International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Cryptosentinel: Crypto Currency Monitoring and Analysing System

Sahil Kumbharkar, (Computer Engineering , Trinity Polytechnic ,Pune, Email:sahilkumbharkar0@gmail.com)
Tanishka Shingote, (Computer Engineering , Trinity Polytechnic ,Pune Email:tanishkashingote2411@gmail.com)
Aditya Kamthe, (Computer Engineering , Trinity Polytechnic ,Pune Email:adityakamthe72@gmail.com)
Tanmay Kale (Computer Engineering Trinity Polytechnic ,Pune Email:ktanmay0506@gmail.com)

----------**********----------

Abstract:

Traditional methods of monitoring cryptocurrency transactions rely heavily on manual analysis and static rule-based systems, which are inefficient in handling the high volume, velocity, and complexity of blockchain data. These limitations hinder timely detection of suspicious activities such as abnormal transaction patterns, potential fraud, and illicit fund movements. This paper presents **CryptoSentinel: Cryptocurrency Monitoring and Analysis System**, an artificial intelligence and machine learning (AI/ML)-based solution designed to automate real-time monitoring and anomaly detection in Bitcoin transactions.

The system integrates real-time blockchain data streams using APIs such as Blockchain.com and Blockstream, extracting key transaction attributes including transaction amount, fee, number of inputs and outputs, timestamp, block height, and sender–receiver addresses. A feature engineering pipeline processes these attributes into structured numerical representations, which are then analysed using the **Isolation Forest** algorithm for unsupervised anomaly detection.

The model identifies deviations from normal transaction behaviour and assigns a risk score, enabling classification into severity levels such as Low, Medium, High, and Critical. The system is implemented with an interactive web-based dashboard using Streamlit, providing real-time visualization of transactions, anomaly distributions, and a dedicated interface for flagged transactions with complete details.

Keywords —Cryptocurrency monitoring, blockchain analytics, anomaly detection, Isolation Forest, Bitcoin transactions, machine learning,.

----------**********----------

I. INTRODUCTION

Cryptocurrency monitoring and analysis have become increasingly critical in the modern digital financial ecosystem, where decentralized blockchain networks facilitate high-speed, borderless transactions. Bitcoin and other cryptocurrencies operate on transparent yet complex distributed ledgers, making them powerful tools for financial innovation while also posing challenges in tracking suspicious or anomalous activities. Accurate monitoring of transaction flows, identification of irregular patterns, and timely detection of potential fraud are essential for ensuring security, regulatory compliance, and trust in blockchain-based systems. However, traditional

approaches to cryptocurrency analysis rely heavily on manual inspection, static rule-based systems, and basic data visualization tools, which are inadequate for handling the massive scale, velocity, and dynamic nature of blockchain data.

With the rapid growth of blockchain technology and the increasing volume of transactions, there is a need for intelligent systems capable of automated monitoring and real-time analysis. Advances in machine learning and data analytics provide an opportunity to transform cryptocurrency monitoring by enabling systems to detect anomalies, identify suspicious transaction behaviour, and generate actionable insights without requiring labeled datasets. In particular, unsupervised learning techniques such as the **Isolation Forest** have demonstrated strong performance in identifying outliers within large and complex datasets, making them well-suited for blockchain transaction analysis where fraudulent patterns are often unknown and evolving.

Existing research in blockchain analytics has explored areas such as transaction graph analysis, wallet clustering, fraud detection using supervised learning, and visualization of blockchain networks. However, a significant gap remains in the development of integrated, end-to-end systems that combine real-time data acquisition from blockchain APIs, automated feature extraction, machine learning-based anomaly detection, and interactive visualization within a single, user-friendly platform. Most available tools are either highly technical, fragmented, or lack real-time capabilities, limiting their usability for students, researchers, and non-specialist users.

CryptoSentinel: Cryptocurrency Monitoring and Analysis System addresses this gap by providing a unified AI/ML-based solution for real-time Bitcoin transaction monitoring and anomaly detection. The system retrieves live transaction data from blockchain APIs, extracts key features such as transaction amount, fee, input-output structure, timestamp, and wallet addresses, and processes them using an Isolation Forest model to detect anomalous behaviour. The results are presented through an interactive Streamlit-based dashboard, enabling users to visualize transaction patterns, identify suspicious activities, and analyse flagged transactions with detailed information including sender, receiver, block height, and severity levels.

