JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Bitcoin Snatching Ransomware Strike Forecast using Data Science

[1]Ruthraprasath.K[1], [2]Sohail Khan.M[2] , [3]Mrs.G.S.Devilakshmi[3]
Department of Information Technology, Meenakshi Engineering College , Tamil Nadu
(Affiliated to Anna University Tamil Nadu)

Abstract :

Ransomware attacks are emerging as a major source of malware intrusion in recent times. While so far ransomware has affected general-purpose adequately resourceful computing systems. Many ransomware prediction techniques are proposed but there is a need for more suitable ransomware prediction techniques for machine learning techniques.

This paper presents an attack of ransomware prediction technique that uses for extracting information features in Artificial Intelligence and Machine Learning algorithms for predicting ransomware attacks.

The application of the data science process is applied for getting a better model for predicting the outcome. Variable identification and data understanding is the main process of building a successful model. Different machine learning algorithms are applied to the pre-processed data and the accuracy is compared to see which algorithm performed better other performance metrics like precision, recall, f1-score are also taken in consideration for evaluating the model. The machine learning model is used to predict the ransomware attack outcome.

I. INTRODUCTION

Internet usage was scarce at first. It was first primarily utilised by the industrial, military, and research sectors, but with time, its use spread across society. A vast number of consumers are drawn to technology because of its rapid advancement and accessibility, which helps the Internet grow. Technology advanced to the point where private information was stored on devices with Internet connection and became a daily necessity for communication and storage.

This made a small subset of Internet users interested in stealing such material for financial gain. As a result, malicious software that targets Internet-connected gadgets started to appear . Ransomware is one of the many distinct kinds of harmful software that can be discovered on the Internet; it is also one of the most destructive and has recently become very popular among cybercriminals. This kind of virus has been used in attack campaigns against several organisations, both public and private.Digital currencies called cryptocurrencies, like Bitcoin, are created to operate independently of the established financial system. Blockchain technology is used by cryptocurrencies to record transactions, making them decentralised money.

A crypto-exchange platform is primarily used to manage cryptocurrency transactions, often known as the buying and selling of digital currency. Cybercriminals are drawn to these transactions because they frequently involve sizable amounts of cryptocurrencies and are typically anonymised using the blockchain. Platforms and exchange methods for cryptocurrencies are susceptible to cyberattacks, just like any other system.

The data set from the Bitcoin Heist is used to categorise the various fraudulent transactions. To identify a classifier label among those that have been classified as ransomware or connected to harmful activity, the various transactional aspects are studied. I create a random forest classifier using ensemble learning and decision tree classifiers. Findings are evaluated for memory, accuracy, and precision.

II. EXISTING SYSTEM

One of the most troublesome forms of cybercrime is ransomware, which affects productivity, accessibility, and reputation in addition to costing a lot of money. Despite having encryption or locking as one of its final goals, ransomware is frequently built to avoid detection by making a sequence of pre-attack API requests, or "paranoia" activities, to find a suitable execution environment. In this paper, we describe a groundbreaking attempt to use such paranoia actions for identifying distinguishing ransomware tendencies.

To accomplish this, we use more than 3K samples from recent/notable ransomware families to identify the specific paranoia-inspiring activities that each sample represents. In this paper, we provide a dynamic analytic method for identifying ransomware samples based on their paranoid pre-attack behaviours. In order to gather information about the behavioural traits/features of more than 3,000 ransomware samples from five major families, we execute them in a sandboxing environment and call 23 pre-attack evasion APIs that are connected to sensing the execution environment

III. PROPOSED SYSTEM

The goal of the system under consideration is to create a ransomware attack prediction model. The first step in the procedure is the identification of variables, such as dependent and independent variables, where we locate the target column. To deal with missing values, pre- processing procedures are then used.

The pre-processed data is then utilised to create a model by splitting the dataset into 7:3 ratios, with 70% of the data being used for training purposes so that the model can learn the pattern and the remaining 30% being used for testing purposes so that our project can be evaluated for performance. The classification model can be employed to forecast the various ransomware attack types that target bitcoin and we are implementing particularly on bitcoin ransomware attacks and the voting classifier, with deployment potential.

