Illicit Activity Detection in Bitcoin Transactions using Timeseries Analysis

Rohan Maheshwari[1], Sriram Praveen V A[2], Shobha G[3], Jyoti Shetty[4], Arjuna Chala[5], Hugo Watanuki[6]
Computer Science and Engineering Department, R. V. College of Engineering, Bengaluru, India[1, 2, 3, 4]
HPCC Systems LexisNexis Risk Solutions, Alpharetta, USA[5, 6]

***Abstract*—A key motivator for the usage of cryptocurrency such as bitcoin in illicit activity is the degree of anonymity provided by the alphanumeric addresses used in transactions. This however does not mean that anonymity is built into the system as the transactions being made are still subject to the human element. Additionally, there is around 400 Gigabytes of raw data available in the bitcoin blockchain, making it a big data problem. HPCC Systems is used in this research, which is a data intensive, open source, big data platform. This paper attempts to use timing data produced by taking the time intervals between consecutive transactions performed by an address and make an identification of the nature of the address (illegal or legal). With the use of three different goodness of fit run tests namely Kolmogorov–Smirnov test, Anderson-Darling test and Cramér–von Mises criterion, two addresses are compared to find if they are from the same source. The BABD-13 dataset was used as a source of illegal addresses, which provided both references and test data points. The research shows that time-series data can be used to represent transactional behaviour of a user and the algorithm proposed is able to identify different addresses originating from the same user or users engaging in similar activity.**

Keywords*—Bitcoin; time-series analysis; HPCC systems; random time interval; illicit activity detection

I. Introduction

The rapid growth of Cryptocurrencies has proved to be a major regulatory challenge. They are used in various illegal activities including illegal trade of drugs, hacks, thefts, illicit pornography and other major crimes. According to the 2022 Crypto Crime report by Chainalysis [1], a total of $14 billion was involved in Cryptocurrency based illicit activities in the year 2021. Although it represents only 0.15% of the total volume of crypto transactions, digital currencies and trading of these assets is becoming increasingly mainstream. The tracked volume of illicit activity is likely to rise in the future as more bad actors are identified. Bitcoin is the first and the most established cryptocurrencies in the world. Bitcoin provides pseudo-anonymity in the form of a 26-35 length alphanumeric addresses. This makes the identification of users difficult. However, it is possible to link transactions to users due to the public nature of the Bitcoin blockchain. After linking the addresses to users, it is further possible to identify which of these users were involved in activities that are of criminal nature [2].

Consider a user who has a dedicated set of Bitcoin addresses which they use to perform some kind of illegal activity. Over time, as they perform their transactions, the timestamps are available publicly and can be used to reveal the identity of the user. Behavioral biometrics rely usually on a rich stream of information to identify a user. RTI is one such biometric which can be considered as any sequence of time intervals between successive events. As we have the timestamps from the bitcoin blockchain we can obtain the RTI of transactions by taking the difference between the timestamp of the current transaction and the previous transaction for the sequence of transactions made by the user. This work uses the random time-interval (RTI) biometric [3].

This paper presents a method to recognise individual users from their behavioral metrics which is the time-series data and is publicly available on the bitcoin blockchain. A review of the related work is presented on the next section. In Section III, we present the methodology used which encompasses the methods and components involved in the research. Finally, the results and discussion of the three goodness-of-fit tests are presented in Section IV followed by conclusion in Section V.

II. Related Work

The most important datapoint that is generated by a bitcoin address for this technique of fraud detection is the RTI or Random Time Interval data which was first made use of in the paper by Laskaris, Zafeiriou, and Garefa [4]. Subjects were given a button to press randomly and it was shown that the time series produced could be used as a biometric signature of a person's cognitive thought process. Monaco [3] made use of 12 total time signatures unique to any address and used Takens's theorem for phase space reconstruction followed by the approximate multivariate wald-wolfowitz test to check whether the two samples originated from the same sources. The best identification rate was 76.

Other types of biometric identification have been attempted such as signature verification using template matching [5]. In this a novel variation of the DTW algorithm is used which produced verification errors of as low as 1.34% at a very rapid speed. Another application of timing data for biometric identification is in gait analysis. Mekruksavanich and Jitpattanakul [6] were able to use the timing data of 22 subjects from portable devices and create a CNN model which achieved an identification accuracy of as high as 93.9%.

