Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation

G. Usha Rani[1], Saini Karthik[2], Shyam Thakkar[2], Lakshmi Manaswini Pulicharla[2],

Gaddam Saketh Kumar[2]

[1,2]Department of Information Technology

[1,2] CMR Engineering College, Kandlakoya, Medchal, Hyderabad.

ABSTRACT

Crypto currencies usage is increasing every year around the world. The Bitcoin is the one of the famous cryptocurrencies, which is an unofficial usable currency in various nations. The bitcoin transactions are increasing, which needs to be monitored carefully. However, the conventional methods are failed to analyze the bitcoin transaction effectively. Therefore, this work focused on development of bitcoin transaction network (BTN) using pattern matching rules (PMR). Initially, the dataset preprocessing is carried out to identify the missed symbols, unknown characters from forensic blockchain dataset. Then, Petri-Net model applied on preprocessed dataset, which identifies the time stamp, transaction id, work tera hash, and work error properties. The Petri-Net model mainly used to parse and build the BTN model. Then, PMR conditions are developed to extract the transaction addresses extracted with time stamp details. So, PMR detects the illegal payment addresses by matching the known data with illegal (spam) addresses. Further, cache based PMR (CPMR) is also applied to detect the fraud transaction, which store all previous detected illegal payment addresses. So, for every new transaction, CPMR will ignore all those previously stored (detected) illegal payment addresses. This phenomenon causes reduction of fraud transaction detection time and processing becomes faster. The simulations shows that the proposed method resulted in reduced transaction processing time (TPT), fraud transaction detection time (FTDT), and improved fault transaction detection accuracy (FTDA) as compared to conventional methods.

Keywords: Bitcoin transactions, pattern matching rules, Blockchain network.

1. INTRODUCTION

Since Satoshi Nakamoto first introduced bitcoin, its popularity as an alternate method of payment has grown significantly over the last several years [1]. At the end of 2021, it was estimated that the market value of Bitcoin had surpassed $200 billion. Bitcoins are often not linked to user identities like usernames. Due to its pseudonymous character, Bitcoin is mistakenly thought of as an anonymous mode of payment on the Internet and as a means of enabling untraceable transactions during illicit dealings. Tracking Bitcoins linked to a known address is often not a problem. However, it has been difficult to trace Bitcoins since criminals often use ambiguous and hazy addresses.

Figure 1 shows the various bitcoin frauds occurred in different countries ike Vietnam, united states, United Kingdom, Ukraine, turkey, south Africa, Russia, south Africa, and China. The bitcoin frauds are majority based on darknet markets, ransomware, scams, and stolen funds [2]. In order to deal with this, various works aims to separate bitcoin fraud addresses. Generally speaking, some transactions may show commonalities and recurring trends. For instance, bitcoin transactions [3] were used to accumulate Bitcoins often link an output address to a number of input addresses. When monitoring ambiguous and improbable transactions, examining the connections between such input and output addresses may provide insightful information. However, such analysis involves additional challenges like defining the characteristics of bitcoin transactions, successfully identifying the characteristics that can be used to identify suspects.

With the help of our pattern matching technology, we have discovered static and dynamic Bitcoin transaction attributes that identify Bitcoin transaction patterns for analysis and locating questionable addresses. The evolution of the Bitcoin gene, which is integrated in Petri-Net transitions [4], is another significant addition. The movement of Bitcoins may be quickly and reliably tracked and analyzed using bitcoin transaction. Additionally, based on the combinations of match rules, this study suggests a set of match criteria to discover transactions and get suspicious addresses [5].

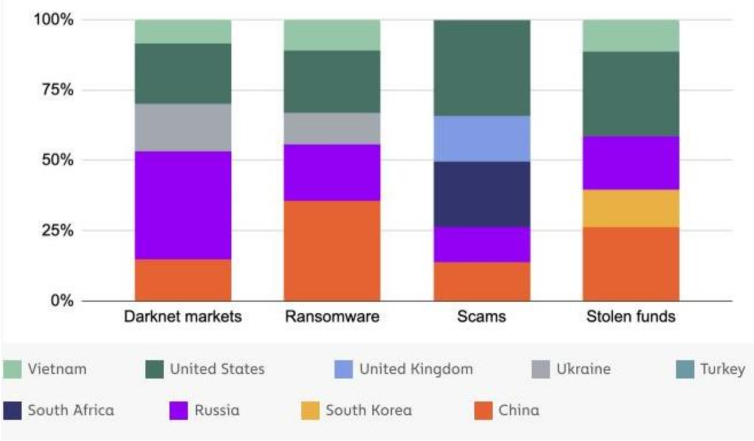


Fig. 1: Bitcoin frauds in various countries.

Objective

On the dark web, popular cryptocurrencies that are based on blockchain technology, such as Bitcoin, are increasingly being used maliciously to launder money. Recent developments in the fields of address clustering and Bitcoin flow analysis are gaining traction as potential solutions to the problems of tracing and analysing suspicious Bitcoin transactions and addresses. However, the currently available approaches only concentrate on Bitcoin addresses and the flow of Bitcoin transactions, ignoring other essential information such as the transaction structure and behaviour characteristics. This article offers a Bitcoin transaction network analytic approach for assisting Blockchain forensic research based on an extended secure Petri Net. The goal of the method is to make advantage of all of the relevant properties that are associated with transactions. In the model that we have suggested, the structural properties and dynamic semantics of Petri nets are used in order to characterise the static and dynamic properties of Bitcoin transactions. There have been found to be nineteen characteristics that describe Bitcoin transaction patterns for the purpose of evaluating and discovering suspicious addresses. The Bitcoin gene has been included into the Petri net transitions in order to correctly track and evaluate the movement of Bitcoin. In conclusion, marginal distribution analysis of Bitcoin transaction attributes and data visualisation methods are used in order to get rid of certain false positive samples even further and to increase the accuracy of detecting questionable addresses. The analytical approach for the Bitcoin transaction network that was suggested offers a trustworthy model for forensic investigations as well as a prototype platform, both of which are helpful for the protection of financial data. Based on the results of an investigation into a real-world case study, our method's effectiveness has been objectively confirmed.