IV. SYSTEM SPECIFICATION

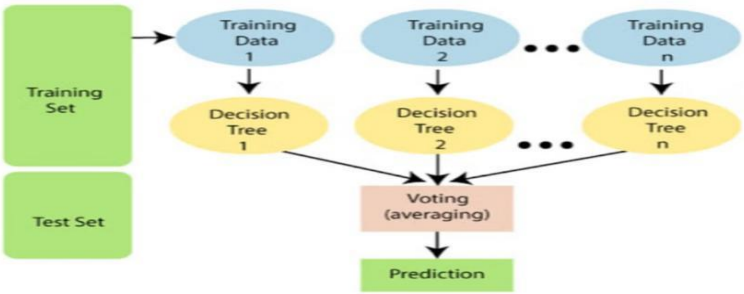
HARDWARE CONFIGURATION:

Processor	-	I5
Speed	-	3 GHz
RAM	-	8 GB(min)
Hard Disk	-	500 GB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	LCD

SOFTWARE CONFIGURATION

Operating System: Linux, Windows/7/10
Server: Anaconda, Jupyter,pycharm
Front End: tkinter |GUI toolkit
Server side Script: Python , AIML

V. RANDOM FOREST AND LINEAR METHOD TECHNIQUES



The following steps explain the working Random Forest Algorithm:

Step 1: Select random samples from a given data or training set.

Step 2: This algorithm will construct a decision tree for every training data.

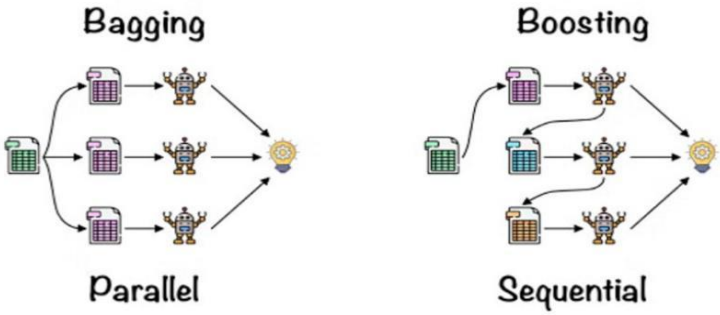
Step 3: Voting will take place by averaging the decision tree.

Step 4: Finally, select the most voted prediction result as the final prediction result.

This combination of multiple models is called Ensemble. Ensemble uses two methods:

Bagging: Creating a different training subset from sample training data with replacement is called Bagging. The final output is based on majority voting.

Boosting: Combing weak learners into strong learners by creating sequential models such that the final model has the highest accuracy is called Boosting. Example: ADA BOOST, XG BOOST.



Bagging: From the principle mentioned above, we can understand Random forest uses the Bagging code. Now, let us understand this concept in detail. Bagging is also known as Bootstrap Aggregation used by random forest. The process begins with any original random data. After arranging, it is organised into samples known as Bootstrap Sample. This process is known as Bootstrapping.Further, the models are trained individually, yielding different results known as Aggregation. In the last step, all the results are combined, and the generated output is based on majority voting. This step is known as Bagging and is done using an Ensemble Classifier.

Key Elements of Random Forest:

Random: Every tree has a distinct quality, range, and characteristic in relation to other trees. Trees differ from one another.The curse of dimensionality does not apply to trees because they are conceptual concepts and do not need characteristics to be taken into account. As a result, there is less feature space.

Parallelization: Since each tree is built independently from distinct data and features, we can produce random forests by utilizing the entire CPU.

Train-Test split: Since the decision tree in a Random Forest never sees 30% of the input, we don't need to separate the data for training and testing.

Stability: The outcome is determined by bagging, which uses the average or majority vote to determine the outcome.

Popular machine learning algorithm Random Forest is a part of the supervised learning methodology. It can be applied to ML issues involving both classification and regression. It is built on the idea of ensemble learning, which is a method of integrating various classifiers to address difficult issues and enhance model performance.

Random Forest, as the name implies, is a classifier that uses a number of decision trees on different subsets of the provided dataset and averages them to increase the dataset's predictive accuracy. Instead than depending on a single decision tree,
the random forest uses forecasts from each tree and predicts the result based on the votes of the majority of predictions.
Another potent supervised machine learning approach utilised for binary classification issues is logistic regression (when target is categorical).

The best approach to conceptualise logistic regression is as a linear regression applied to classification issues. In essence, logistic regression models a binary output variable using the logistic function described below (Tolles & Meurer,2016). The main distinction between logistic regression and linear regression is that the range of logistic regression is constrained to values between 0 and 1. Moreover, logistic regression does not require a linear relationship between the input and output variables, in contrast to linear regression. Logistic regression is another powerful supervised ML algorithm used for binary classifications problems (when target is categorical).

