Detecting Malicious Behaviors on Ethereum

Hallee Ray
Computer Science Post Bacc.
University of Colorado - Boulder
Boulder, Colorado, USA
hara7620@colorado.edu

BD Tinsley
Computer Science Post Bacc.
University of Colorado - Boulder
Boulder, Colorado, USA
beti7384@colorado.edu

Evan Bean
Computer Science Post Bacc.
University of Colorado - Boulder
Boulder, Colorado, USA
evbe5723@colorado.edu

ABSTRACT

This project aims to analyze transaction data from the Ethereum blockchain to detect fraudulent or suspicious activities using data mining techniques. By utilizing real-time data and feature engineering, we will develop models that can predict malicious behaviors, identify trends, and detect anomalies in blockchain transactions. We also aim to enhance our dataset by incorporating public interest data from Google Trends and findings from FBI reports. Our final objective is to build a system that can flag potentially fraudulent wallets and predict suspicious activity before it escalates.

KEYWORDS

Ethereum blockchain, anomaly detection, data mining, prediction modeling, fraud detection, feature engineering, machine learning

1 PROBLEM STATEMENT/MOTIVATION

We will mine data from the Ethereum blockchain to discover trends in transactions. Through this analysis, we aim to detect fraudulent or scam-like trends involving investment fraud, kiosk scams, and other malicious behaviors. Our goal is to answer several interesting questions related to the relationships between different features of Ethereum transactions. One question we seek to explore is whether there is a correlation between the time of the transaction and the amount of the transaction. This could help us identify patterns regarding the time of day or night during which larger transactions are reported. Intuitively, we suspect that transactions for large amounts occurring outside of normal trading hours may be more likely to be fraudulent.

Another question we aim to investigate is the average time between a new coin going live and its first purchase. Exploring this question could provide insights into potential pump-and-dump activities. For instance, if there is a higher-than-average volume of transactions within the first 24 hours of a coin becoming available for trading, it could drive up the price of the coin, leading to a quick sell-off by investors while the price is artificially inflated.

Additionally, we are interested in the average values of transactions during the lifetime of Ethereum and how they map to early-stage hype. This analysis could help us recognize patterns related to a coin's overall health in the market. For coins with a history of fraudulent activity, we will examine the general statistics of the transactions that were reported as fraudulent. This investigation could allow us to establish guidelines for identifying transactions that should be flagged as potentially fraudulent.

By analyzing these questions, we aim to gain a deeper understanding of Ethereum transactions and improve our ability to predict whether a transaction is likely to be fraudulent.

2 LITERATIVE SURVEY

2.1 The Literature of Fraud Data and Identification

Unfortunately, scams and fraudulent activity in cryptocurrency occur frequently and are difficult to monitor due to the anonymous and decentralized nature of cryptocurrency. The Federal Bureau of Investigation (FBI) and the Homeland Defense & Security Information Analysis Center (HDIAC) have made efforts to monitor and predict fraudulent behavior in cryptocurrency.

"In February 2022, the FBI formed the Virtual Assets Unit (VAU), a specialized team dedicated to investigating cryptocurrency-related crimes" [5]. In the report, the FBI identified common types of malicious behaviors using cryptocurrencies: investment fraud, recovery schemes, and kiosk scams. Investment fraud is described as malicious actors who encourage individuals to make "investments" in cryptocurrency. The FBI found that losses from cryptocurrency-related investment fraud schemes reported to the IC3 rose from $2.57 billion in 2022 to $3.96 billion in 2023, an increase of 53

Recovery schemes involve representatives from fraudulent businesses claiming to provide cryptocurrency tracing and promise an ability to recover lost funds. They request an upfront fee for this service and then immediately cease communication with the sender once the funds are received. Kiosk scams are scams where a criminal actor will direct an individual to withdraw cash from their bank and locate and send money via a kiosk or ATM-like machine. In 2023, the IC3 received more than 5,500 complaints reporting the use of cryptocurrency kiosks, with losses over $189 million.

Knowing the existence of these scams and monitoring statistical metrics regarding cryptocurrency transactions and fraudulent activity is not sufficient to predict and prevent such scams. The Homeland Defense & Security Information Analysis Center (HDIAC) has employed various real-time predictive machine learning models and on- and off-chain monitoring to detect fraudulent activity. The HDIAC describes the purpose of the off-chain module as preventing fraud before it occurs, whereas the on-chain module utilizes real-time surveillance to detect fraud after it has occurred [8].

