Toward anomaly detection in the blockchain using Numerical Association Rule Mining

Renata Kramberger
Department of Information Technology and Computing
Zagreb University of Applied Sciences
Zagreb, Croatia
https://orcid.org/0000-0002-5433-217X

Iztok Fister
Faculty of Electrical Engineering and Computer Science
University of Maribor
Maribor, Slovenia
https://orcid.org/0000-0002-9964-6957

Iztok Fister, Jr.
Faculty of Electrical Engineering and Computer Science
University of Maribor
Maribor, Slovenia
https://orcid.org/0000-0002-6418-1272

Aida Kamišalić
Faculty of Electrical Engineering and Computer Science
University of Maribor
Maribor, Slovenia
https://orcid.org/0000-0002-8574-8506

***Abstract*—With the global increase in the popularity of cryptocurrencies, the need for anomaly detection and fraudulent behavior is reaching an all-time high. In our paper, we propose a novel method of anomaly detection with the use of Numerical Association Rule Mining with Differential Evolution. The experiment was conducted by using the Dogecoin blockhain, and the dataset contained all of the transactions from one month. Our results contained 303 rules, with the best fitness function value of 0.8.**

***Index Terms*—Numerical Association Rule Mining, Differential Evolution, Blockchain, Dogecoin, Anomaly Detection**

I. Introduction

As a pioneer of cryptocurrencies, Bitcoin paved the way for the development and popularization of many blockchain based cryptocurrencies. The ability of secure transaction processing has increased the popularity of blockchain technologies greatly. Despite the popularization and increase in the number of transactions, the technology is still prone to various security, privacy, and reliability issues. Therefore, the need has also increased to detect anomalies and fraudulent behavior.

In their overview paper, Hassan et al. [1] gave a comprehensive classification of anomalous attacks that can be found on blockchain technology. The main attack categories included account based, transaction based, smart contract based, system based, and consensus based attacks. This shows that anomalies can be seen in multiple layers and parts of the blockchain.

Many studies are engaged in research into which algorithms provide the best results with anomaly detection. In their work, Sayadi et al. [2] analyzed electronic transactions from the Bitcoin blockchain. They used the One-Class SVM algorithm to perform anomaly detection and the K-means clustering algorithm to group similar attacks. The grouped anomalies were then labeled by type into the following categories: (1) Distributed Denial-of-Service DDoS attack, double spending attack, and (2) Other attacks. Among the other attacks, a Blockchain Anomaly Detection (BAD) solution, was indicated by Signorini et al. [3], where authors created a complete framework that uses blockchain metadata to detect potential malicious activity. An anti-money laundering anomaly detection system was proposed by Alarab et al. [4]. They used Graph Convolutional Networks to detect and predict illicit transactions within the Bitcoin blockchain. As can be seen, a wide range of algorithms is used to conduct anomaly detection in blockchains.

The purpose of our paper is to provide a novel method of anomaly detection of blockchain transactions by using Numerical Association Rule Mining (NARM). The method consists of five steps: preprocessing, feature extraction, attribute identification, NARM, and explanation of the results using XAI. Thus, the NARM was applied to a transaction database consisting of Dogecoin data, while those mined association rules were detected using visualization techniques that are suspicious for the anomaly detection.

The main contributions of the proposed approach are as follows:

- a new method based on NARM was developed capable of blockchain analysis,
- the method supplements the collection of the existing XAI techniques,
- the blockchain analysis can be used for anomaly detection.

The used methods and materials are described in Section II. Section III explains the data preparation, feature extraction, identification of attributes, and used methods. The experiments and results are depicted in Section IV, and a summary of the performed work is given in Section V, where directions are also outlined for the future work.

II. Materials and methods

In this section, the following topics are discussed:

- Association Rule Mining,

- Numerical Association Rule Mining,
- Blockchain basics.

In the remainder of the paper, all three subjects are illustrated in detail.