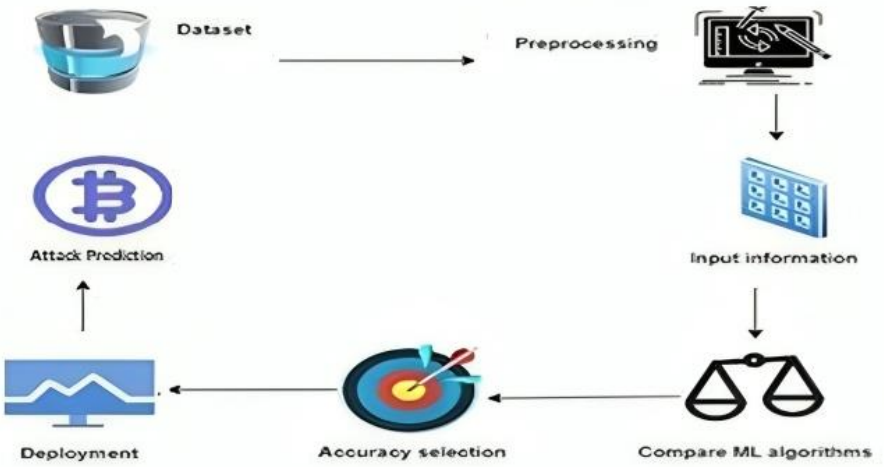
It routinely surpasses all other algorithms designed for supervised learning tasks because to its unmatched speed and performance. The main algorithm can run on clusters of GPUs or even over a network of computers because the library is parallelizable. This enables the high- performance training of ML tasks using hundreds of millions of training instances.

A voting classifier is a machine learning model that learns from a collection of many models and forecasts an output (class) based on the class that has the highest likelihood of being the output to predict the output class based on the highest majority of voting, it merely aggregates the results of each classifier that was passed into the voting classifier. The concept is to build a single model that learns from these models and predicts output based on their aggregate majority of voting for each output class, rather than building separate dedicated models and determining the accuracy for each of them.