2. LITERATURE SURVEY

The owner of bitcoin addresses an is unknown and is under suspicion. However, the owner of Bitcoin addresses a may be inferred if the owner of bitcoin address, which is inside the same cluster as a, is known. Addresses are divided into clusters, when they are utilized as transaction inputs. Addresses a and b, for instance, are grouped together into a single cluster if they are used as inputs for transaction t1. Numerous research [6] make advantage of this input address clustering technique. A user graph was built using the bitcoin address clustering (BAC) method, and significant users were identified using PageRank. To recover the "change" from whatever transaction the user has issued.

In addition to clustering addresses using the input and change address clustering methods [7], they also assessed the efficacy of the change address clustering approach. BitIodine is a modular framework developed in [8], that parses the blockchain, groups address that are probably owned by the same person or group of users, and visualizes complicated data taken from the Bitcoin network. The two clustering techniques are also used by BitIodine to group addresses.

Transactions involving Bitcoin have been examined using modified Petri-Net [9]. Bitcoin addresses are represented by Petri-Net locations and transitions. This Petri-Net model employed to group addresses and discovered common patterns of behavior, such as the use of a specific address just once.

In [10] authors examined disposable addresses to addresses using machine learning based approaches. A power-law distribution characterizes the lengths of these chains. The Bitcoin addresses were employed in these two models as Petri-Net locations or inputs for Bitcoin transactions. On the BTN, inputs for transactions are often coins rather than addresses. As a result, such models are unable to assess and quantify transaction aspects effectively. The expanded Petri-Net for the study of Bitcoin transactions that were presented in contrast to the current approaches that seek to identify behavior factors underlying Bitcoin transactions.

The research conducted by Pinna [11] using Petri net looked at disposable addresses, sometimes known as addresses that are only used once. Transactions are thought to form chains, and the lengths of these chains are thought to follow a power-law distribution, according to the writings of [12]. In these two models, Bitcoin addresses were employed as Petri net sites or inputs for Bitcoin transactions. However, in the Bitcoin transaction net, the inputs of Bitcoin transactions are not often addresses but rather currencies.

In contrast to the current approaches, which focus on identifying the behaviour patterns that are behind Bitcoin transactions, our method seeks to create transaction patterns based on the attributes of transactions and then locate problematic addresses on the basis of the patterns. Monaco [13] established and validated an assumption that it is possible to identify and verify Bitcoin users by observing the characteristics of Bitcoin transactions as they evolve over time. This assumption remains true. This investigation came to the conclusion, after doing an analysis of the behavioural characteristics utilising 366 user samples, that the behavioural patterns seen over time may be utilised to deprive a user; however, this information was not further developed into a model. In a manner similar to that of Monaco [13],

In addition, data visualisation techniques have been used in the process of Bitcoin transaction analysis. Coin mixing services were evaluated for their ability to thwart tracing efforts by Moser et al. [14]. A data visualisation tool known as BitConeView was created by Battista et al. [14] to demonstrate how successful coin mixing services are.

Christin [15] carried out an exhaustive measuring examination of the data acquired from web crawling that pertained to Silk Road. The data characters have been shown using these several approaches of data visualisation.

3. PROPOSED SYSTEM

Criminals, on the other hand, plan to conceal their Bitcoin addresses in the real world. Due to the paucity of known samples, it is challenging to locate their locations in order to study the transaction attributes, which restricts their practical application. Figure 2 shows the proposed BTN framework. This effort focused on the creation of BTN utilizing PMR. Initially, dataset pre-processing is performed to discover missing symbols and unfamiliar characters in the forensic blockchain dataset. Here, the information is saved into the database using an open-source program called Bitcoin Database Generator. The Petri-Net model is then used to the pre-processed dataset, identifying the time stamp, transaction id, work tera hash, and work error attributes. The Petri-Net model is primarily used to parse and construct the BTN model. Then, PMR conditions are created to retrieve the collected transaction addresses with time stamp data. As a result, PMR identifies illicit payment addresses by comparing known data to illegal addresses. Furthermore, CPMR is used to identify fraudulent transactions, which stores all previously recognized unlawful payment addresses. As a result, for each new transaction, CPMR will disregard any previously recorded (detected) unlawful payment addresses. This effect reduces the time required to identify fraud in transactions and speeds up processing. When compared to existing approaches, the simulations demonstrate that the suggested method TPT, FTDT, and enhanced FTDA.

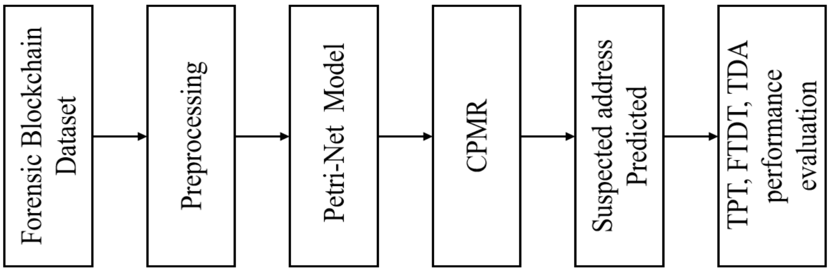

Forensic Blockchain Dataset
Preprocessing
Petri-Net Model
CPMR
Suspected address Predicted
TPT, FTDT, TDA performance evaluation

Fig. 2: Proposed BTN-CPME block diagram.