Dynamic Time Warping (DTW) [7] is used to align two different time series data to recognize the origin of data. The only fundamental difference between various time series analysis and bitcoin time series analysis is that the measurement is event driven versus being measured

continuously by sensors. Often transactions also occur over various weeks or even months whereas with a specific application such as gait analysis data can be continuously captured over a period of day or two. Another difference is the accuracy of the time stamps being used, with sensor data the data is quite precise, but since transaction level timestamps are not available in the blockchain, block level time stamps are used which reduces accuracy.

Many techniques have been developed for bitcoin fraud detection apart from the usage of timestamp data such as clustering [8], various techniques such as trimmed k-means, DBSCAN etc. are tested to this end. Various network analysis techniques have also been analyzed [9] such as use of LOF (local outlier factor). The difference between these and time series analysis is that network analysis attempts to find anomalies in behavior globally whereas time series attempts to show that two addresses originate from the same user with no mention of the nature of the transactions being made.

III. Methodology

Bitcoin is the oldest and most widely adopted cryptocurrency to date. It becomes the natural choice for study given the amount of transactional data available publicly. Having 389 Gigabytes of raw data to work with immediately makes this a big data problem as bare metal resources cannot handle this much data. Hence, HPCC Systems is used. Another reason to choose HPCC is its ROXIE delivery engine which let us query new addresses quickly.

A. Bitcoin

On October 28th, 2008 Satoshi Nakamoto made public his research on a trustless peer to peer electronic cash system called bitcoin [10]. The system has its origins in the growing distrust around centralized financial institutions [11]. It relies on the proof of work model where the longest chain is the one that is trusted first amongst all the miners. Bitcoin also makes its transaction history easily accessible to the public making it easy to perform analysis.

The two most relevant data points on the chain to this particular research effort are the bitcoin addresses and bitcoin transactions. Every bitcoin user has access to a private key and a public key. The bitcoin address is derived from the public key by the use of one-way cryptographic hashing [12]. It represents the origin and destination of bitcoin in transactions. RTI data will be aggregated

Bitcoin transactions represent a transfer of bitcoin from one address to another. The transactions are then added to a mempool and miners race to add it to their blocks and obtain the block reward. The transactions are contained in blocks which are chained together.

B. HPCC Platform

As of April 2022, the amount of raw blk data a bitcoin client would download is 389 Gigabytes and if the solution is expanded to other crypto such as ethereum it can take as much as 658 Gigabytes of data. This makes the problem of fraud detection a big data problem where regular computational resources will struggle to keep up with the demands of scaling data. HPCC (High-Performance Computing Cluster) is a big data platform developed by LexisNexis Risk Solutions. It supports both parallelized batch computing and online querying using a declarative and data centric programming language called ECL (Enterprise Control Language) [13].

The HPCC platform has three main components as shown in Fig. 1, namely THOR, Roxie and ECL. The THOR cluster is used for batch processing, while ROXIE is used to run multiple online queries. ECL is the language used to interact with these two clusters. There is a plethora of other components used for housekeeping and maintenance of logical files and work units such as Dali, Sasha etc. The clusters themselves can be made up of commodity hardware to supercomputers as nodes. Thus, HPCC provides for scalability.

C. Goodness-of-Fit Tests

These statistical tests check whether a given set of observations were drawn from the same distribution. Generally, these tests check whether the observations were drawn from the normal distribution. For our purposes, a two-sample variation of the tests is used, to compare the underlying continuous distributions of two sets of independent observations.