2.2 The Literature of Formal Fraud Detection Methods

2.2.1 Formal Methods. In a manuscript submitted in August of 2024, Jin et. al from Zhejiang University of Technology, China discuss methods of designing fraudulent action detection system in the Ethereum Blockchain. Noting, as we have, that many forms of fraud require expert systems with subject matter expertise to rule on what seems "irregular" and what does not, Jin suggested using transaction-graph-based methods of anomaly detection to target individual malicious accounts and behaviors. At its core, the methodology is pattern recognition and segregation. The challenge of this approach, however, is gracefully handling complex, frequent, and imbalanced data. As one might expect, some perfectly good actors will trade at a low volume and frequency – consider the armchair cryptocurrency enthusiast, while other good actors will technical capabilities and more personal investment in cryptocurrency might seek out arbitrage, small-margin, or options plays, all of which characteristically "look" very different. To be able to segregate the good from the bad, one must categorize these different behavioral patterns, normalize the weighting that may be associated with high volumes of varying types of trades or traders so that a predictive engine is not overly focused on a majority, but rather equally focused on varying patterns. Within graph analysis, we will follow and expand upon the methodologies presented by Jin. This will be discussed in greater detail below. [1]

3 PROPOSED WORK

The work that we will need to perform in order to progress through this project involves **data preprocessing**, **exploratory data analysis**, **feature engineering**, **anomaly detection**, and **prediction modeling**.

3.1 Data Preprocessing

For data preprocessing, there are two steps that will need to be completed: **data cleaning and preparation**, and **normalization and standardization**. This can be achieved by handling any missing values in a variety of ways, such as:

- **Dropping records** that cannot be filled in through integrating multiple sources.
- **Filling in missing values** with a predetermined default value.
- **Imputing missing values** based on similar records.

Additionally, the fields of each feature will need to be analyzed and converted to the correct **data types** and formats.

3.2 Exploratory Data Analysis

Exploratory data analysis will help us uncover inter- and intra-feature relationships. **Univariate analysis** and **visualization** will allow us to identify outliers and anomalous values. Visualizing a **heatmap** of the correlations between features will allow us to understand general trends between features.

3.3 Feature Engineering

An important step in feature engineering is deriving new attributes, such as:

- New attributes for **transaction time intervals**, including **in-degree** and **out-degree** for sender and recipient addresses.
- Grouping various features according to their **transaction patterns** and frequency.

3.4 Anomaly Detection

One of the main objectives of this project is to detect anomalies that could potentially be fraudulent. Statistical methods such as **Z-score** and **Percentile-Based Detection** can be utilized to isolate transaction values of unusually high or low values. We can use the same methods to identify feature values that fall well outside the standard range.

Distance-based methods can be employed to detect anomalies. By defining several dimensions such as **time_between_transactions** and **gas_price** (based on gas and value), we can measure the distance between transactions and compare the record to its nearest neighbors using the **K-Nearest Neighbors (KNN)** methodology. Outliers here will also show interesting trends.

Using **K-means Clustering**, we can determine address behavior based on clusters by transaction activity, including:

- **Total transaction value**
- **Transaction frequency**
- **Gas usage**

Addresses that fall into distant clusters or small clusters could indicate suspicious behavior.

3.5 Prediction Modeling

Finally, we propose **prediction modeling**. Using statistics defined in **FBI's Internet Crime Complaint Annual Reports**, we will enhance our dataset by incorporating fraud patterns from FBI statistics, such as:

- Prevalence of specific fraud types.
- Trends in fraudulent activity over time.

Labeling the historical transaction data and using a **Random Forest** supervised machine learning model will help us discover potential fraudulent activity. Using real-time Ethereum data and our predictive models, we aim to **predict and mark transactions as potentially fraudulent** as they occur.

4 DATASET

The datasets that we will utilize for this project are an Ethereum ledger that provides real-time transaction data showing transaction type, the sender and recipient wallets, and the transaction amount via API integration [4]. Additionally, we contrast our dataset with maps of public interest in specific Ethereum-built cryptocurrencies using Google Trends [7]. Finally, we will compare our results to FBI findings to determine if we can detect trends before complaints of fraudulent activity using the FBI Fraud Report [6].