3.1 Pre-processing

The raw forensic blockchain dataset contains noises, missing values, which caused to complicated training of CPMR model. Further, it will reduce the classification, prediction performance. So, the data preprocessing operation is performed to overcome these problems. The preprocessing operation will replace unknown symbols, missing vales with the known nearest values. The efforts of filtering options that may be used to discover certain transactions or addresses. This approach defined a transaction pattern and instead took into account transaction characteristics independently. Methods of visualization often rely on visual perception to get outcomes. However, certain outcomes often go unnoticed because of the capacity limitations of the human brain. In our solution, the pattern matching algorithm instead of visual perception is used to match transaction patterns. As a result, it is more effective and seldom overlooks details. In order to filter out certain false positive samples using our suggested strategy, marginal distributions of several transaction characteristics are shown using visualization methods rather than being directly analyzed.

3.2 Petri-Net Model

A formal mathematical model called a Petri-Net, which is used to explore concurrent and asynchronous processes in distributed systems. An alternative name for it is a place/transition (PT) net. It was initially developed in 1962 by Carl Adam Petri. It is a bipartite directed graph with two different kinds of nodes, locations, and transitions. Directed arcs link the locations with the transitions, indicating which locations serve as inputs before transitions take place and outputs once they do. Arcs can only link locations to transitions or locations to transitions. Tokens are stored in places. Transitions cannot keep any tokens, but places may store an endless amount of them. The distribution of tokens among locations determines the state or marking of a Petri-Net. Figure 3 depicts a straightforward net with all the components of a Petri-Net, where the circles represent locations and the rectangle a transition. The formal definition of a Petri-Net is a tuple N = (P, T, F, M0), where P and T are disjoint finite sets of locations and transitions, respectively, F is a set of arcs (or incidence function), and M0 is the initial marking where M0: 1, 2, 3, J.

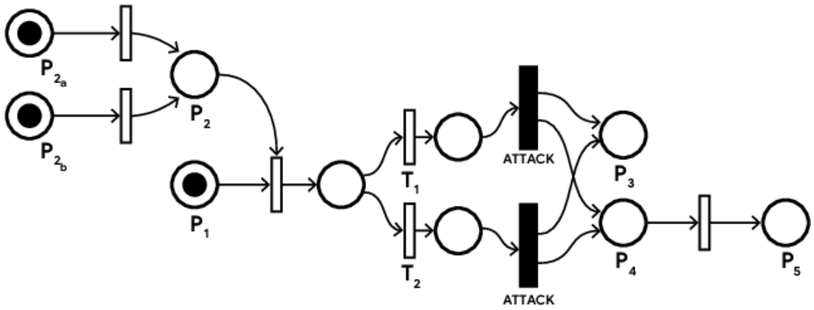


Fig. 3: Operational flow of Petri-Net.

The Petri-Net are more effective at catching concurrent actions as an assault progressed. Petri-Net have since been used to simulate physical and digital assaults on a variety of systems and networks. An innovative approach for studying the BTN was suggested in this research. This system formalizes Bitcoin transactions as an extended Safe Petri-Net known as BTN. The static and dynamic properties of a Bitcoin transaction are described by its structure and semantic features. The Bitcoin flow analysis may exploit the gene characteristic of bitcoins. Different transaction patterns may be developed based on the qualities that have been stated. It is possible to identify the addresses that fit the patterns. Based on a review of actual case studies, the suggested technique has been shown to be a useful tool for forensic investigation of future Bitcoin transactions. Pattern expressions are manually designed for our investigations. The next stage will be to create a compiler to automatically turn patterns into code. We shall then keep looking on ways to preserve BTN's interim states. In our tests, the Bitcoin Blockchain is parsed using the open-source application bitcoin database generator. The performance of the analysis is impacted since it does not retrieve information about block and transaction ordering.

3.3 CPMR Prediction

The CPMR to locate suspected addresses that does not fit a predetermined pattern of bitcoin transactions. Directly describing a complex pattern, however, is difficult. As a result, we provide guidelines for defining a pattern using logical expressions. Figure 4 shows suspected activity detection using CPMR. A collection of attributes describes a pattern and collection of feature expressions constitutes a property. According to this, a feature expression is a logical expression over features. In actuality, all three expressions—pattern, property, and feature—are logical expressions over features. A feature expression is used to characterize a Bitcoin transaction feature's personality. An element in a pattern is described by a property expression. An expression of a pattern depicts a pattern. In general, we should examine the elements that make up a pattern before attempting to describe it. A property should then be specified by one or more feature expressions for each aspect.

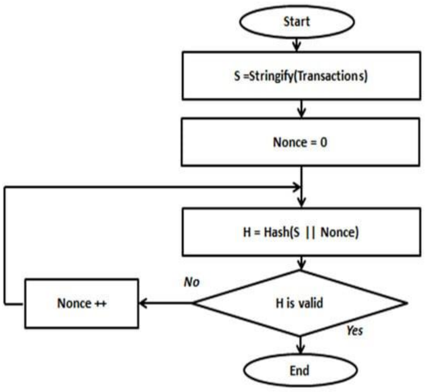


Fig. 4: Suspected address prediction using CPMR.