A. Association Rule Mining

The ARM problem is defined formally as follows: Let us suppose a set of objects $O = \{o_1, \ldots, o_m\}$ and transaction database D are given, where each transaction T is a subset of objects $T \subseteq O$. Thus, the variable m designates the number of objects. Then, an association rule can be defined as an implication:

$$X \Rightarrow Y, \tag{1}$$

where $X \subset O$, $Y \subset O$, in $X \cap Y = \emptyset$. The following two measures are normally defined for evaluating the quality of the association rule [5]:

$$conf(X \Rightarrow Y) = \frac{n(X \cup Y)}{n(X)}, \tag{2}$$

$$supp(X \Rightarrow Y) = \frac{n(X \cup Y)}{N}, \tag{3}$$

where $conf(X \Rightarrow Y) \geq C_{min}$ denotes the confidence and $supp(X \Rightarrow Y) \geq S_{min}$ the support of association rule $X \Rightarrow Y$. There, N in Eq. (3) represents the number of transactions in the transaction database D, and $n(.)$ is the number of repetitions of the particular rule $X \Rightarrow Y$ within D. Additionally, C_{min} denotes minimum confidence and S_{min} minimum support, determining that only those association rules with confidence and support higher than C_{min} and S_{min} are taken into consideration, respectively.

Usually, an additional measure for measuring the proportion between the number of attributes arising in the antecedent and consequent, and the total number of attributes within the transaction database is expressed mathematically as follows [6]:

$$incl(X \Rightarrow Y) = \frac{|ante(X \Rightarrow Y)| + |cons(X \Rightarrow Y)|}{m}, \tag{4}$$

where the functions $ante(X \Rightarrow Y)$ and $cons(X \Rightarrow Y)$ represent a set of items belonging to the antecedent and the consequent, respectively, and the variable m denotes the number of all the attributes within the transaction database. Thus, it is valid, the closer the measure to 1, the more attributes are included into the association rule $X \Rightarrow Y$.

B. Numerical Association Rule Mining

Numerical Association Rule Mining (NARM) extends the idea of ARM, and is intended for mining association rules where attributes in a transaction database are represented by numerical values. Usually, traditional algorithms, e.g. Apriori, require a discretization of numerical attributes before they are ready to use. The discretization is sometimes trivial, and sometimes does not have a positive influence on the results of the mining. On the other hand, many methods for ARM exist that do not require the discretization step before applying the process of mining.

Most of these methods are based on population-based nature-inspired metaheuristics, such as, for example, Differential Evolution or Particle Swarm Optimization. NARM has also recently been featured in some review papers [7], [8] which emphasize its importance in the data revolution era.

Each numerical attribute is determined by an interval of feasible values limited by their lower and upper bounds. The broader the interval, the more association rules are mined. The narrower the interval, the more specific relations are discovered between attributes. Introducing intervals of feasible values has almost two effects on the optimization: To change the existing discrete search space to continuous, and to adapt these continuous intervals to suit the problem of interest better.

Mined association rules can be evaluated according to several criteria, like support and confidence. However, these cover only one side of the coin. If we would also like to discover the other side, additional measures must be included into the evaluation function.

C. Blockchain basics

The development of Blockchain technology began when Satoshi Nakamoto introduced Bitcoin to the world. This digital transaction exchange framework became popular, and led to the development of a wide range of Blockchain based currencies. Blockchain operates as a peer-to-peer network that does not need any kind of central authority. The transactions are stored chronologically within blocks, and are connected to form a chain. These blocks are stored on all nodes within the blockchain network in the form of .dat files. One .dat file contains multiple blocks, whose data are hashed and written in bytes that are not in a human-readable format.

In each block, the first transaction is called a Coinbase transaction. It contains the fee (reward) given to the miner for creating the block. The other transactions contain information about payments made from one account to another. In some cases transactions contain multiple senders and multiple recipients. With such transactions it is not clear what amount was sent from what sender to what recipient. In case, the sender has more funds than intended for the recipient, all funds are taken from the sender, the recipient receives his amount, and the difference is recorded as a new payment to the sender [9].