The system also supports user-uploaded datasets, allowing flexibility in analysis and experimentation. By integrating real-time data processing, machine learning, and intuitive visualization, CryptoSentinel provides a scalable, efficient, and practical solution for cryptocurrency monitoring. This paper presents the design, implementation, and evaluation of CryptoSentinel, demonstrating its effectiveness as a modern tool for blockchain analysis, cybersecurity applications, and financial data monitoring.

II. Materials and Methods

A. System Architecture

CryptoSentinel adopts a modular, layered architecture comprising four principal components: a data acquisition layer, a feature engineering module, a machine learning inference engine, and a presentation and visualization layer. The architecture is designed to ensure scalability, flexibility, and maintainability by enabling loose coupling between system components, allowing independent updates and integration of new data sources or analytical models without affecting overall system functionality.

The **data acquisition layer** is responsible for retrieving real-time Bitcoin transaction data from blockchain APIs such as Blockchain.com and Blockstream. This layer collects essential transaction attributes including transaction amount, transaction fee, number of inputs and outputs, timestamp, block height, and sender–receiver wallet addresses. To ensure reliability, the system incorporates caching mechanisms and controlled refresh intervals to handle API rate limits and network constraints.

The **feature engineering module** processes the raw transaction data and transforms it into a structured numerical feature set suitable for machine learning. This includes normalization of transaction values, extraction of behavioural indicators such as input-output ratios and transaction complexity, and preparation of feature vectors that capture the statistical characteristics of transaction patterns.

The **machine learning inference engine** utilizes the **Isolation Forest** to perform unsupervised anomaly detection. The model is trained on transaction data to learn normal behavioural patterns and identify deviations. It generates anomaly scores that are further converted into risk scores, which are used to classify transactions into severity levels such as Low, Medium, High, and Critical. This enables efficient identification of suspicious transactions without requiring labeled datasets.

The **presentation and visualization layer** is implemented using a Streamlit-based web interface, providing an interactive dashboard for

real-time monitoring and analysis. This layer displays transaction data in tabular and graphical formats, highlights anomalous transactions, and offers a dedicated view for flagged transactions with detailed information including sender and receiver addresses, transaction amount, timestamp, block height, and severity level. Additionally, the system supports dataset upload functionality and allows users to export analysis results in CSV format for further investigation.

Layer	Component	Technology / Source
Data Acquisition	Bitcoin Transaction Data	Blockchain.com API / Blockstream API
Data Acquisition	Transaction Metadata	Block Height, Timestamp, Fee, Inputs/Outputs (Blockchain APIs)
Data Acquisition	Wallet Address Information	Sender–Receiver Addresses (Blockchain APIs)
Feature Engineering	Preprocessing & Normalisation	Python (pandas, scikit-learn)
Feature Engineering	Feature Extraction	Transaction Amount, Fee, Input/Output Count, Behavioural Metrics
ML Inference	Anomaly Detection Model	Isolation Forest
ML Inference	Risk Scoring & Severity Classification	Custom Logic (Low, Medium, High, Critical)
Presentation	Web Dashboard	Streamlit (Python)
Presentation	Data Visualisation	Plotly / Matplotlib
Presentation	Real-Time Monitoring	Auto-refresh + API Integration
Presentation	Report Export	CSV Download

Table I: System Architecture Components and Technologies

B. Data Sources and API Integration

The system retrieves real-time Bitcoin transaction data from blockchain-based APIs, primarily leveraging services such as

Blockchain.com and Blockstream. These APIs provide access to live blockchain data, enabling the system to extract detailed transaction-level information including transaction amount, transaction fee, number of inputs and outputs, timestamp, block height, and sender–receiver wallet addresses.

Each transaction is uniquely identified using its transaction hash, and relevant metadata are parsed from API responses in JSON format. The system focuses on extracting both structural and behavioural attributes of transactions. Structural attributes include input and output counts, which represent the complexity of the transaction, while behavioural attributes include transaction value, fee patterns, and temporal characteristics. Sender and receiver addresses are derived from input and output scripts, providing insight into fund flow between wallets.