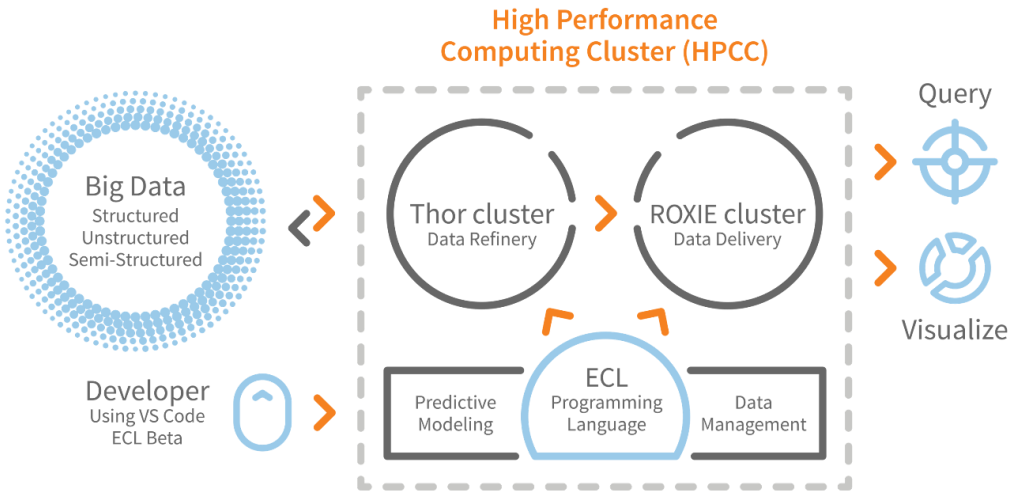

Fig. 1. HPCC systems architecture.

The three tests chosen show a varying degree of bias in different situations and have varying statistical power measures. Various studies have been conducted which include a series of goodness of fit tests and compare its statistical powers on different distributions. The Kolmogorov-Smirnov test has more statistical power against the distributions with n<100. The Anderson Darling test shows a better power against the distributions with higher sample sizes. The Cramer-von-Mises test performs similar to the KS test but for certain distributions with n>50, this test outperforms the Kolmogorov-Smirnov test [14].

1) Kolmogorov-smirnov test: A variation of the one sample test is used which compares a set of two observations [15]. It creates a cumulative distribution for both of the sets of observations after sorting them. The difference between the two distributions is obtained and the maximum of these differences is compared against the critical value. Should the Kolmogorov test statistic be less than critical value, the null hypothesis that both the observations originate from the same distribution will be rejected. In this paper the scipy implementation of this test is used [16]. We use the 'two-sided' option for the alternative parameter which states that the alternative hypothesis is that two distributions are not identical, F(x) is not equal to G(x) for all x; and the statistic is given by the maximum absolute difference between the empirical distribution functions of the samples. The method parameter is set to 'auto' option which means that for small arrays, it uses the exact distribution of test statistic; and for large arrays, it uses asymptotic distribution of test statistic.

2) Anderson-darling test: The Anderson Darling test is another test to check for data coming from a particular distribution. K-sample Anderson-Darling test is a modification that tests the null hypothesis that k-samples are drawn from the same population without any specification of the distribution function of that population [17]. The critical values depend on the number of samples. In this paper the scipy implementation of the k-sample Anderson-Darling test is used, taking the value of k as 2 [18]. The *midrank* parameter is set to *True* which computes the test using the midrank empirical distribution function applicable to continuous and discrete data.

3) Cramér–von mises criterion: Cramér–von Mises is another goodness of fit test. The two-sample Cramér-von Mises test is a test where the null hypothesis is that the samples are from the same, unspecified continuous distribution [19]. The scipy implementation of the Cramér-von Mises test is used in this paper [20]. The only parameter is *method* which is set to '*auto'* option working similar to the parameter in the Kolmogorov-Smirnov test.

D. Dataset

The Bitcoin Address Behavior Dataset [21] contains 13 categories of bitcoin addresses, each containing a list of addresses from a different type of crime. The reference illegal addresses have been picked from these. Specifically, to make the task of obtaining RTI simpler, 1500 addresses were randomly picked from labels 0, 10 and 11 and their RTI data was obtained by using a simple python script using the JSON-RPC interface to fetch all the information about transactions made by that address. The BABD-13 is used as it is a robust dataset which addresses not only the crime that the address was involved in, but also provides the degree of certainty that the illegal activity has occurred. Not only does this allow for the selection of illegal activities of interest, but it also enables the selection of only those addresses that are labeled as illegal with a high degree of certainty.

The other dataset in use is the one generated by the parser described later on, it has been obtained by parsing the binary data to CSV. There are many more data points that can be extracted from the raw blk data such as Merkle root, block hash etc. but have been ignored as the crucial data point for the research is the timing data.

E. Methods

The main idea was to retrieve all the bitcoin raw data using bitcoin core followed by the usage of a parser to retain the data in CSV format. This is followed by the generation of Random Time Interval data. This data is also generated for the addresses extracted from the BABD-13 dataset which will be part of both reference and test dataset. Each sample in the test dataset is then run using all three goodness of fit tests against each of the reference addresses. The statistic values are then averaged and if the average is greater than 0.5, the test returns a positive hit for suspicious activity from the tested address. The workflow is shown in Fig. 2.