Most blockchain cryptocurrencies are based on Bitcoin and, because of this, they share a similar data structure and principles of operation. Interestingly, all the other blockchains except the Bitcoin cryptocurrency are considered to be Altcoin. One of these custom blockchains is Dogecoin (i.e., Altcoin). The Dogecoin blockchain was developed based on the Litecoin and Bitcoin blockchains. Both Litecoin and Dogecoin were launched in December 2013 and their value has still increased considerably since then [13]. We chose Dogecoin in our experiment.

III. Proposed method for anomaly detection

The proposed method for anomaly detection in the blockchain consists of the following steps:

1) data preprocessing,

2) feature extraction,
3) identification of attributes,
4) Association Rule Mining using Differential Evolution, and
5) explanation of the results.

Data preprocessing is an essential step in the proposed method, which enables that the raw data are mapped into a suitable form using methods like data cleaning, data integration, data transformation and feature extraction. In the second step, the preprocessed data are separated into related groups (i.e., features), and, thus, dimensionality reduction is proven. In the third step, features' domains need to be defined, and a transaction database is created. The fourth step serves for searching for the relations between the attributes hidden in transaction database. The final step is devoted to an explanation of the results.

In the remainder of the paper, the mentioned steps are described in detail.

A. Data preprocessing

The complete cryptocurrency blockchain needs to be downloaded and preprocessed in the data preparation step. Data in blockchain technology consisting of transaction information and a block, are stored into files with the .dat extension. Thus, the transaction information is written in encrypted form, and, therefore, it is omitted in our study. On the other hand, the block data are the subject of the parsing process. Indeed, the results of parsing allow us to access the block data, from which features are extracted in the next step.

B. Feature extraction

The result of data preprocessing is an archive of the cryptocurrency blockchain transactions in raw form, where each transaction is represented as a quadruple:

$$T = \langle Address\_1, Address\_2, Amount, Timestamp \rangle, \quad (5)$$

where $Address\_1$ denotes a sender wallet address, $Address\_2$ a receiver wallet address, $Amount$ an amount of coins, and $Timestamp$ is the date/time of accomplishing the transaction. Those elements of the quadruple are identified as the features in our study. In the sense of an association rule, the transaction's quadruple can be interpreted as the following relation:

$$Address\_1 \wedge Timestamp \wedge Amount \Rightarrow Address\_2,$$

which means that at the date/time $Timestamp$, the wallet $Address\_1$ has transferred the $Amount$, number of coins, to wallet $Address\_2$.

In the feature extraction step, the quadruple as illustrated in Eq. (5) is constructed for each transaction. Additionally, some data transformation and filtering were applied in the step. For instance, when one recipient received coins from multiple senders, the exact amount was retrieved with the use of the previous transaction hash and previous transaction identification. The parsed data were processed further to exclude:

- transactions that did not take place in January,
- no Dogecoin transactions (e.g., Coinbase transactions),
- transactions that contained multiple input and output addresses,
- transactions that contained the return of unspent coins, and
- transactions that contained addresses that could not be parsed, or had other errors.

C. Identification of attributes

Cryptocurrency blockchain transactions consist of features that can be specified by two types of attributes: categorical and/or numerical. Actually, numerical attributes need to be mapped into categorical by discretization. In our case, the features $Address\_1$ and $Address\_2$ are discrete, the feature $Amount$ is the number, while the feature $Timestamp$ represents the date/time, and therefore demands a special treatment.

The purpose of this step is to identify the domains of the attributes and to create a corresponding transaction database. The transaction database is a matrix, with rows designated transactions and columns denoting attributes by features, where each element of the matrix is assigned to value one or zero. Indeed, the value one means that the corresponding attribute is included in a transaction. In contrast, the element with zero indicates that it is not part of the transaction.

The attributes are identified as follows: At first, sets of sender wallets, S_{Addr_1}, and receiver wallets, S_{Addr_2}, need to be conformed, consisting of discrete attributes. As a result, the former set contains all the unique wallets that have arisen as senders, while the latter all the unique receivers in the cryptocurrency blockchain transactions.