API requests are dynamically generated to retrieve data from the latest blocks in the Bitcoin network. The system employs controlled request intervals and caching mechanisms to handle API rate limits and reduce redundant calls. Error handling strategies are implemented to manage scenarios such as API timeouts, incomplete responses, or network failures, ensuring system robustness and continuity of operation. In cases where primary APIs fail, fallback mechanisms can be integrated to maintain data availability.

C. Feature Engineering

The raw transaction data obtained from blockchain APIs undergo a structured feature engineering pipeline to transform it into a format suitable for machine learning analysis. Key numerical features extracted include transaction amount (in BTC), transaction fee, number of inputs, number of outputs, and derived behavioural indicators such as transaction complexity and flow characteristics.

Continuous variables such as transaction amount and fee are scaled to ensure uniformity across features with different magnitudes. The system applies normalization techniques using tools from Python libraries such as pandas and scikit-learn, enabling improved model performance and stability. Temporal features, such as timestamps, are retained for contextual analysis but are not directly used in model training unless transformed into derived metrics such as transaction frequency.

The final feature vector used for anomaly detection consists of four primary dimensions: transaction amount, transaction fee, input count, and output count. These features are selected based on their relevance in capturing deviations from normal transaction behaviour and their effectiveness in identifying unusual patterns such as high-value transfers, abnormal transaction structures, or suspicious activity bursts.

This feature set is designed to balance computational efficiency with analytical effectiveness, ensuring that the system can process real-time transaction data while maintaining reliable anomaly detection performance using the **Isolation Forest**.

D. Machine Learning Models

CryptoSentinel employs an unsupervised anomaly detection approach using the **Isolation Forest**, which is well-suited for identifying unusual patterns in large, unlabeled datasets such as blockchain transactions. Unlike traditional supervised learning models that require labeled data, Isolation Forest detects anomalies by isolating data points through recursive partitioning.

The algorithm constructs an ensemble of isolation trees, where each tree is built by randomly selecting features and split values. Anomalous data points, being rare and significantly different from normal observations, tend to be isolated in fewer splits, resulting in shorter path lengths within the tree structure. These path lengths are aggregated across the ensemble to compute an anomaly score for each transaction.

The model is trained on transaction features including transaction amount, transaction fee, number of inputs, and number of outputs. Prior to training, the data undergoes normalization using standard scaling techniques to ensure uniform feature contribution. The contamination parameter is used to define the expected proportion of anomalies within the dataset, enabling the model to establish a threshold for anomaly detection.

During inference, the model generates anomaly scores, which are subsequently transformed into a normalized risk score ranging from 0 to 100. Based on this risk score, transactions are categorized into severity levels: Low, Medium, High, and Critical. Transactions identified as anomalous are flagged as suspicious and displayed in the monitoring dashboard, enabling users to focus on potentially risky activities.

E. Graphical User Interface

The graphical user interface (GUI) of CryptoSentinel is implemented using the Streamlit framework, providing a lightweight, interactive, and web-based dashboard for real-time cryptocurrency monitoring and analysis. The interface is designed to be intuitive and accessible, enabling users to interact with complex blockchain data without requiring advanced technical expertise.

The primary interface consists of multiple functional sections, including real-time monitoring and dataset upload modes. In real-time mode, the system fetches live Bitcoin transactions from blockchain APIs and displays them in a structured tabular format, along with key attributes such as transaction amount, fee, timestamp, block height, and sender–receiver addresses. An integrated anomaly detection module processes this data and highlights suspicious transactions.

A dedicated section for flagged transactions presents detailed information about anomalies, including severity level and risk score. Visualizations such as scatter plots and distribution graphs are incorporated using Plotly and Matplotlib to provide insights into transaction behaviour and anomaly distribution. The interface also includes interactive controls such as refresh buttons and configurable data limits.

In dataset upload mode, users can upload custom transaction datasets in CSV format for analysis. The system processes the uploaded data, applies the anomaly detection model, and presents the results within the same dashboard environment. Additionally, users can export the analysed results as CSV files for further investigation and reporting purposes.

F. Testing Methodology

The validation of CryptoSentinel was conducted through a comprehensive multi-level testing strategy to ensure system reliability, performance, and usability. Unit testing was performed to verify the correctness of individual modules, including data fetching functions, feature preprocessing routines, and model inference logic. Mock data and controlled inputs were used to isolate components and validate expected outputs.