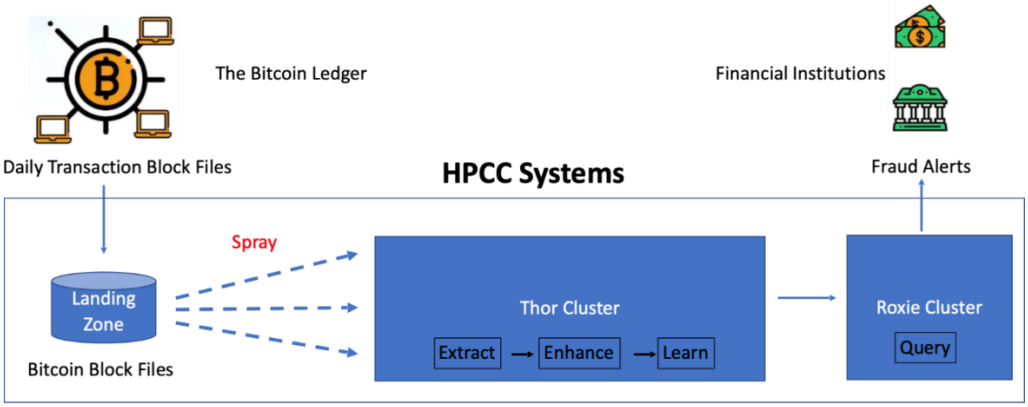

Fig. 2. Algorithm methodology.

The bitcoin data is obtained by initializing a bitcoin core node which downloads the data in the form of blk.dat files which are then sprayed onto the thor cluster for processing after passing through the landing zone. Here it goes through various ETL operations after which financial institutions will be able to submit queries of particular addresses to check for fraud on Roxie.

1) Parser: A bitcoin core node was set up to fetch the bitcoin blockchain raw data. The blockchain has been broken down into multiple blk.dat files (3185 as of writing this paper). The blk files consist of multiple blocks and are limited to 128 MiB. This data is all in binary form and a parser must be used to convert to CSV, JSON or some other structured format.

A modified Bitcoin parser [22] written in python is implemented using the HPCC Systems platform. The parser is embedded into ECL to take advantage of the parallel architecture available on the platform. This embedded python parser showed a significant improvement over a single node parser. For a set of 50 random blk.dat files, the HPCC parser ran at 5% of the time of a single node parser for 41 minutes.

This first phase of the parser extracted the following data points

a) Transaction Hash
b) Input Index
c) Input Transaction Hash
d) Output Index
e) Output Address
f) Output Value
g) Timestamp

The only missing data here is the input address, this data is obtained by using ECL. The blockchain stores the input transaction hash and the corresponding output index to refer to the inputs of a transaction. Thus, to find the actual input address ECL's self-join operation is used. Here the transaction in question and the previous transaction are joined, the corresponding output address of the previous transaction is the input address of the current transaction. This is shown in Fig. 3. This process leads to all the data being collected as required by the various algorithms used in this paper.

2) RTI Generation: Random Time Interval (RTI) data [4] can be generated from the timestamp data obtained from the blockchain. The data is generally precise and captured by sensors in cases such as gait analysis [7], this is where bitcoin presents a slight hurdle. Transaction level timestamps are not recorded and the closest substitute is the block level timestamp. Even with this rougher granularity however, some promising results do arise as will be seen later on.

The RTI data is calculated by taking consecutive timestamps in UNIX time of transactions made by a given address and subtracting them from their successor. Thus, if an address has made n transactions it will have an RTI of length n-1. RTI generation was done by making use of ECL's ITERATE function.

3) Algorithm: A set of reference addresses are taken which can contain legal and illegal addresses. A test query is then taken and one of the three aforementioned run tests is used to evaluate the test address against each and every reference address. If the reference address is illegal and returns a p value greater than the threshold or if the reference address is legal and the test returns a p value less than the threshold it is added to the total number of hits. Finally, if the fraction of hits is greater than a threshold value, the query will return as illegal.

This threshold on the fraction can be thought of as a level of risk tolerance, where a lower threshold means that financial institutions would want to investigate addresses even with a small fraction of hits and vice versa.