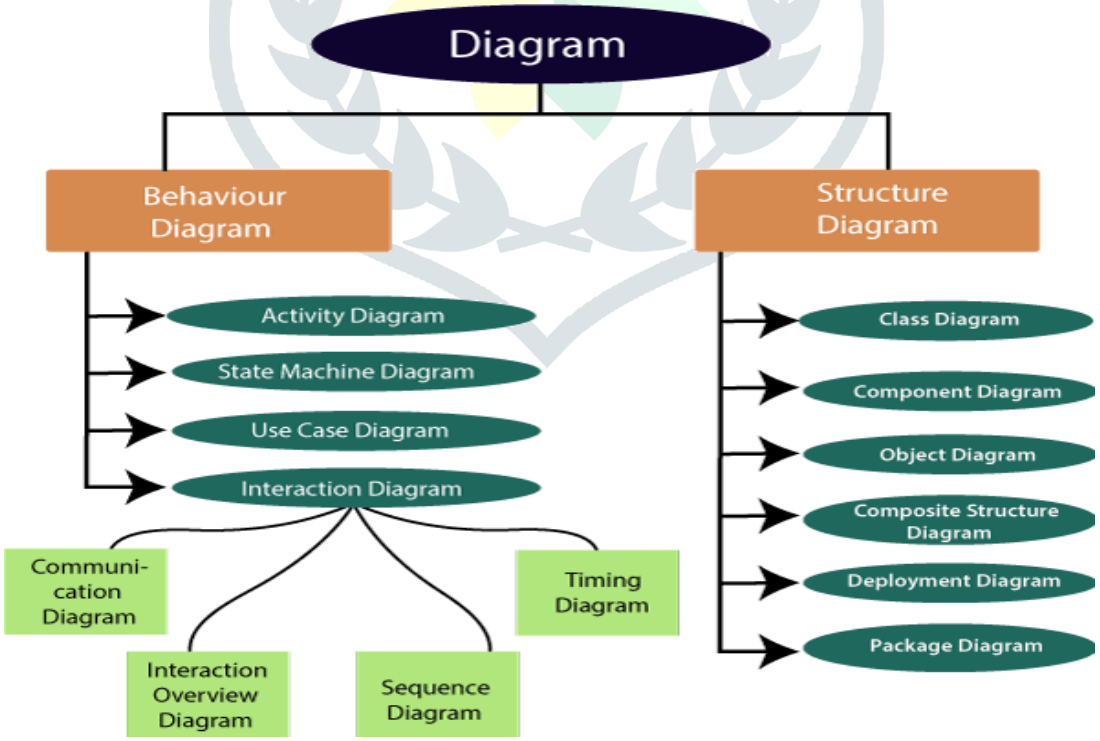
5.1 ADVANTAGE OF PROPOSED ALGORITHM

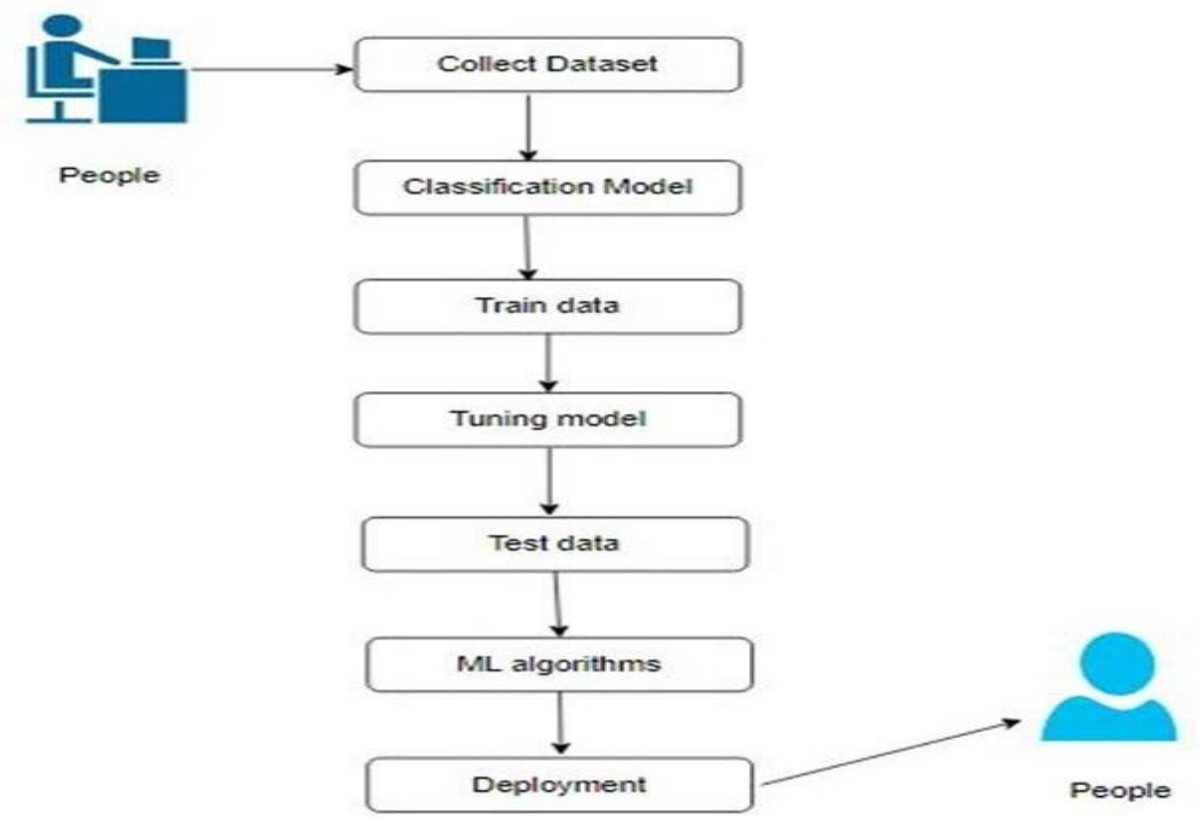
- **Diverse Machine Learning Techniques:** The algorithm leverages a hybrid approach by combining the strengths of Random Forest (RF) and Linear Method (LM).
- **Improved Prediction Techniques:** By integrating different characteristics of RF and LM, the algorithm aims to overcome the limitations of individual methods, providing a synergistic effect that enhances the overall prediction accuracy.
- **Identification of Significant Features:** The algorithm focuses on identifying and utilizing significant features within the raw bitcoin data.
- **Increased Performance in Heart Disease Prediction:** Through the combination of RF and LM characteristics, the Hybrid HRFLM algorithm has demonstrated efficacy in accurately predicting the ransomeware attrack.
- **Potential for Real-world Application:** The algorithm's success in simulations suggests its potential applicability to real-world datasets.