The nonce N that generates a valid hash, as shown in Figure 4, represents the precise number of hashes that were executed in order to find the valid address and suspected address. As a result, the time T needed to mine a block with a specified hash power H can be accurately calculated using the straightforward formula T=H /N. Since the nonce is used to create the lock hash when a new block is mined and the block mining time is known to all network peers, it is simple to determine the hash rate that was used. Only if the newly proposed block was hashed at the permitted hash rate should it be approved.

3.4 Blockchain

Blockchain is a decentralized, digital ledger technology that is used to record and store data in a secure and transparent manner. It is a distributed ledger, meaning that it is maintained by a network of computers, rather than being controlled by a single entity. Each block in the chain contains a set of transactions, and once a block is added to the chain, it cannot be altered or deleted. This makes blockchain an immutable and tamper-resistant technology that is particularly well-suited for storing and transmitting sensitive data.

Blockchain technology is perhaps best known for its use in cryptocurrencies like Bitcoin and Ethereum, but it has a wide range of other potential applications as well. These include supply chain management, identity verification, voting systems, and more. The decentralized nature of blockchain means that it has the potential to disrupt a variety of industries and business models by enabling trust and transparency in transactions and data exchange.

Concepts

There are several key concepts that are important to understand when it comes to blockchain technology:

Decentralization: Blockchain is a decentralized technology, meaning that it is not controlled by any single entity, but rather maintained by a network of participants. This increases transparency, security, and resilience.

Distributed ledger: Blockchain technology uses a distributed ledger to record and store data. Each block in the chain contains a set of transactions, and once a block is added to the chain, it cannot be altered or deleted.

Cryptography: Blockchain technology uses advanced cryptographic algorithms to secure transactions and data exchange, making it highly resistant to hacking and cyber attacks.

Consensus mechanism: In a blockchain network, participants must agree on the validity of transactions before they are recorded on the blockchain. Different blockchain networks use different consensus mechanisms to achieve this, such as Proof of Work or Proof of Stake.

Smart contracts: Smart contracts are self-executing contracts with the terms of the agreement directly written into code. They can be used to automate complex transactions and ensure that all parties involved in a transaction adhere to the terms of the contract.

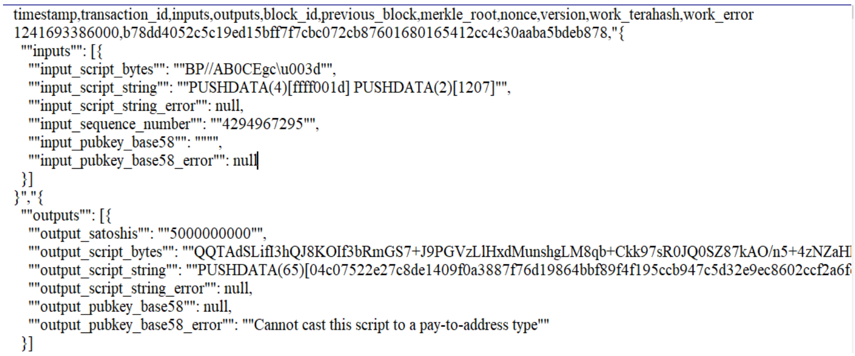
Tokenization: Blockchain technology enables the creation of digital tokens that can be used to represent a variety of assets, such as currencies, commodities, or even real estate.

4. RESULTS AND DISCUSSION

This section gives the detailed results analysis of proposed method, which are implemented using BTN-CPMR model.

Dataset

This Data Set associates actual entities with Bitcoin transactions that fall into the licit and illegal categories, such as exchanges, wallet providers, miners, and other licit service providers (scams, malware, terrorist organizations, ransomware, Ponzi schemes, etc.). Sorting the graph's illegal and legitimate nodes is the job at hand with this dataset. The information, which includes 200,000 transactions totaling more than $6 billion, will be used to assist the bitcoin industry detect dishonest individuals. The biggest collection of annotated bitcoin transaction data in the world has been created by researchers at MIT's IBM-funded AI Lab and blockchain forensics firm Elliptic. The labeling draws attention to distinctive transaction features and may be used to identify criminal actors operating in the cryptocurrency industry." Figure 5 shows the sample dataset with time stamp, transaction id, work tera hash, and work error properties.



```
timestamp,transaction_id,inputs,outputs,block_id,previous_block,merkle_root,nonce,version,work_terahash,work_error
1241693386000,b78dd4052c5c19ed15bff7f7cbc072cb87601680165412cc4c30aaba5bdeb878,"{
 ""inputs"": [{
  ""input_script_bytes"": ""BP//AB0CEgc\u003d"",
  ""input_script_string"": ""PUSHDATA(4)[ffff001d] PUSHDATA(2)[1207]"",
  ""input_script_string_error"": null,
  ""input_sequence_number"": ""4294967295"",
  ""input_pubkey_base58"": """",
  ""input_pubkey_base58_error"": null
 }]
}","{
 ""outputs"": [{
  ""output_satoshis"": ""5000000000"",
  ""output_script_bytes"": ""QQTAdSLifI3hQJ8KOIf3bRmGS7+J9PGVzLlHxdMunshgLM8qb+Ckk97sR0JQ0SZ87kAO/n5+4zNZaH
  ""output_script_string"": ""PUSHDATA(65)[04c07522e27c8de1409f0a3887f76d19864bbf89f4f195ccb947c5d32e9ec8602ccf2a6f
  ""output_script_string_error"": null,
  ""output_pubkey_base58"": null,
  ""output_pubkey_base58_error"": ""Cannot cast this script to a pay-to-address type""
 }]
```

Fig. 5: Sample dataset.