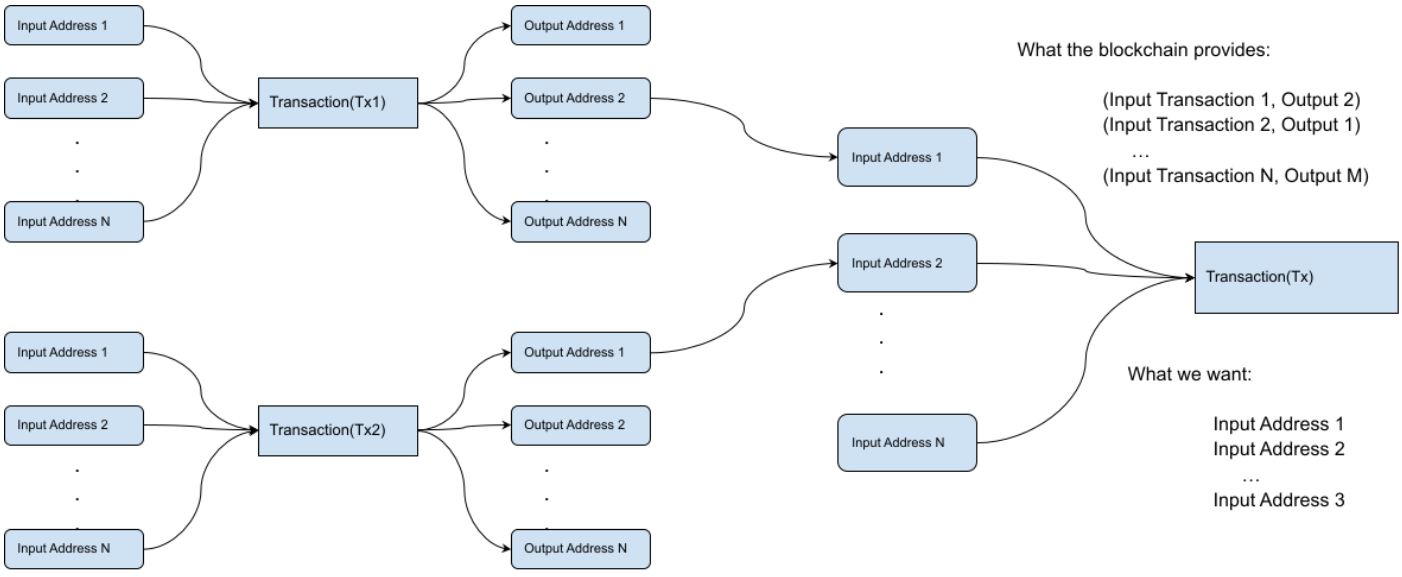

Fig. 3. Input to output address relation.

These illegal reference addresses are taken from the Bitcoin Address Behavior Dataset (BABD) [21] and the legal addresses are addresses which have been randomly taken with the assumption that 1% of bitcoin addresses are illegal. There is very high variability in the consideration of percentage of illegal activity in bitcoin [2][23]. The implementation of this querying will eventually be done on ROXIE for financial institutions to use.

IV. Results and Discussion

Before there are three run tests that have been used to evaluate distributional similarities between the RTI data of two addresses. They are:

- Kolmogorov–Smirnov test
- Anderson–Darling test
- Cramér–von Mises criterion

Based on the composition of the reference and test query dataset, three evaluations were made,

- Illegal only test vs illegal only reference
- Illegal only test vs illegal and legal reference
- Illegal and legal test vs illegal only reference

The results obtained were evaluated based on precision, accuracy, f1-score and recall and only those addresses with at least 25 transactions were considered in both reference and query sets.

A. Method 1

In this method the test dataset consisted of only known illegal addresses and the references also consisted of only known illegal addresses. The size of the reference set was 263 and the size of the query set was 66. Table I summarizes these findings.

TABLE I. Illegal Only Test Vs Illegal Only Reference

Algorithms	Precision	Recall	f1-score	Accuracy
Kolmogorov–Smirnov test	1.00	0.85	0.92	0.85
Anderson–Darling test	1.00	0.77	0.87	0.77
Cramér–von Mises criterion	1.00	0.80	0.89	0.80

B. Method 2

In this method the test dataset consisted of only known illegal addresses but the references consisted of both known illegal addresses and random addresses which were considered legal. The size of the reference set was 304 and the size of the query set was 66. Table II summarizes these findings.