VI. ARCHITECTURE DIAGRAM

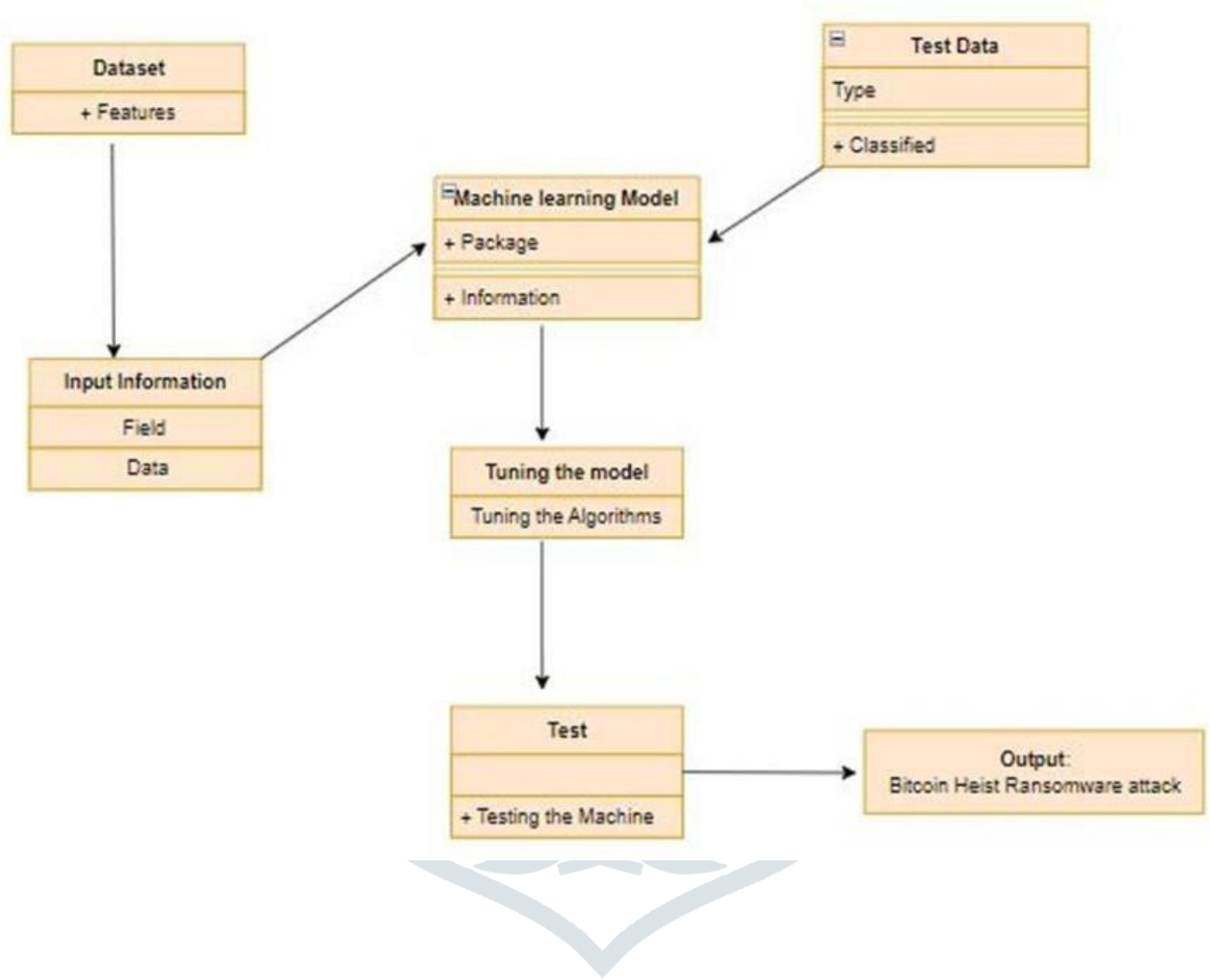


6.1UML DIAGRAM

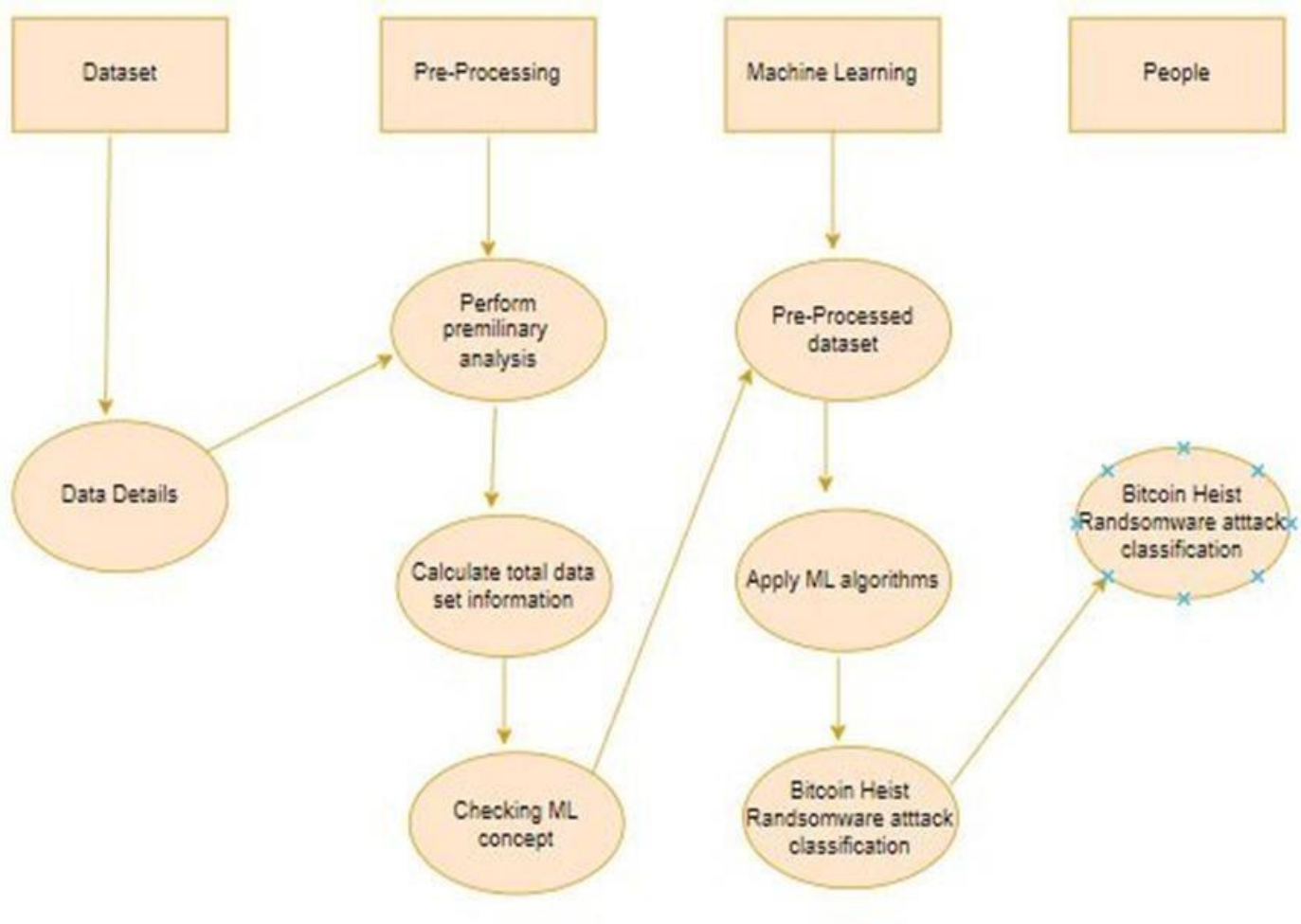



People
Collect Dataset
Classification Model
Train data
Tuning model
Test data
ML algorithms
Deployment
People