Performance evaluation

In Figure 6, x-axis represents total withdraw from account 0 to 1 and vice versa and y-axis represents number of gather addresses for that withdrawal. In Figure 7, x-axis represents number of account ID and y-axis represents number of deposit transaction made by that account. Further, Table 1 shows that the proposed BTN-CPMR protocol resulted in higher security standards compared to BAC [10], BitIodine [13], and BlockChainVis [20]. Because, the proposed BTN-CPMR approach reduced the TPT (ms), FTDT (ms), and increased the FTDA (%).

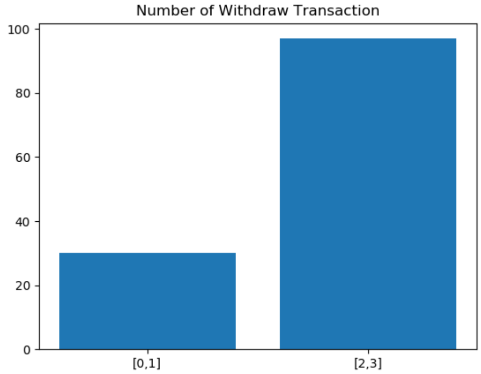


Fig. 6: Number of withdraw transactions.

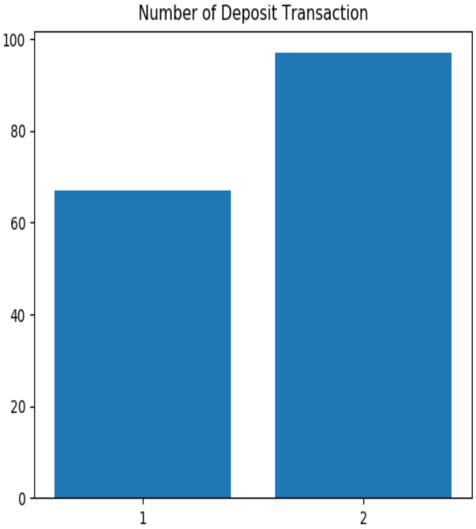


Fig. 7: Number of withdraw transactions.

Table. 1: Performance comparison.

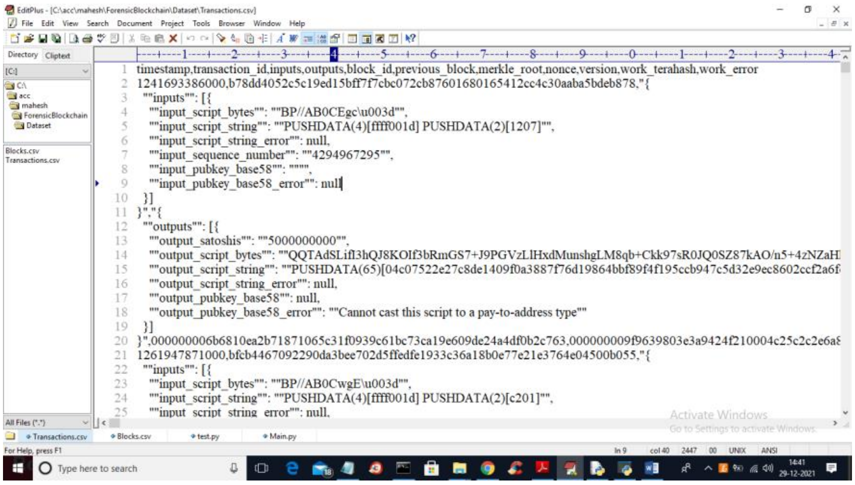
Method	FTDA (%)	TPT (ms)	FTDT (ms)
BAC [10]	91.056	43.614	42.516
BitIodine [13]	92.969	21.661	35.905
BlockChainVis [20]	93.636	17.308	17.456
Proposed BTN-CPMR	98.927	9.352	8.440

UI OUTPUT

To implement above project we have downloaded ‘Blockchain’ dataset from below link

https://console.cloud.google.com/bigquery?project=drops-311506&j=bq:US:bquxjob_d45c0c3_17e016a5f6a&page=queryresults

Dataset contains below static features shown in screen shots


EditPlus - [C:\acc\mahesh\ForensicBlockchain\Dataset\Transactions.csv]
File Edit View Search Document Project Tools Browser Window Help
Directory Cliptext
Blocks.csv
Transactions.csv
Transactions.csv
Blocks.csv
test.py
Main.py
For Help, press F1
ln 9 col 40 2447 00 UNIX ANSI
Type here to search
14:41
29-12-2021

