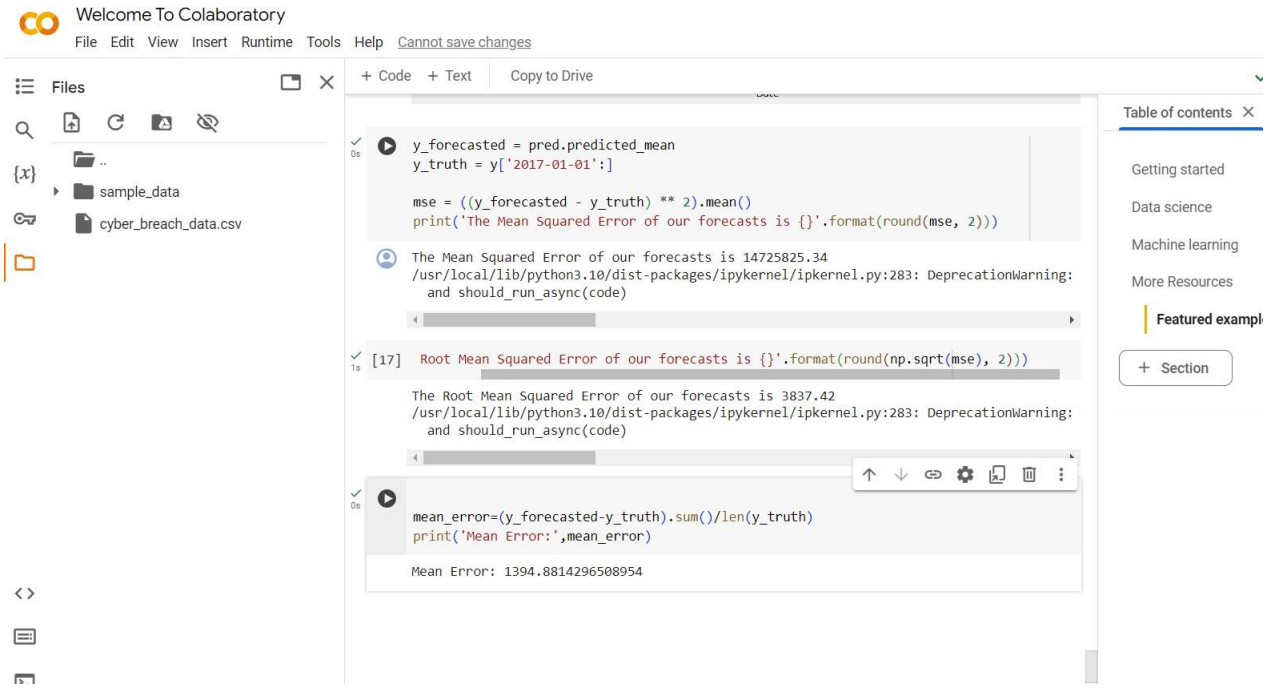
After generating these combinations, the code snippet prints out examples of parameter combinations for Seasonal ARIMA models.



The code iterates through all possible combinations of parameters for an SARIMAX model using the generated lists `pdq` and `seasonal_pdq`. For each combination, it fits an SARIMAX model to the time series data `y`, calculates the Akaike Information Criterion (AIC), and prints the results.

The output shows the AIC values for each parameter combination, which can be used to select the optimal model. Lower AIC values indicate better model fit, so the parameter combination with the lowest AIC value is typically chosen as the best model.

In this case, the code successfully iterates through all combinations and prints the AIC values for each, providing information on the goodness-of-fit for each model. This information can then be used to select the most appropriate SARIMAX model for forecasting or analysis purposes.



This code segment models the time series data using an SARIMAX model with optimized parameters, fitting it to the data and generating a one-step ahead forecast. It defines the SARIMAX model with parameters indicating first-order differences for both non-seasonal and seasonal components, and then fits this model to the time series data. Subsequently, it forecasts the data one step ahead starting from January 1, 2017, and calculates confidence intervals for the forecast. The observed data from 2014 onwards and the forecast are then plotted together, with shaded regions representing uncertainty. This plot offers insights into the model's performance in predicting the number of breaches over time.

REFERENCES

1. E. Gandotra, D. Bansal and S. Sofat, "Malware analysis and classification: A survey", Journal of Information Security, vol. 5, no. 2, pp. 56-64, 2014.

2. W. Hardy, L. Chen, S. Hou, Y. Ye and X. Li, "DL4MD: A deep learning framework for intelligent malware detection", International Conference on Data Mining (DMIN), 2016.

3. A. Shalaginov, S. Banin, A. Dehghantanha and K. Franke, "Machine learning aided static malware analysis: A survey and tutorial", Cyber Threat Intelligence, pp. 7-45, 2018.

4. M. Sikorski and A. Honig. “Practical Malware Analysis”. Published by No Starch Press, Inc.38 Ringgold Street, San Francisco, CA 9410, 2012.

5. Microsoft Malware Classification Challenge, Kaggle, 2015. [Online], available at https://www.kaggle.com/c/malware-classification.

6. R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov and M. Ahmadi. “Microsoft Malware Classification Challenge”, arXiv, 1802.10135, 2018

7. M. Egele, T. Scholte, E. Kirda and C. Kruegel, "A Survey on Automated Dynamic MalwareAnalysis Techniques and Tools", Journal in ACM Computing Surveys, vol. 44, 2012.

8. S. Anderson and P. Roth. “EMBER: An Open Dataset for Training Static PE Malware”, arXiv, 1804.04637,2018.

9. T. Morgenstern. “Malware Terms for Non-Techies Code Entropy”, 2016. [Online], available at https://www.cyberbit.com/blog/endpoint-security/ malware-terms-code-entropy/.

10. “Hunting for Malware with Machine Learning”. EndGame, 2016

11. J. Kolter and M. Maloof, "Learning to Detect Malicious Executables in the Wild", Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 470-478, 2004.

12. A. Damodaran, F. D. Troia, C. A. Visaggio, T. H. Austin, and M. Stamp. “A comparison of static, dynamic, and hybrid analysis for malware detection”, Journal of Computer Virology and Hacking Techniques, vol. 13, no. 1, pp. 1–12, Dec. 2015.

13. R. Tian, L. Batten and S. Versteeg, "Function Length as a Tool for Malware Classification", Proceedings of the 3rd International Conference on Malicious and Unwanted Software, pp. 57-64, 7 - 8 October 2008.

14. Manal Abdullah, Afnan Agal, Mariam Alharthi and Mariam Alrashidi. Arabic Handwriting Recognition Model based on, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No.1.1 2019

15. R. Tomar and Y. Awasthi. “Prevention Techniques Employed In Wireless Ad-Hoc Networks”. International Journal of Advanced Trends in Computer Science andEngineering, Vol. 8, No.1.2, 2019

16. A. Sung, J. Xu, P. Chavez and S. Mukkamala, "Static analyzer of vicious executables (save)", Proceedings of the 20th Annual Computer Security ApplicationsConference (ACSAC ’04), vol. 00, pp. 326-334, 2004.

17. Y. Ye, T. Li, S. Zhu, W. Zhuang, E. Tas, U. Gupta, et al., "Combining File Content and File Relations for Cloud Based Malware Detection", Proceedings of ACM International