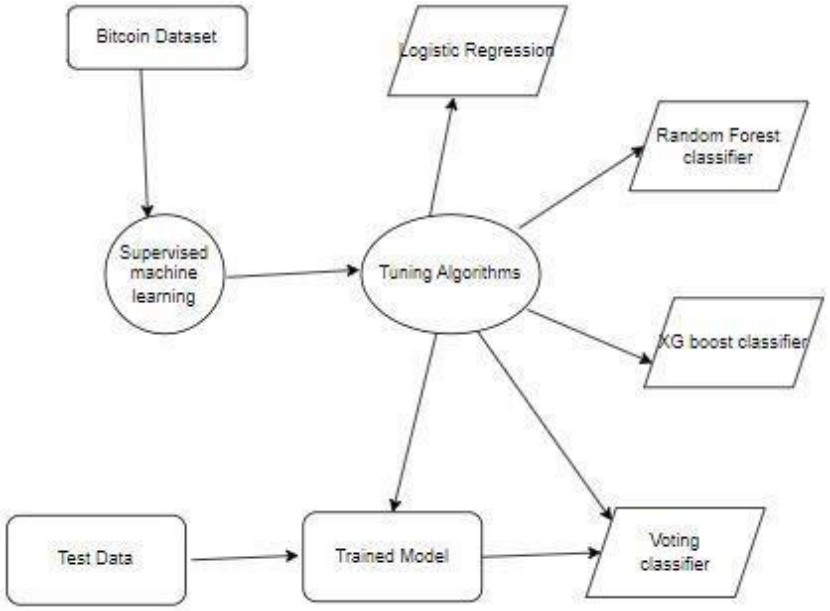
6.2 CLASS DIAGRAM



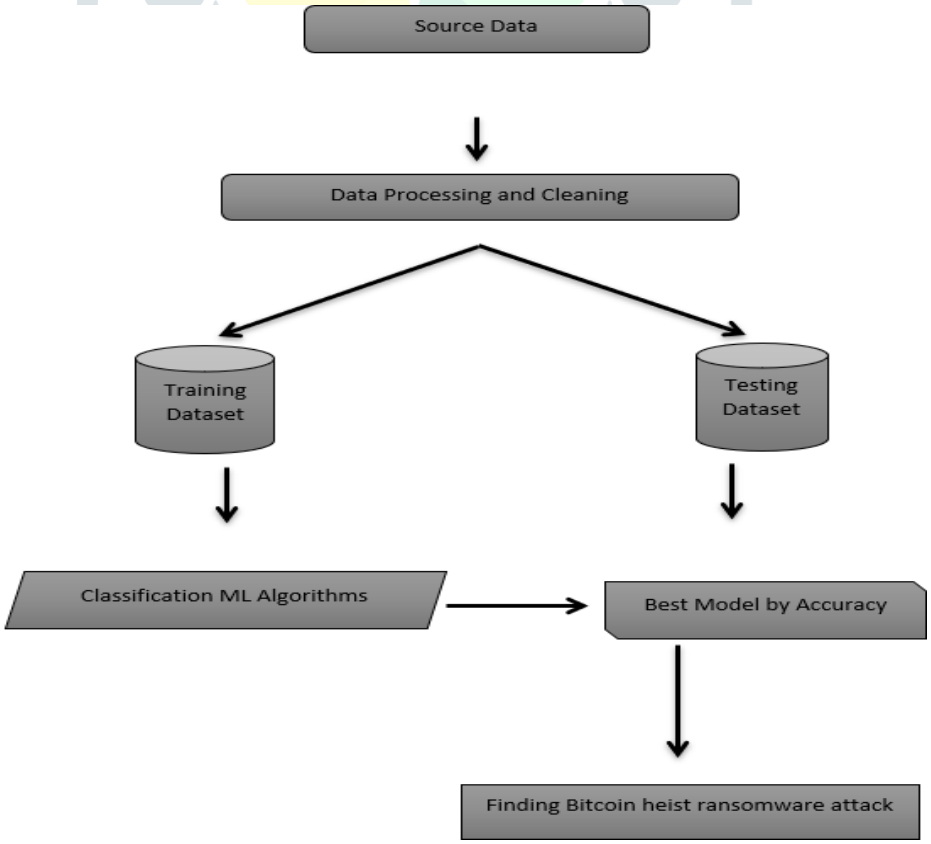
6.3 ACTIVITY DIAGRAM



6.4 ENTITY RELATIONSHIP DIAGRAM



6.5 WORKFLOW DIAGRAM



DATASET MODULES

Blockchain Data: Blockchain explorers like Blockchain.info or Bitaps provide APIs for accessing transaction data, block information, and more.
Cryptocurrency Exchanges: Exchanges often provide APIs for accessing historical trading data, including transactions, trading volumes, and price movements.
Bitcoin Address Clustering Data: Some research projects have attempted to cluster Bitcoin addresses belonging to the same entity based on transaction patterns. These datasets can provide insights into the flow of funds.
Security Incident Reports: Some organizations publish reports on security incidents related to cryptocurrencies. These reports might include details of past Bitcoin thefts or heists.
Market Data: Historical market data for Bitcoin, including price movements and trading volumes, can provide context for analyzing potential heists.
Social Media Data: Monitoring social media platforms for discussions related to Bitcoin and cryptocurrency can provide insights into sentiment and potential threats.
Machine Learning Datasets: There might be curated datasets specifically for machine learning tasks related to cryptocurrency analysis, though they might not directly focus on heist prediction.

VII. IMPLEMENTAION

Data Collection and Preprocessing: Collect the Bitcoin Heist dataset or a relevant dataset containing transactional data.
Preprocess the data by handling missing values, encoding categorical variables, and scaling numerical features.
Feature Engineering: Extract relevant features from the dataset that could help in predicting ransomware transactions. This may involve analyzing transaction patterns, addresses, transaction amounts, etc.
Model Selection: Choose a suitable machine learning model for ransomware prediction. Random forest classifiers are commonly used due to their effectiveness with high-dimensional data and ensemble learning capabilities.
Model Training: Split the dataset into training and testing sets.
Train the selected model on the training data.
Model Evaluation: Evaluate the trained model's performance using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.
Hyperparameter Tuning: Fine-tune the hyperparameters of the model to optimize its performance. This can be done using techniques like grid search or randomized search.
Cross-Validation: Validate the model's performance using techniques like k-fold cross-validation to ensure its robustness and generalization.
Deployment: Once a satisfactory model is obtained, deploy it in a production environment where it can be used to predict ransomware transactions in real-time.
Monitoring and Maintenance: Continuously monitor the performance of the deployed model and retrain it periodically to adapt to evolving patterns of ransomware attacks.

VIII. DOMAIN OF THE PROJECT

PYTHON

Python is a high-level, interpreter-based, object-oriented programming language featuring dynamic semantics. Its dynamic typing and dynamic binding, along with its high-level built-in data structures, make it an appealing language for Rapid Application Development and for usage as a scripting or glue language to join existing components. Because of its straightforward, basic syntax, Python promotes readability, which lowers software maintenance costs. Python's support for packages and modules promotes code reuse and program modularity. The large standard library and the Python interpreter are freely distributable and accessible for free on all major platforms in source or binary form.