Then, the numerical attributes of the feature $Amount$ need to be put into the predefined discrete classes. Although this mapping can be performed in many ways, intervals of feasible values of attributes are divided into the number of classes in our study. Thus, the interval is defined by the minimum and maximum values of the attributes as found in the transaction database. Consequently, the size of each class $\Delta Attr^{(Num)}$ is calculated according to the following equation:

$$\Delta Attr^{(Num)} = \frac{Attr_{\max}^{(Num)} - Attr_{\min}^{(Num)}}{K}, \quad (6)$$

where $Attr_{\max}^{(Num)}$ and $Attr_{\min}^{(Num)}$ denote the maximum and minimum values of the numeric attribute, and K is the number of classes. Obviously, the numeric attributes belong to the k-th class if the following relation is fulfilled:

$$\begin{aligned} \Delta Attr^{(Num)} \cdot k + Attr_{\min}^{(Num)} &\leq Attr_k^{(Num)} \\ &< \Delta Attr^{(Num)} \cdot (k+1) + Attr_{\min}^{(Num)}, \end{aligned} \quad (7)$$

for $k = 0, \ldots, K$.

Finally, the $Timestamp$ value needs to be classified into corresponding time intervals. For instance, if transactions within one year are observed, the attributes can be created with regard to the month in which the transaction took place. In this way, $M = 12$ attributes are defined within the transaction database.

In summary, the number of attributes in the transaction database is determined as follows:

$$N_t = |S_{Addr\_2}| + K + M_tp + 1, \tag{8}$$

where the first term determines the number of unique receivers of specific cryptocurrency arising in the observed transactions, K is the number of numeric attributes determining the $Amount$ feature, M_tp determines the time periods, and one refers to the ordinal number $ord(S_{Add\_1_j})$ of the j-th sender involved in the i-th transaction.

D. Association Rule Mining using Differential Evolution

Differential Evolution (DE) [10] is selected for solving the Association Rule Mining (ARM), where each solution $\mathbf{x}_i$ represents a transaction in a database. Thus, the transaction consists of a 4-dimensional vector:

$$\mathbf{x}_i = (x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}), \quad \text{for } i = 1, \ldots, Np, \tag{9}$$

with elements $x_{i,j} \in [0, 1]$ for $j = 1, \ldots, 4$ and Np denotes the number of individuals in the population. Indeed, the elements represent encoded values of variables that are decoded by genotype-phenotype mapping [11] using the following rules: The sender wallet address is governed by the following mapping:

$$Address\_1 = \lfloor x_{i,1} \cdot |S_{Addr\_1}| \rfloor, \tag{10}$$

where the variable $Address\_1$ is calculated from the element $x_{i,1}$ multiplied by the size of the antecedent's set S_{Addr_1}. Similarly, the receiver wallet address $Address\_2$ considers the size of the consequent set S_{Addr_2}, in other words:

$$Address\_2 = \lfloor x_{i,2} \cdot |S_{Addr\_2}| \rfloor. \tag{11}$$

The variable $Amount$ is decoded from the second element of the individual $x_{i,2}$ according to the following mapping:

$$Amount = \lfloor x_{i,3} \cdot K \rfloor, \tag{12}$$

where K denotes the number of classes in which the specific payments are classified. Finally, the variable $Timestamp$ is determined with regard to the mapping, as follows:

$$Timestamp = \lfloor x_{i,4} \cdot T \rfloor + 1, \tag{13}$$

where the variable T designates the number of classes in which the observed classes are classified. When the classes are divided according to year basis, the variable is set as $T = 12$, and each class refers to the appropriate month.

The fitness function of the DE algorithm is defined as a linear combination of ARM metrics, in other words:

$$f(\mathbf{x}_i^{(t)}) = \frac{\alpha \cdot supp(X \Rightarrow Y) + \beta \cdot conf(X \Rightarrow Y) + \gamma \cdot incl(X \Rightarrow Y)}{\alpha + \beta + \gamma}, \tag{14}$$

where $supp(.)$, $conf(.)$, and $incl(.)$ denote support, confidence, and inclusion ARM metrics, while parameters α, β, and γ are weights introducing biases.