Integration testing evaluated the interaction between different system components, ensuring that data retrieved from blockchain APIs was correctly processed, transformed into feature vectors, and passed to the machine learning model for accurate anomaly detection. This phase also verified seamless communication between backend processing and frontend visualization.

System testing was conducted to assess the complete end-to-end workflow under real operating conditions. This included real-time transaction fetching, anomaly detection, and visualization of flagged transactions within the dashboard. Performance testing focused on measuring API response times, data processing latency, and model inference speed, particularly under repeated refresh scenarios.

Usability testing was carried out with representative users to evaluate the effectiveness of the interface design, ease of navigation, and clarity of presented information. Feedback from users was used to refine the dashboard layout, improve interaction mechanisms, and enhance overall user experience. The results of testing demonstrate that CryptoSentinel is a stable, efficient, and user-friendly system suitable for real-time cryptocurrency monitoring and analysis.

III. Results and Discussion

A. Anomaly Detection Performance

The anomaly detection model was evaluated using transaction datasets comprising both synthetic and real-time Bitcoin transactions. Since the system employs an unsupervised learning approach using the **Isolation Forest**, traditional classification metrics such as accuracy and recall are not directly applicable in the absence of labeled ground-truth fraud data. Instead, model performance was assessed based on its ability to identify statistically significant deviations from normal transaction behaviour.

The Isolation Forest model demonstrated effective separation between normal and anomalous transactions, successfully identifying high-value transfers, unusual transaction structures (high input/output counts), and abnormal fee patterns. The contamination parameter was tuned to 0.05, allowing the model to flag approximately 5% of transactions as anomalous. Risk scores generated by the model exhibited a clear distribution, enabling meaningful categorization into severity levels.

Model	Approach Type	Detection Capability	Interpretability
Isolation Forest	Unsupervised	High (Anomaly Detection)	Moderate

Table II: ***Anomaly Detection Model Characteristics***

B. System Testing Results

A total of 20 structured test cases were executed across five testing tiers to evaluate the robustness and correctness of the CryptoSentinel system. All test cases passed successfully, demonstrating reliable end-to-end functionality across data acquisition, processing, anomaly detection, and visualization components.

Minor usability issues were identified during testing, including slight delays in dashboard refresh, column alignment inconsistencies in data tables, and limited responsiveness under high-frequency refresh conditions. These issues were classified as low severity and were resolved through interface optimization and improved caching mechanisms.

Testing Tier	Test Cases	Passed	Issues Identified	Severity
Unit Testing	6	6	0	—
Integration Testing	4	4	0	—
System Testing	4	4	0	—
Performance Testing	3	3	0	—
Usability Testing	3	3	3 (UI)	Low
Total	**20**	**20**	**3 (UI)**	**Low**

Table III: Summary of Testing Results Across All Evaluation Tiers

C. Performance Metrics

The system's performance was evaluated based on end-to-end processing latency, measured from transaction data retrieval to anomaly detection and visualization. Under standard network conditions (≥10 Mbps), the average processing time was approximately 3.8 seconds.

The primary contributor to latency was external API data retrieval, accounting for nearly 70% of total processing time. Feature engineering and model inference were computationally efficient, requiring less than 0.3 seconds due to the lightweight nature of the selected feature set and the efficiency of the Isolation Forest algorithm. Dashboard rendering and visualization contributed an additional 0.5 seconds on average.

These results indicate that CryptoSentinel is capable of near real-time monitoring and is suitable for interactive analytical applications where timely detection of suspicious transactions is critical.

D. Discussion

The results demonstrate that CryptoSentinel provides an effective and scalable solution for real-time cryptocurrency monitoring and anomaly detection. The use of the Isolation Forest algorithm enables the system to identify suspicious transaction patterns without requiring labeled datasets, making it particularly suitable for blockchain environments where fraudulent behaviour is dynamic and not always explicitly defined.

The integration of real-time blockchain data with machine learning and interactive visualization offers a significant advantage over traditional rule-based monitoring systems. By analysing transaction features such as amount, fee, and input-output structure, the system is able to detect irregularities that may indicate potential fraud, abnormal activity, or unusual transaction behaviour.