TABLE II. Illegal Only Test Vs Illegal and Legal References

Algorithms	Precision	Recall	f1-score	Accuracy
Kolmogorov–Smirnov test	1.00	0.85	0.92	0.85
Anderson–Darling test	1.00	0.79	0.88	0.79
Cramér–von Mises criterion	1.00	0.80	0.89	0.80

C. Method 3

In this method the test dataset consisted of known illegal addresses and legal addressee and the references consisted of only known illegal addresses. The size of the reference set was 263 and the size of the query set was 107. Table III summarizes these findings.

The following observations can be made from the above data. The Kolmogorov–Smirnov test consistently across all three methods either outperforms or is at par with the other two run tests in terms of accuracy. However, the f1-score is lower in general.

Given the f1-score is lower a further look into the recall and precision of the methods is called for. In the use case of detecting illegal addresses for financial institutions with virtually unlimited resources, it is possible to look into false positives and so a higher recall would be preferable. Here as well, the Kolmogorov–Smirnov test gives the highest recall except in the detection of legal addresses in method 3 where the Anderson –Darling test does better with 33%.

From method 3 we see a sharp contrast in detection between legal and illegal addresses. The f1 is consistently over double for all three run tests. This is due to the initial assumption that only 1% of addresses being involved in illegal activity is wrong and random selection of addresses has led to addresses that are not legal to be included making the query and test dataset impure. It is also because the activity that legal addresses are involved in and thereby the time series patterns generated vary too widely for a small sampling to represent them. This can cause legal addresses to be misclassified as illegal, leading to lower overall detection rates for legal addresses.

TABLE III. Illegal and Legal Test, Illegal Only Reference

Algorithms	Kolmogorov–Smirnov test					Anderson–Darling test					Cramér–von Mises criterion				
	class		*accuracy*	*macro avg*	*weighted avg*	*class*		*accuracy*	*macro avg*	*weighted avg*	*class*		*accuracy*	*macro avg*	*weighted avg*
	0	*1*				*0*	*1*				*0*	*1*			
Precision	0.47	0.64		0.56	0.57	0.46	0.65		0.55	0.58	0.41	0.62		0.52	0.54
Recall	0.22	0.85		0.53	0.61	0.32	0.77		0.54	0.6	0.22	0.8		0.51	0.58
f1-score	0.3	0.73	0.61	0.51	0.56	0.38	0.7	0.62	0.54	0.58	0.29	0.7	0.58	0.49	0.54
Support	41	66	107	107	107	41	66	107	107	107	41	66	107	107	107

The number of addresses with more than 25 transactions is a smaller fraction of the number of data points collected. This also points to other transactional behavior where bitcoin addresses are not reused as much or that bitcoin is often unused and parked in addresses. The limited number of addresses with more than 25 transactions suggests that the behavior of bitcoin addresses may vary widely and may not be accurately represented in the dataset.

The lack of data on illegal transactions and addresses limits the certainty of identifying illegal addresses. Therefore, the accuracy of the results may be affected, and false positives may occur.

V. Conclusion

The work supports the original hypothesis that bitcoin transaction behavior is nonrandom. The paper presents strong evidence that users engaging in illegal activity can be detected by use of previously known illegal addresses.

The simple use of timing data has proven to be quite effective. Paired with external information about the owner of an address or even network information available on the blockchain such as the amount being sent, the effectiveness could go higher.

One problem with any bitcoin illegal detection approach is the lack of data on illegal transactions and addresses. The dataset used in this paper does provide illegal addresses but only with a limited degree of certainty. Legal addresses are also seen to have a lower overall detection rate. This is likely due to the assumption of a percent of addresses being illegal being off while picking random addresses. While the Kolmogorov-Smirnov test may outperform the other run tests in terms of accuracy, the limitations of the dataset and assumptions made in the paper must be considered when interpreting the results. Further research and more comprehensive datasets are needed to improve the accuracy and reliability of illegal address detection in bitcoin transactions.

The method is only dependent on one feature which is the time interval data being generated, and since this is not a feature specific to bitcoin it can very easily be extended to other cryptocurrencies such as Ethereum, Luna etc. Similarly, it can be extended to other applications such as credit card fraud detection which have the added advantage of having precise transaction timestamp information and easy access to a person's history.

References

[1] Grauer, K., Updegrave, H. and Kueshner, W. (2022) The chainalysis 2022 crypto crime report, Chainalysis. Retrieved from https://go.chainalysis.com/2022-Crypto-Crime-Report.html (Accessed: December 2, 2022).