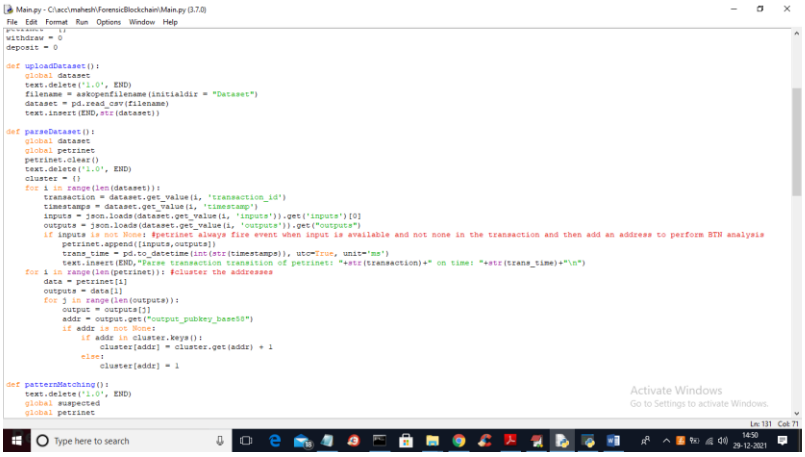
```
timestamp,transaction_id,inputs,outputs,block_id,previous_block,merkle_root,nonce,version,work_terahash,work_error
1241693386000,b78dd4052c5c19ed15bff7f7cbc072cb87601680165412cc4c30aaba5bdeb878,"{
 ""inputs"": [{
  ""input_script_bytes"": ""BP//AB0CEgc\u003d"",
  ""input_script_string"": ""PUSHDATA(4)[ffff001d] PUSHDATA(2)[1207]"",
  ""input_script_string_error"": null,
  ""input_sequence_number"": ""4294967295"",
  ""input_pubkey_base58"": """",
  ""input_pubkey_base58_error"": null
 }]
}","{
 ""outputs"": [{
  ""output_satoshis"": ""5000000000"",
  ""output_script_bytes"": ""QQTAdSLifl3hQJ8KOIf3bRmGS7+J9PGVzLlHxdMunshgLM8qb+Ckk97sR0JQ0SZ87kAO/n5+4zNZaH
  ""output_script_string"": ""PUSHDATA(65)[04c07522e27c8de1409f0a3887f76d19864bbf89f4f195ccb947c5d32e9ec8602ccf2a6f
  ""output_script_string_error"": null,
  ""output_pubkey_base58"": null,
  ""output_pubkey_base58_error"": ""Cannot cast this script to a pay-to-address type""
 }]
}",000000006b6810ea2b71871065c31f0939c61bc73ca19e609de24a4df0b2c763,000000009f9639803e3a9424f210004c25c2c2e6a8
1261947871000,bfcb4467092290da3bee702d5ffedfe1933c36a18b0e77e21e3764e04500b055,"{
 ""inputs"": [{
  ""input_script_bytes"": ""BP//AB0CwgE\u003d"",
  ""input_script_string"": ""PUSHDATA(4)[ffff001d] PUSHDATA(2)[c201]"",
  ""input_script_string_error"": null,
```

In above screen first row contains dataset column names as STATIC features and then we will calculate deposit and received values as dynamic features. Above Blockchain transaction will be analysed to detect suspected addresses.

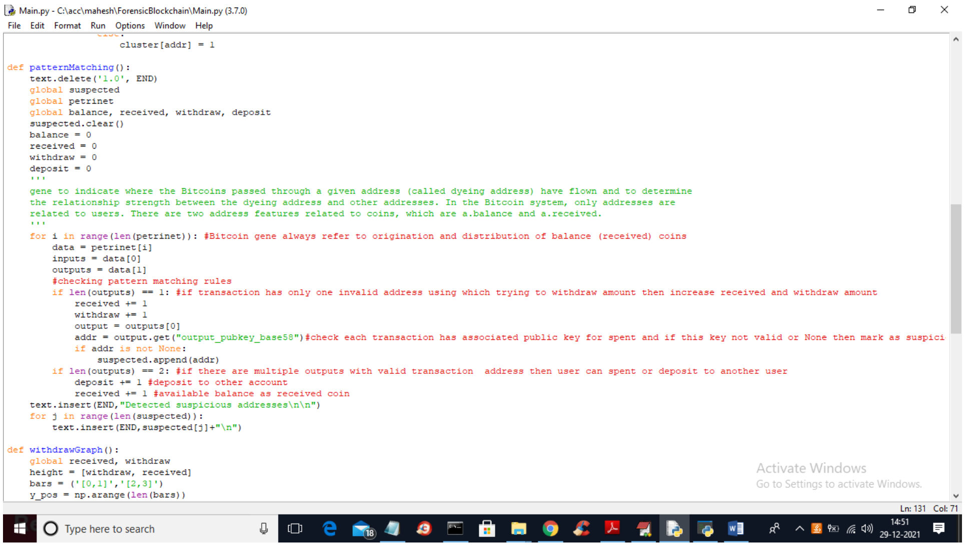
To implement this project, we have designed following modules

1) Upload Blockchain Transaction: using this module we will upload Blockchain Bitcoin transaction dataset to application
2) Parse & Build BTN Petrinet Simulation: using this module we will parse dataset and then build BTN model using PETRINET.
3) Run Pattern Matching Rules Algorithm: using this module pattern matching rules will be applied on BTN network to identify all deposit and withdraw transaction and then detect suspected address by identifying those transaction with None or invalid address.

Below screen with red and green colour comments showing code for above concept implementation


Main.py - C:\acc\mahesh\ForensicBlockchain\Main.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 131 Col: 71
Type here to search
14:50
29-12-2021

```
withdraw = 0
deposit = 0

def uploadDataset():
    global dataset
    text.delete('1.0', END)
    filename = askopenfilename(initialdir = "Dataset")
    dataset = pd.read_csv(filename)
    text.insert(END,str(dataset))

def parseDataset():
    global dataset
    global petrinet
    petrinet.clear()
    text.delete('1.0', END)
    cluster = {}
    for i in range(len(dataset)):
        transaction = dataset.get_value(i, 'transaction_id')
        timestamps = dataset.get_value(i, 'timestamp')
        inputs = json.loads(dataset.get_value(i, 'inputs')).get('inputs')[0]
        outputs = json.loads(dataset.get_value(i, 'outputs')).get("outputs")
        if inputs is not None: #petrinet always fire event when input is available and not none in the transaction and then add an address to perform BTN analysis
            petrinet.append([inputs,outputs])
            trans_time = pd.to_datetime(int(str(timestamps)), utc=True, unit='ms')
            text.insert(END,"Parse transaction transition of petrinet: "+str(transaction)+" on time: "+str(trans_time)+"\n")
    for i in range(len(petrinet)): #cluster the addresses
        data = petrinet[i]
        outputs = data[1]
        for j in range(len(outputs)):
            output = outputs[j]
            addr = output.get("output_pubkey_base58")
            if addr is not None:
                if addr in cluster.keys():
                    cluster[addr] = cluster.get(addr) + 1
                else:
                    cluster[addr] = 1

def patternMatching():
    text.delete('1.0', END)
    global suspected
    global petrinet
```