5 EVALUATION METHODS

To quantitatively evaluate the performance of our fraud detection models, we will employ several metrics and methods:

5.1 Performance Metrics

We will assess model performance using standard classification metrics such as:

- **Accuracy**: The ratio of correctly predicted instances over the total instances.
- **Precision and Recall**: Critical in fraud detection, precision measures the fraction of actual frauds in the predicted fraud cases, while recall measures the fraction of actual frauds correctly identified. The **F1-Score**, a harmonic mean of precision and recall, will also be calculated to balance both.
- **ROC Curve & AUC**: We will use the ROC curve to visualize the trade-off between true positive and false positive rates

and calculate the AUC as a quantitative measure of model discrimination.

5.2 Cross-Validation

We will perform **k-fold cross-validation** to ensure that our model's performance generalizes well across different subsets of data. This will mitigate overfitting and provide more robust performance metrics.

5.3 Confusion Matrix

A **confusion matrix** will be generated to break down our predictions into true positives, false positives, true negatives, and false negatives. This allows us to analyze misclassifications in more detail.

5.4 Threshold Tuning

By adjusting the **decision threshold**, we will explore how varying thresholds impact precision and recall, allowing us to tune the model based on the relative costs of false positives and false negatives.

5.5 Anomaly Detection Metrics

For anomaly detection, we will calculate **Z-scores** and use **distance-based metrics** (e.g., K-Nearest Neighbors) to measure how well the model isolates outliers from the normal transaction patterns.

5.6 Scalability and Time Complexity

We will evaluate the **time complexity** of our models and their ability to scale by testing performance on progressively larger subsets of Ethereum transactions.

5.7 Feature Importance

Using a **Random Forest** model, we will assess the importance of various features (e.g., transaction amount, gas price, time intervals) in predicting fraudulent behavior. Feature importance scores will be used to identify the most impactful features.

5.8 Baseline Comparison

We will compare the performance of our models against simpler baselines such as logistic regression, measuring improvements in precision, recall, and AUC over the baseline models.

5.9 Statistical Significance

We will conduct **paired t-tests** and **ANOVA tests** to determine whether improvements in performance metrics between different models are statistically significant.

6 TOOLS

The programming language that we will use for this project is **Python**, the beloved language for data science projects. It has a vast ecosystem of libraries for data analysis and integrates well with third-party services. Two useful libraries for our objectives include **SciPy** and **NumPy**. Both extend Python to include a range of computing tools to handle matrix operations, statistical analysis, optimization, and signal processing. Additionally, the **Pandas** library will be necessary in our exploration of the data due to its ability to handle structured datasets. It will be useful in most steps of this project but will especially excel in data cleaning, EDA, and feature engineering.

Due to the size of the dataset, we may encounter some limitations with Pandas' processing speed. Due to the size of the Ethereum blockchain ledger, we will likely be viewing and previewing our work within small time frames to take advantage of Python and Pandas processing power alone. However, when evaluating larger time frames of Ethereum transactions, **Dask** will be necessary to parallelize Python processing.

Visualization libraries such as **Matplotlib**, **Seaborn**, and **Plotly** will be greatly helpful in our data exploration and performance analysis. Data visualization will be necessary to display outliers and trends we uncover. Matplotlib will be useful for rudimentary visualizations, but we will likely reach for Seaborn for more control over aesthetics and Plotly for interactive visualizations.

GitHub will be a necessary tool for this project, both from a grading perspective and as a way to manage multiple contributors to the same codebase by offering a single source of truth, backups, and trackable changes. Admittedly, the task for **Snowflake** may be beyond the need for this project, but due to the large volume of data in evaluating Ethereum transactions, we may need more scalability than our standard configuration.

7 MILESTONES

Milestone 0: Proposal (October 14)
This encompasses parts 1 and 2 of the project requirements. This integrates any peer or instructor feedback from our initial proposal in order to finalize and improve our project proposition.

Milestone 1: Pull Dataset & Data Cleaning (October 18)
In this milestone, we determine the specific windows of Ethereum transactions we will use for the project and clean them in order to successfully employ our data mining objectives.

Milestone 2 & 3: EDA & Feature Engineering (October 25)
This milestone will allow us to gain insights into the dataset and uncover important information regarding the features. This milestone is a crucial step toward uncovering patterns and relationships between features.