E. Explanation of the results

Decisions of AI models have a crucial impact on the human life. Today, AI models operate as a "black boxes" whose results are hard to interpret. Consequently, users do not comprehend their results, and they even do not trust them. On the other hand, the "black box" models are created directly from data. Therefore, it is difficult for the developers to explain what exactly happened to the AI algorithms and how they arrived at the results. As a result, the explainable AI (XAI) has emerged, which helps developers ensure that the AI system is working as expected [12].

One of the more important ways of XAI how to explain the AI models: some AI models are interpretable by design (i.e., transparent models), while the others need external XAI techniques for interpreting (i.e., post-hoc explainability). In our study, visualization was considered for a post-hoc explainable technique applied for anomaly detection in the blockchain.

Indeed, a set of association rules emerged after applying the ARM algorithm on the preprocessed transaction database. The knowledge from the set is hard to interpret due to their big number. In line with this, visualization of the more important association rules was used, in order to see which partners are involved in blockchain transactions more often, and what amount they operate with. The visualization can enable us to detect any anomaly in payment transactions between two partners easily.

IV. Experiment and results

The main goal of our experiment was to show that the proposed method can be applied for ARM in mining the transaction database created from the Dogecoin cryptocurrency blockchain. Furthermore, the selected mined association rules can be used for anomaly detection using the XAI post-hoc processing (i.e., visualization).

Let us emphasize that all five steps of the proposed method were taken into consideration during the experimental work. As a base for our experiment we selected the blockchain data from January 2021.

The dataset was processed with the help of ARM using DE. The tool used for this was the uARMSolver [6], a framework that provides the processing and visualization of data with the help of ARM. The parameters for the ARM used in this experiment included 10,000 function evaluations and a population size of 100.

The selected set of mined association rules are visualized in the last step. To be able to visualize the results we pruned the rules further, to extract only those rules that contained sender or receiver addresses. The parallel coordinates plot was used for the visualization.

A. The results

The obtained results contained the mined association rules with the best fitness function of 0.8 over 100 generations of the Differential Evolution ARM algorithm. This algorithm produced 303 association rules.

Figure 1 depicts the relationship between support and confidence given by the results of Association Rule Mining using Differential Evolution.

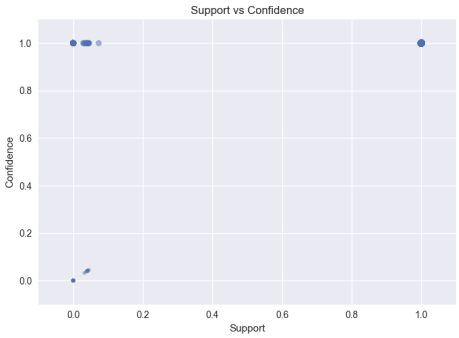

Fig. 1. Support vs Confidence.

The diagram shows a high percentage of rules that resulted in a high rate of both confidence and support. Actually, three situations can be distinguished from Fig. 1:

1) high support and high confidence (upper-right corner),
2) low support and high confidence (upper-left corner),
3) low support and low confidence (bottom-left corner).

The higher support denotes those in some way degenerate association rules that consist only of day and amount attributes (i.e., without any addresses). These rules typically indicate amounts spent on certain days. Considering that the aforementioned rules represent a specific pattern of behavior that deviates from the transaction norm, it can be assumed that this is an anomaly.

Although the rules with a low rate of support but a a high rate of confidence rate are rarely mined, they have a high chance of being predicted. This can be an indicator of patterned behavior between two or more participants within the network. Patterned behavior and co-occurring patterns can be an indicator of anomalous behavior within the cryptocurrency blockchain network. This shows that the algorithm has a strong confidence in predicting the likelihood of the rules appearing.

The last situation captures the association rules with low support and low confidence. These rules signal the normal trade in a blockchain network, where each person can enter into a commercial transaction as either a customer or salesman with different partners. Although also, here, a deviant behavior could be indicated, these situations are not treated as anomalies in our study.

Figure 2 depicts the relationship between the Right Hand Side (RHS), and Left Hand Side (LHS) of the mined rules and their associated support and confidence. The line colors depict the fitness values given to each mined rule. The values connected with the color can be seen in the color chart located next to the plot.