Main.py - C:\acc\mahesh\ForensicBlockchain\Main.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 131 Col: 71
Type here to search
14:51
29-12-2021

```
                    cluster[addr] = 1

def patternMatching():
    text.delete('1.0', END)
    global suspected
    global petrinet
    global balance, received, withdraw, deposit
    suspected.clear()
    balance = 0
    received = 0
    withdraw = 0
    deposit = 0
    '''
    gene to indicate where the Bitcoins passed through a given address (called dyeing address) have flown and to determine
    the relationship strength between the dyeing address and other addresses. In the Bitcoin system, only addresses are
    related to users. There are two address features related to coins, which are a.balance and a.received.
    '''
    for i in range(len(petrinet)): #Bitcoin gene always refer to origination and distribution of balance (received) coins
        data = petrinet[i]
        inputs = data[0]
        outputs = data[1]
        #checking pattern matching rules
        if len(outputs) == 1: #if transaction has only one invalid address using which trying to withdraw amount then increase received and withdraw amount
            received += 1
            withdraw += 1
            output = outputs[0]
            addr = output.get("output_pubkey_base58")#check each transaction has associated public key for spent and if this key not valid or None then mark as suspici
            if addr is not None:
                suspected.append(addr)
        if len(outputs) == 2: #if there are multiple outputs with valid transaction  address then user can spent or deposit to another user
            deposit += 1 #deposit to other account
            received += 1 #available balance as received coin
    text.insert(END,"Detected suspicious addresses\n\n")
    for j in range(len(suspected)):
        text.insert(END,suspected[j]+"\n")

def withdrawGraph():
    global received, withdraw
    height = [withdraw, received]
    bars = ('[0,1]','[2,3]')
    y_pos = np.arange(len(bars))
```

In above screen read red and green colour comments to know about coding concept described in the ref. paper.

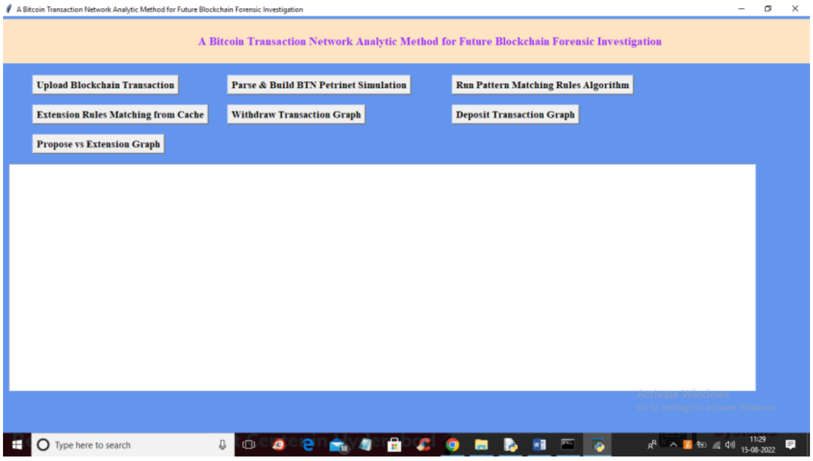
Extension work

BTN network analysing Bitcoin Gene which will track address for illegal payment by searching received and send account address and due to growing usage of Bitcoin Transaction a large amount of data will be gather and analysing such huge data to detect illegal payment will require huge computing resources and execution or processing time and to avoid this huge processing time as extension we are adding CACHE memory. Cache memory will store all previous detected illegal payment addresses and while reprocessing Cache Memory will ignore all those illegal payment addresses which already processed and scan only new addresses and by ignoring process address, we can reduce execution time and processing will be faster.

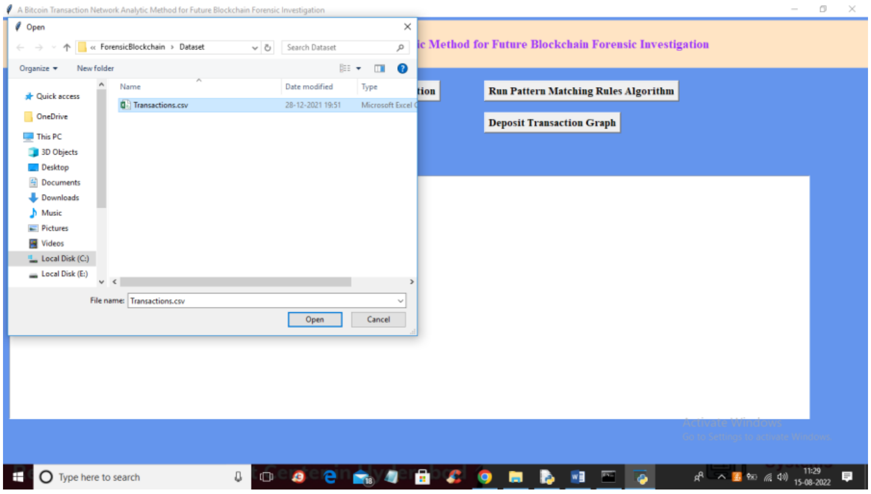
In this project we are detecting illegal payment addresses using Propose Algorithm and Extension Cache memory algorithm and then we are comparing execution time of both algorithms.