Milestone 4: Progress Report (October 28)
As required by the course, this will be a mid-project check-in as defined in part 3 of the project requirements. This will be a good chance for us to review our progress, review our plan, and reassess our goals.

Milestone 5: Anomaly Detection (November 8)
This milestone is another important step toward project completion. By this point, we should all have a decent understanding of the data and which attributes are most implicated in fraudulent transactions, and we can shift focus to understanding which metrics would indicate suspicious activity.

Milestone 6: Prediction Modeling & Analysis (December 2)
Our goal to implement prediction modeling is contingent upon the successful completion of milestone 5. This milestone generally encompasses how we could apply the results of data mining in practice.

Milestone 7: Final Report (December 9)
As required by the course, this final report will highlight our procedures, findings, and next steps for this project. It will provide an opportunity to reflect on our processes, what did and did not work well, and provide an overview of our understanding of the material.

8 PROPOSAL REVIEW

8.1 Milestones

Two additional milestones not included in the project proposal, but are required for the course are added:

Milestone 8: Presentation (December 9)
As required by the course, this the final presentation of our findings. Originally omitted in the project proposal.

Milestone 9: Project & Code Description (December 9)
As required by the course, this final code update for the project along with links to presentation, final report and outline of our findings. Originally omitted in the project proposal.

9 MILESTONES COMPLETED

Milestone 0: Proposal
This was the prior deliverable and was completed October 14, 2024.

Milestone 1: Pull Dataset & Data Cleaning
This milestone proved more challenging than anticipated. Ethereum transaction data is abundant and accessible, but only through paid third party services or solutions that require storage well beyond the common size on typical home laptops and computers. We initially worked with a sample dataset of 500 transactions, but knew the findings would not be adequate for later milestones. We used the sample dataset for data cleaning purposes. We opted for a paid service for streaming access to pull approximately 140,000 records daily, until our fetch limits are reached. Through this, we can select a high degree of transactions from specific windows to validate our findings and develop and train a prediction model.

Due to the high degree of validation in a blockchain transaction, we found the values to be high fidelity and trustworthy. As Ethereum evolves, new attributes are added to each transaction and we were able to remove several null value attributes that were added in the London Hard Fork update [2] (integrated August 2021) and the Proto-Danksharding update [3] (integrated March 2024). Noting that we will need to adequately include these attributes in windows that occur after the update timestamp, for the initial portion of our work we cleaned these features.

Completed October 24, 2024.

Milestone 2: EDA
Completed October 27, 2024.

Milestone 4: Progress Report
This is self-referential as this progress report correlates to this milestone. Completed October 28, 2024.

10 MILESTONES TODO

Milestone 3: Feature Engineering (October 25)
Slated to be completed by October 25, we are running behind schedule completing this milestone. However, due to high degree of fidelity in the dataset, we believe we can make up for lost time

Milestone 5: Anomaly Detection (November 8)

Milestone 6: Prediction Modeling & Analysis (December 2)

Milestone 7: Presentation (December 9)

Milestone 8: Final Report (December 9)

Milestone 9: Project & Code Description (December 9)

11 RESULTS SO FAR

11.1 Correlations

We found that the correlation matrices fail to uncover interesting correlations between attributes, specifically that transaction value is not strongly correlated with any one feature. Our preliminary feature correlation matrix, shown in Figure 1, reveals intuitive correlations, such as `block_number` being strongly correlated with `transaction_month`. Additionally, there is a strong negative correlation between `transaction_month` and `transaction_day`, which suggests temporal clustering of certain types of transactions.

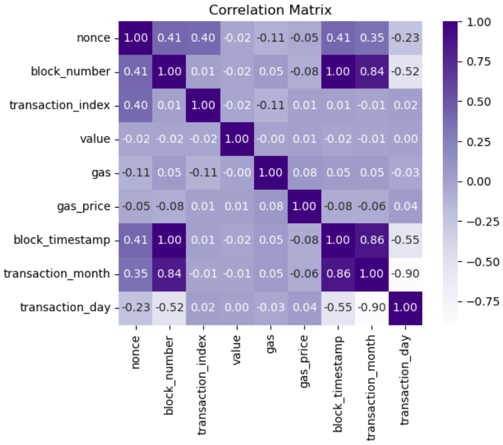

Figure 1: Correlation matrix of features showing intuitive correlations.