Python's increased efficiency is one of the main reasons programmers fell in love with it. The edit, test, and debug cycle is extremely quick because there is no compilation step. Python program debugging is simple because segmentation faults are never caused by bugs or incorrect input. Rather, the interpreter raises an exception when it finds a mistake. The interpreter prints a stack trace if the application fails to catch the exception. Setting breakpoints, evaluating arbitrary expressions, inspecting local and global variables, stepping through the code one line at a time, and other features are all possible with a source level debugger. The fact that the debugger is developed in Python attests to the language's capacity for introspection.
However, adding a few print statements to the source code is frequently the fastest way to debug a program since it creates a short edit-test-debug cycle. This straightforward method is also highly effective.It includes anything from basic automated jobs to web development, gaming, and even sophisticated corporate systems.

On the other hand, adding a few print statements to the code is frequently the fastest way to debug a program because of how quickly the edit-test-debug cycle may be completed.Simple automation jobs, web development, games, and even sophisticated enterprise systems are all included.

PYCHARM

An easy-to-use environment for efficient Python, web, and data science development is created by PyCharm, an IDE specifically designed for Python developers. It offers a large selection of necessary tools for Python developers.
For details, see the editions comparison matrix.
Supported languages
To start developing in Python with PyCharm you need to download and install Python from python.org depending on your platform.
PyCharm supports the following versions of Python:
Python 2: version 2.7
Python 3: from the version 3.6 up to the version 3.10
Additionally, the Professional edition allows for the development of Pyramid, Flask, and Django applications. Additionally, it supports HTML (including HTML5), CSS, JavaScript, and XML to the fullest extent possible. These languages are included in the IDE by default and are packed with plugins. Plugins can be used to add support for additional languages and frameworks; to learn more about them or set them up on the first IDE run, go to Settings | Plugins or PyCharm | Preferences | Plugins for macOS users

IX. RESULT ANALYSIS

Data Preprocessing: This includes cleaning the dataset, handling missing values, encoding categorical variables, and scaling numerical features.
Feature Selection/Extraction: Identifying relevant features that contribute to the prediction of ransomware transactions. This could involve techniques like correlation analysis, feature importance ranking, or dimensionality reduction methods.
Model Training: Utilizing machine learning algorithms such as random forest classifiers, as mentioned, for training the model on the preprocessed dataset. Ensemble learning methods like random forest are suitable for handling high-dimensional data and capturing complex relationships between features.
Evaluation Metrics: Assessing the performance of the trained model using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC). These metrics help in understanding the model's ability to correctly classify ransomware transactions and distinguish them from non-ransomware transactions.
Cross-Validation: Employing techniques like k-fold cross-validation to ensure the robustness and generalization of the model by training and testing it on different subsets of the data.
Hyperparameter Tuning: Fine-tuning the parameters of the random forest classifier to optimize its performance. This could involve grid search or randomized search techniques to find the best combination of hyperparameters.
Result Analysis: Analyzing the results obtained from the trained model, including the confusion matrix to examine the true positive, true negative, false positive, and false negative predictions. Understanding the misclassifications can provide insights into areas for improvement.
Deployment and Monitoring: Once a satisfactory model is obtained, deploying it in a production environment for real-time prediction of ransomware transactions. Continuous monitoring and periodic retraining of the model may be necessary to adapt to evolving patterns of cyber threats.
ne can conduct a thorough analysis of the Bitcoin Heist dataset for ransomware prediction and develop an effective model for detecting and mitigating ransomware attacks in cryptocurrency transactions.

X. CONCLUSION

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set of higher accuracy score algorithm will be find out. The founded one is used in the application which can help to find the Bitcoin Heist randsomeware attack.

After testing six different machine learning algorithms, we discovered that Random Forest was the most accurate. To determine how well this model performs, you should test it using the test set. Random Forest served as the model. To further improve it, we can utilize more advanced models and train on models to forecast the different kinds of bitcoin heist problems and provide advice to users.

XI. REFERENCES

[1] Hesham Alshaikh, Nagy Ramadan, and Hesham Ahmed Hefny; "Ransomware Prevention and Mitigation Methods" Volume 177, No. 40, February 2020;
[2] On Utilizing Pre-Attack Paranoia Activity to Attribute Ransomware Families; Ricardo Khaled Sarieddine, Elias Bou-Harb, Sadegh Torabi, and Misael Ayala Molina; VOL. 19, NO. 1,
MARCH 2022

[3] J. Hernandez-Castro, A. Cartwright, and E. Cartwright; On August 30, 2022; "An Economic Analysis of Ransomware and Its Welfare Consequences"
[4] Context-aware AI in IoT Systems for Predictive Analysis of Ransomware Attacks; P.V. Lakshmi and Vytarani Mathane; Vol. 12, No. 4, 2021
[5] Bitcoin Transactions Associated with Ransomware: A Study; Sabira Karim, Shemitha PA; Vol. 7 Issue 3 2021
[6] Micheline Al Harrack, "The Bitcoinheist: Classifications Of Ransomware Crime Families," Vol. 13, No. 5, October 2021