Anomaly Detection System in Blockchain

Project Domain / Category

Blockchain/Machine Learning

Abstract / Introduction

With the growing adoption of Blockchain technologies, ensuring the integrity and security of decentralized systems like Bitcoin has become critical. This project will focus on providing an efficient method to identify malicious activity within the Blockchain, improving the security and reliability of decentralized networks.

This project proposes developing an **anomaly detection system** to detect a 51% attack in realtime by analyzing Blockchain transaction data using machine learning techniques. The proposed solution will involve injecting artificial anomalies into a Bitcoin dataset and then building machine learning models such as **SVM, Random Forest**, **AdaBoost**, and **XGBoost** to detect those anomalies.

Functional Requirements:

The functional requirements define the features that the system must implement to fulfill its purpose.

1. **Data Loading and Processing**:
   - The system will load the Bitcoin Blockchain dataset containing transaction data and relevant block information. (A dataset link will be provided)
   - It will preprocess the dataset by cleaning, transforming, and handling missing values or inconsistencies.
2. **Anomaly Injection**:
   - Artificial anomalies mimicking a 51% attack will be injected into the dataset. These anomalies will affect key features such as:
     - **Confirmations**: Set anomalous blocks to have zero or unusually low confirmations.
     - **Height**: Duplicate or irregular block heights.
     - **Number of Transactions**: Set anomalously high or low transaction volumes in certain blocks.
     - **Difficulty**: Simulate reduced difficulty in attack scenarios.
     - **Timestamps**: Create irregular or overlapping timestamps for the blocks.

3. **Anomaly Labeling**:
   - The system will label the dataset with a binary indicator (Anomaly column), where 1 represents an anomalous block, and 0 indicates normal blocks.

4. **Machine Learning Model Building**:
   - The system will implement four machine-learning algorithms for anomaly detection:
     - **Support Vector Machine (SVM)**
     - **Random Forest Classifier**
     - **AdaBoost Classifier**
     - **XGBoost Classifier**
   - These models will be trained to identify patterns associated with a 51% attack based on the labeled dataset.

5. **Model Evaluation**:
   - Each model will be evaluated using performance metrics such as:
     - **Accuracy**: Proportion of correctly classified instances.
     - **Precision**: Proportion of correctly predicted positive cases (anomalies).
     - **Recall**: Ability to detect true positives (actual anomalies).
     - **F1-score**: Harmonic mean of precision and recall.
   - Confusion matrices will be generated for each model to visualize the distribution of true positives, true negatives, false positives, and false negatives.

6. **Comparison of Models**:
   - The system will compare the performance of the models and identify the best algorithm based on the above metrics.
   - The results will be visualized using bar plots showing the performance metrics (accuracy, precision, recall, F1-score) for each model.

Tools:

Python (programming language)

Scikit-learn (Library)

Kaggle or Jupyter Notebook (open-source web application) or Google Colab Matplotlib (library)

Numpy (library for the python)

Supervisor:

Name: Fouzia Jumani Email ID: fouziajumani@vu.edu.pk

Skype ID: fouziajumani