11.2 Gas distributions

Furthermore, tracking `gas` (the computational effort required to execute a transaction) is an interesting attribute to examine in correlation with transaction value. A high transaction value paired with low gas may reveal unique patterns, as could the inverse. In our initial EDA, we found no strong correlation between `gas` and `value`.

We created several distributions of gas-related values, examining both `gas usage` (the total computational effort required for a transaction) in Figure 2 and `gas price` (the amount a wallet is willing to pay to expedite transactions) in Figure 3.

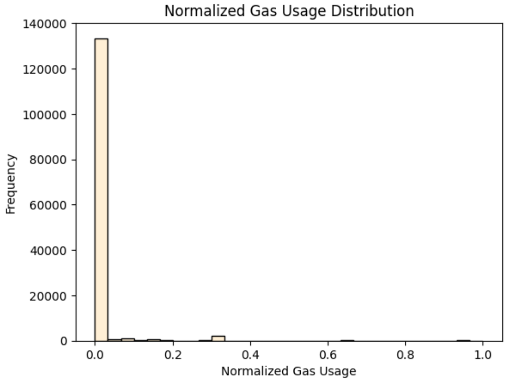

Figure 2: Distribution of gas usage, showing a high frequency in low gas prices. The outliers may reveal valuable insights.

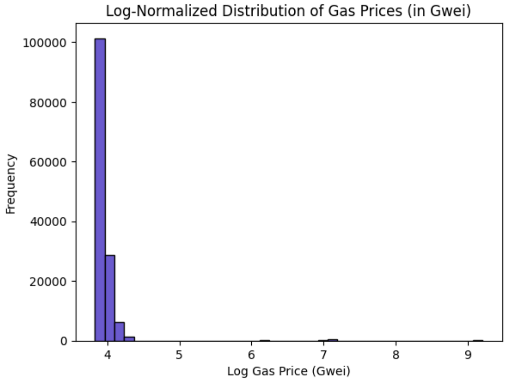

Figure 3: Distribution of gas price, showing a high frequency in low gas prices. Wallets paying significantly more for faster transactions could indicate interesting trends.

11.3 Timeplot

While not directly useful, a time plot of transactions over time in Figure 4 demonstrates how a brief period represents approximately $\approx 139,000$ transactions.

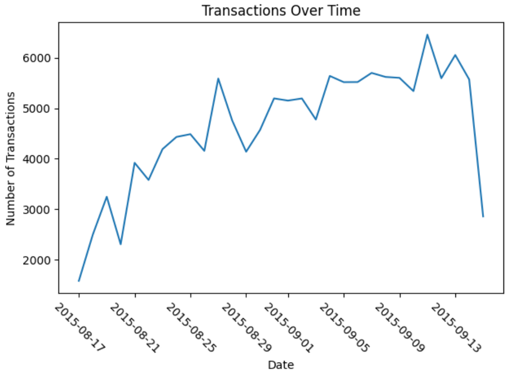

Figure 4: Daily transactions between August 17, 2015, and September 13, 2015.

11.4 Active wallets

Identifying the most active senders and receivers is highly informative. As mentioned earlier, fraud may occur within the margins (somewhere between the most and least frequent transactors). By identifying high-traffic vendors or entities, we can remove valid businesses from the dataset or magnify their holdings if they appear compromised. Figure 5 shows the most active senders, and Figure 6 shows the most active receivers.

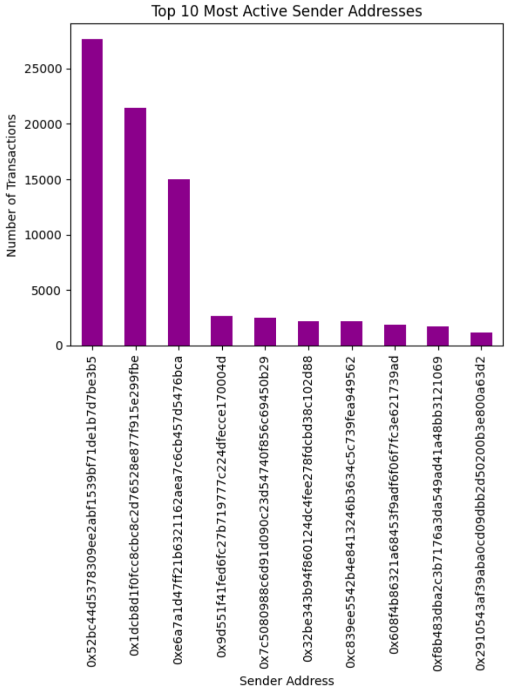

Figure 5: 10 most active sender wallets.