The diagram shows relationships between the sender's address, the recipient's address, the amounts sent, and the time when the transactions were conducted. All the rules shown in the diagram have high confidence, and thus could serve as a means of predicting which sums could be performed. Our method shows a set of rules showing predictable partial or full transactions which could predict repetitive actions being performed on the blockchain.

Also, it is possible to see that certain addresses participate in a large number of transactions, and that they are also included in the predicted behavior patterns. The fact that one user (address) is involved in such a large number of transactions that his address enters a pattern of behavior may indicate that he is part of anomalous behavior that may possibly be malicious.

B. Discussion

The results of the preliminary study showed that the NARM could be a valuable tool to detect anomalies in terms of transactions by giving high confidence predictions. Indeed, this does not mean that all rules with high confidence are anomalous, but only points out that these rules could be potentially anomalous, and, therefore, they demand a special treatment. Although also rules with higher support could be very valuable, especially in the case, when the enormous amount of Dogecoins are traded, the additional analysis is needed of this situation. However, this could be a promising direction for the future work.

Patterns of repetitive transactions can also be found by using this method, as well as identifying common and repetitive patterns in transaction behavior. This can then lead to a better understanding of how the flow of transactions is being conducted on a certain blockchain network. Knowing the usual behavior of users can also lead to easier detection of anomalies, by comparing new transactions with an already known pattern base.

Given that the experiment derived over 300 rules, (with a large set of addresses and amounts) visualization has shown to be an issue. Although heatmaps are often used to visualize ARM rules, the parallel coordinates plot seemed to be a better method of visualization for this specific problem.

V. Conclusion

The need for anomaly detection is also growing with the growth in the number of transactions on the Blockchain network. In our work, we used NARM to conduct an analysis of the Dogecoin blockchain transactions, in order to detect patterns and reoccurring behavior which can lead to the detection of anomalies. The anomalies can be identified by either comparing new transactions to the discovered patterns, or by detecting unusual transaction volumes and frequencies. Although not all anomalous transactions can be labeled as malicious, this also provides a step in the right direction towards identifying malicious behavior.

The presented experimental work was performed in a period of only one month. Therefore, our future work would involve a longer period of time, to see if it is possible to derive more rules with an even better confidence rate. An additional analysis of association rules with higher support could be discovered in the future.