[2] Foley, S., Karlsen, J. R., & Putniņs, T. J. (2019). Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? The Review of Financial Studies, 32(5), 1798-1853.

[3] Monaco, J. V. (2015, May). Identifying bitcoin users by transaction behavior. In Biometric and surveillance technology for human and activity identification XII (Vol. 9457, pp. 25-39). SPIE.

[4] Laskaris, N. A., Zafeiriou, S. P., & Garefa, L. (2009). Use of random time-intervals (RTIs) generation for biometric verification. Pattern Recognition, 42(11), 2787-2796.

[5] Okawa, M. (2019). Template matching using time-series averaging and DTW with dependent warping for online signature verification. IEEE Access, 7, 81010-81019.

[6] Mekruksavanich, S., & Jitpattanakul, A. (2021). Convolutional neural network and data augmentation for behavioral-based biometric user identification. In ICT Systems and Sustainability: Proceedings of ICT4SD 2020, Volume 1 (pp. 753-761). Springer Singapore.

[7] Permanasari, Y., Harahap, E. H., & Ali, E. P. (2019, November). Speech recognition using dynamic time warping (DTW). In Journal of physics: Conference series (Vol. 1366, No. 1, p. 012091). IOP Publishing.

[8] Shayegan, M. J., Sabor, H. R., Uddin, M., & Chen, C. L. (2022). A Collective Anomaly Detection Technique to Detect Crypto Wallet Frauds on Bitcoin Network. Symmetry, 14(2), 328.

[9] Turner, A. B., McCombie, S., & Uhlmann, A. J. (2020). Analysis techniques for illicit bitcoin transactions. Frontiers in Computer Science, 2, 600596.

[10] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review, 21260.

[11] De Filippi, P., Mannan, M., & Reijers, W. (2020). Blockchain as a confidence machine: The problem of trust & challenges of governance. Technology in Society, 62, 101284.

[12] Antonopoulos, A.M. (2014) Mastering bitcoin, O'Reilly Online Learning. O'Reilly Media, Inc. Retrieved from https://www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch04.html (Accessed: December 2, 2022).

[13] Karthik, A., Mishra, H., Jayanth, S., Shobha, G., & Shetty, J. (2022, January). Performance skew prediction in HPCC systems. In 2022 12th International Conference on Cloud Computing, Data Science Engineering (Confluence) (pp. 94-97). IEEE.

[14] Fusek, M. (2023). Statistical Power of Goodness-of-Fit Tests for Type~ I Left-Censored Data. Austrian Journal of Statistics, 52(1), 51-61.

[15] Hodges, J. L. (1958). The significance probability of the Smirnov twosample test. Arkiv for Matematik, 3(5), 469-486.¨

[16] scipy.stats.ks_2samp (2022). SciPy v1.9.3 Manual. Retrieved from https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ks\ _2samp.html (Accessed: December 2, 2022).

[17] Scholz, F. W., & Stephens, M. A. (1987). K-sample Anderson–Darling tests. Journal of the American Statistical Association, 82(399), 918-924.

[18] scipy.stats.anderson_ksamp (2022). SciPy v1.9.3 Manual. Retrieved from https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.anderson\ ksamp.html (Accessed: December 2, 2022).

[19] Anderson, T. W. (1962). On the distribution of the two-sample Cramervon Mises criterion. The Annals of Mathematical Statistics, 1148-1159.

[20] scipy.stats.cramervonmises_2samp (2022) SciPy v1.9.3 Manual. Retrieved from https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.cramervonmises\ 2samp.html (Accessed: December 2, 2022).

[21] Xiang, Y., Ren, W., Gao, H., Bao, D., Lei, Y., Li, T., Yang, Q., Liu, W., Zhu, T., & Choo, K. K. R. (2022). BABD: A Bitcoin Address Behavior Dataset for Address Behavior Pattern Analysis. arXiv preprint arXiv:2204.05746.

[22] Calvez, A.L. (2022) Alecalve/python-bitcoin-blockchain-parser: A python 3 bitcoin blockchain parser, GitHub. alecalve. Retrieved from https://github.com/alecalve/python-bitcoin-blockchain-parser (Accessed: December 3, 2022).

[23] Maheshwari, R. (2022) Breaking down Bitcoin blockchain using HPCC systems, HPCC Systems. HPCC Systems. Retrieved from https://hpccsystems.com/blog/RVCE-RohanM-Blockchain (Accessed: December 3, 2022).