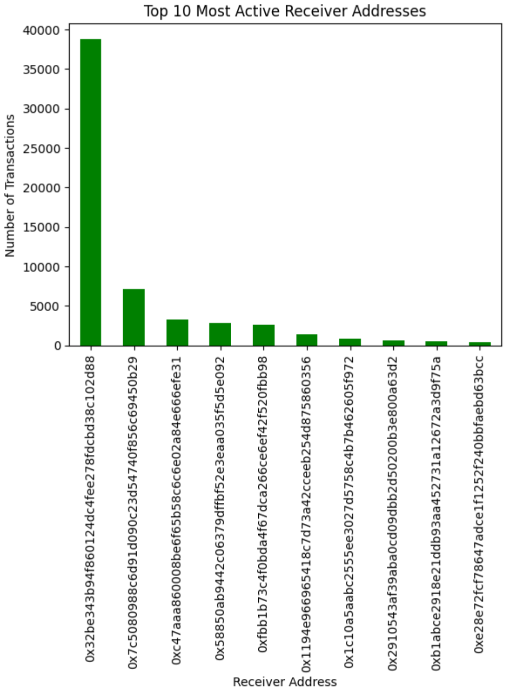

Figure 6: 10 most active receiver wallets.

11.5 Nonce distribution

We have also discovered the importance of the nonce attribute, which increments for each wallet with every transaction. We can use this to track wallets with a high degree of transactions, even if many of them occurred before the start of our observed window. This also allows us to identify wallets completing their first transaction. We believe both the maximum and minimum nonce values within a given window will reveal interesting trends.

Examining the nonce distribution reveals a high degree of first-time senders, as shown in Figure 7. This pattern may indicate a trend in creating "burner" wallets to avoid detection. Closer analysis of low-nonce transactions could provide more insight.

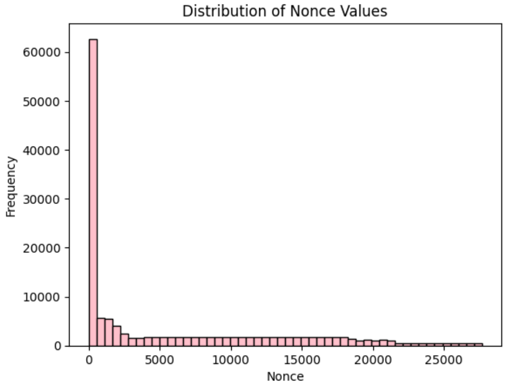

Figure 7: Distribution of nonce values, highlighting a high proportion of first-time senders.

REFERENCES

[1] Chenxiang Jin, Jiajun Zhou, Chenxuan Xie, Shanqing Yu, Qi Xuan, Xiaoniu Yang. 2024. Enhancing Ethereum Fraud Detection via Generative and Contrastive Self-supervision. Available at: https://arxiv.org/abs/2408.00641.

[2] Ethereum Improvement Proposals. 2019. EIP-1559: Fee market change for ETH 1.0 chain. Available at: https://eips.ethereum.org/EIPS/eip-1559.

[3] Ethereum Improvement Proposals. 2022. EIP-4844: Shard Blob Transactions. Available at: https://eips.ethereum.org/EIPS/eip-4844.

[4] Etherscan. 2024. Etherscan: The Ethereum Blockchain Explorer. https://etherscan.io/. Accessed: 2024-10-14.

[5] Federal Bureau of Investigation. 2023. FBI Cryptocurrency Report. https://www.fbi.gov/services/laboratory/cryptocurrency-report. Available at: https://www.fbi.gov/services/laboratory/cryptocurrency-report.

[6] Federal Bureau of Investigation. 2023. FBI Internet Crime Report 2023. Available at: https://www.fbi.gov/internet-crime-report.

[7] Google. 2024. Google Trends. https://trends.google.com/. Accessed: 2024-10-14.

[8] Homeland Defense & Security Information Analysis Center. 2023. Real-Time Cryptocurrencies Monitoring for Criminal Activity Detection: A Comprehensive System. https://www.hdiac.org/cryptocurrency-fraud-detection. Available at: https://www.hdiac.org/cryptocurrency-fraud-detection.