However, the system has certain limitations. Its performance is dependent on the availability and reliability of external blockchain APIs. Network delays, rate limits, or API failures can impact data acquisition and overall system responsiveness. Additionally, while the anomaly detection model effectively identifies unusual patterns, it does not explicitly classify transactions as fraudulent, as it operates without labeled training data.

Future enhancements may include the integration of multiple API sources to improve reliability, implementation of real-time streaming via WebSockets, and incorporation of graph-based analysis techniques to better understand wallet-to-wallet relationships. Expanding the feature set and incorporating semi-supervised or supervised learning approaches could further improve detection accuracy and interpretability.

The risk scoring and severity classification mechanism provides users with a clear and intuitive understanding of transaction risk levels, enabling efficient prioritization of suspicious activities. This enhances decision-making capabilities and supports practical applications in cybersecurity, financial monitoring, and blockchain analytics.

IV. Conclusions

This paper has presented **CryptoSentinel: Cryptocurrency Monitoring and Analysis System**, an AI/ML-based solution that integrates real-time blockchain data acquisition, automated feature engineering, anomaly detection using the **Isolation Forest**, and an interactive web-based dashboard into a unified end-to-end platform. The system is designed to monitor Bitcoin transactions, identify anomalous patterns, and classify transactions into risk severity levels including Low, Medium, High, and Critical.

The proposed system demonstrates effective performance in detecting unusual transaction behaviour, successfully identifying anomalies based on transaction amount, fee structure, and input-output characteristics. The lightweight feature set and efficient model design enable rapid processing, making the system suitable for near real-time monitoring scenarios. The integration of real-time data retrieval with machine learning and visualization provides a practical and scalable approach to cryptocurrency analysis.

Comprehensive testing across unit, integration, system, performance, and usability levels confirmed the functional correctness and stability of the system, with all test cases executed successfully. The system achieves low processing latency, with the majority of time attributed to external API data retrieval, while feature engineering and model inference remain computationally efficient. Minor usability issues identified during evaluation were addressed through interface enhancements and optimization of refresh and caching mechanisms.

CryptoSentinel represents a significant step toward the automation and intelligent monitoring of blockchain transactions, offering improvements in detection capability, speed, and usability compared to traditional manual or rule-based analysis methods. The system is designed to serve a wide range of users, including cybersecurity professionals, financial analysts, researchers, and regulatory authorities involved in cryptocurrency monitoring and fraud detection.

Future work will focus on enhancing system robustness through multi-source API integration, implementing real-time streaming using WebSocket-based data feeds, and incorporating graph-based analysis techniques to better understand wallet-to-wallet transaction networks. Additionally, extending the system to support supervised or semi-supervised learning approaches with labeled datasets could further improve detection accuracy and interpretability in complex fraud scenarios.

Acknowledgment

The authors express their sincere gratitude to the developers and maintainers of the open data platforms and tools utilized in this research, including Blockchain.com and Blockstream for providing access to real-time Bitcoin transaction data. The authors also acknowledge the contributions of the open-source community, particularly the developers of Python libraries such as pandas, scikit-learn, and Streamlit, which played a crucial role in the implementation of this system.

Special thanks are extended to the institution and faculty members for their guidance, support, and

resources that facilitated the successful development of this project. The authors are also grateful to peers and contributors who provided valuable feedback during the testing and refinement phases of the system.

REFERENCES

[1] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Pisa, Italy, 2008, pp. 413–422.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[3] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.

[4] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[6] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[7] Streamlit Inc., "Streamlit: The fastest way to build data apps," 2024. [Online]. Available: https://streamlit.io

[8] Blockchain.com, "Blockchain explorer and API," 2024. [Online]. Available: https://www.blockchain.com

[9] Blockstream, "Blockstream API documentation," 2024. [Online]. Available: https://blockstream.info/api

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[11] M. Conti, S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of Bitcoin," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[12] D. Ron and A. Shamir, "Quantitative analysis of the full Bitcoin transaction graph," in *Financial Cryptography and Data Security*, 2013.

[13] M. Fleder, M. S. Kester, and S. Pillai, "Bitcoin transaction graph analysis," *arXiv preprint arXiv:1502.01657*, 2015.

[14] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