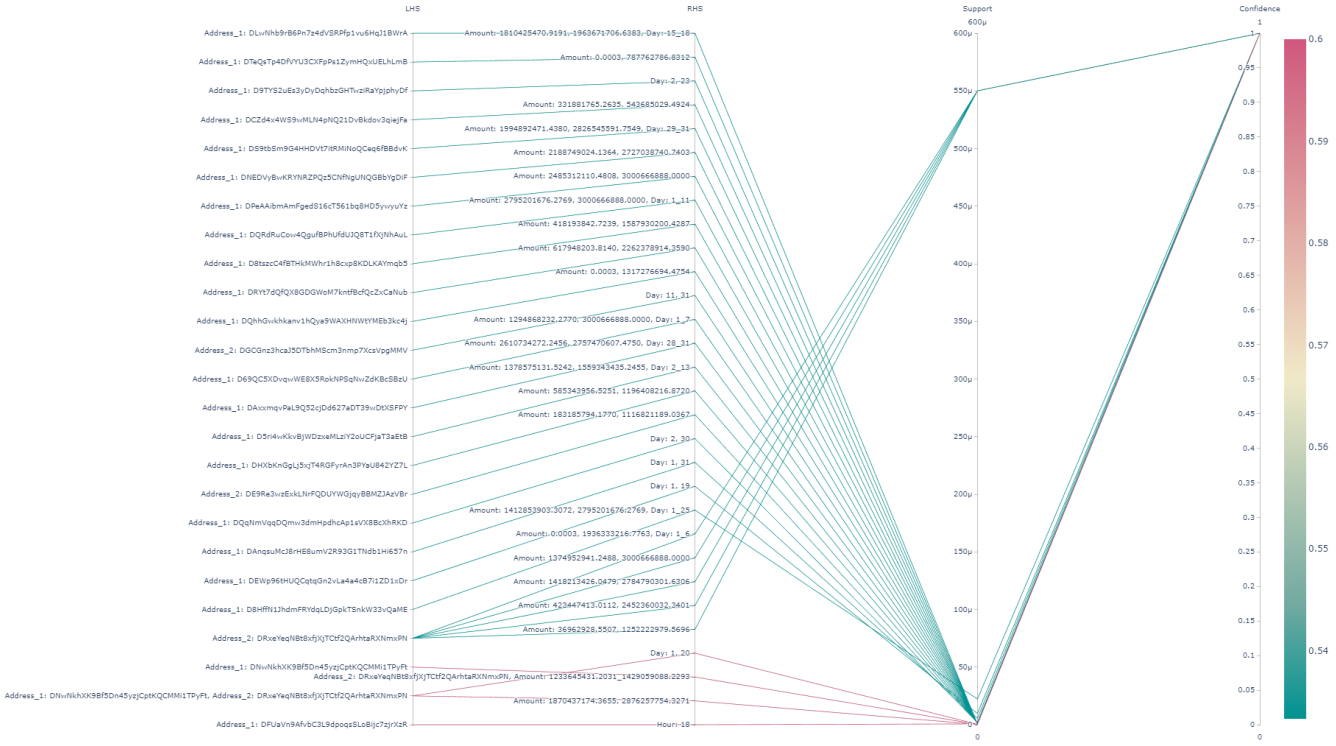

Fig. 2. Visual representation of the top 30 mined rules.

REFERENCES

[1] Hassan, M., Rehmani, M. & Chen, J. Anomaly Detection in Blockchain Networks: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*. pp. 1-1 (2022)

[2] SAYADI, S., BEN REJEB, S. & CHOUKAIR, Z. Anomaly Detection Model Over Blockchain Electronic Transactions. *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. pp. 895-900 (2019)

[3] Signorini, M., Pontecorvi, M., Kanoun, W. & Di Pietro, R. BAD: A Blockchain Anomaly Detection Solution. *IEEE Access*. **8** pp. 173481-173490 (2020)

[4] Alarab, I., Prakoonwit, S. & Nacer, M. Competence of Graph Convolutional Networks for Anti-Money Laundering in Bitcoin Blockchain. *Proceedings Of The 2020 5th International Conference On Machine Learning Technologies*. pp. 23-27 (2020), https://doi.org/10.1145/3409073.3409080

[5] Agrawal, R., Srikant, R. & Others Fast algorithms for mining association rules. *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*. **1215** pp. 487-499 (1994)

[6] Fister, Iztok and Fister, Iztok Jr (2020), uARMSolver: A framework for Association Rule Mining, arXiv, [online], https://github.com/firefly-cpp/uARMSolver (Accessed June 10, 2023)

[7] Altay, E. & Alatas, B. Performance analysis of multi-objective artificial intelligence optimization algorithms in numerical association rule mining. *Journal Of Ambient Intelligence And Humanized Computing*. pp. 1-21 (2019)

[8] Telikani, A., Gandomi, A. & Shahbahrami, A. A survey of evolutionary computation for association rule mining. *Information Sciences*. (2020)

[9] Nakamoto, S. Bitcoin: a peer-to-peer electronic cash system. (2008), https://bitcoin.org/bitcoin.pdf

[10] Storn, R. & Price, K. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal Of Global Optimization*. **11**, 341-359 (1997)

[11] Eiben, A. & Smith, J. Introduction to Evolutionary Computing, Second Edition. (Springer,2015), https://doi.org/10.1007/978-3-662-44874-8

[12] Phillips, P., Hahn, C., Fontana, P., Yates, A., Greene, K., Broniatowski, D. and Przybocki, M. (2021), Four Principles of Explainable Artificial Intelligence, NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.IR.8312, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=933399 (Accessed June 10, 2023)

[13] Young, I. Dogecoin: A Brief Overview & Survey. *SSRN ELibrary*. pp. 1-19 (2018)