SCREEN SHOTS

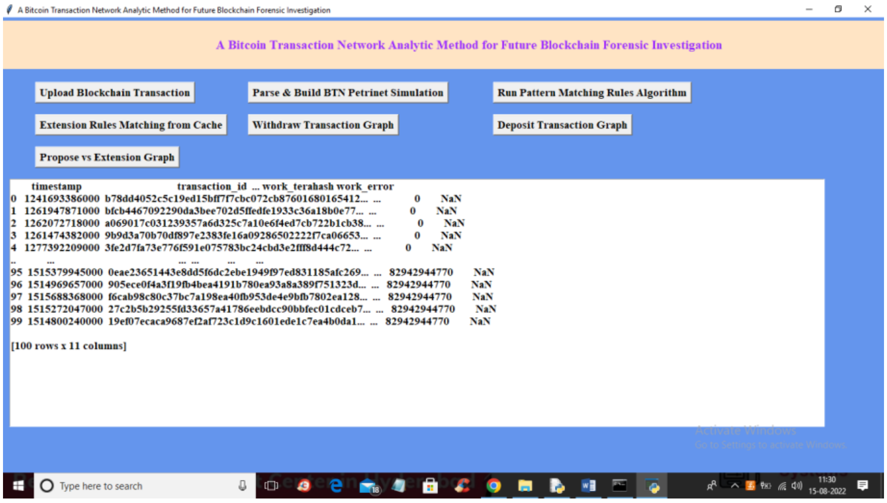
To run project double click on 'run.bat' file to get below screen


A Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation
A Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation
Upload Blockchain Transaction
Parse & Build BTN Petrinet Simulation
Run Pattern Matching Rules Algorithm
Extension Rules Matching from Cache
Withdraw Transaction Graph
Deposit Transaction Graph
Propose vs Extension Graph
Type here to search
11:29
15-08-2022

In above screen click on 'Upload Blockchain Transaction' button to upload transaction dataset and get below output

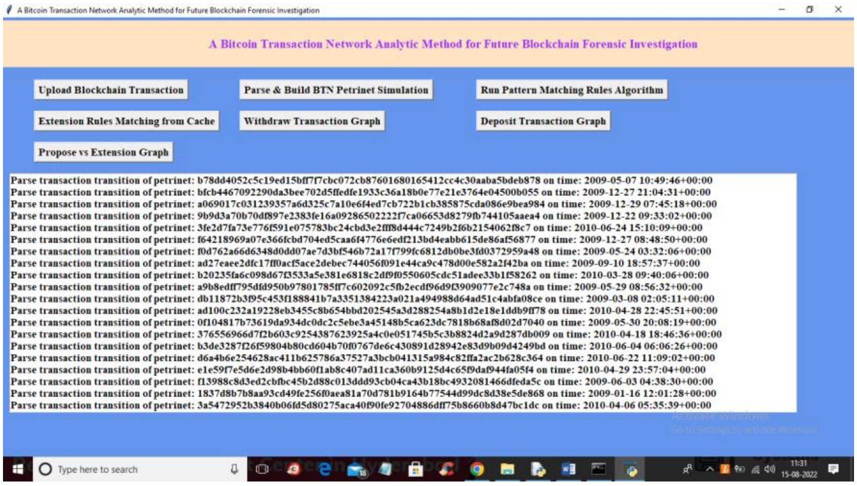


In above screen selecting and uploading Transaction.csv file and then click on 'Open' button to get below output


A Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation
A Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation
Upload Blockchain Transaction
Parse & Build BTN Petrinet Simulation
Run Pattern Matching Rules Algorithm
Extension Rules Matching from Cache
Withdraw Transaction Graph
Deposit Transaction Graph
Propose vs Extension Graph
Type here to search
11:30
15-08-2022

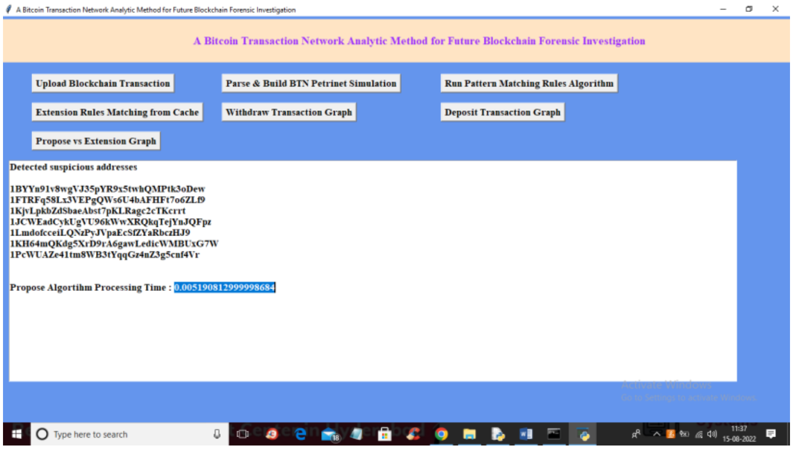
```
     timestamp                        transaction_id  ... work_terahash work_error
0  1241693386000  b78dd4052c5c19ed15bff7f7cbc072cb87601680165412...  ...         0      NaN
1  1261947871000  bfcb4467092290da3bee702d5ffedfe1933c36a18b0e77...  ...         0      NaN
2  1262072718000  a069017c031239357a6d325c7a10e6f4ed7cb722b1cb38...  ...         0      NaN
3  1261474382000  9b9d3a70b70df897e2383fe16a09286502222f7ca06653...  ...         0      NaN
4  1277392209000  3fe2d7fa73e776f591e075783bc24cbd3e2fff8d444c72...  ...         0      NaN
..           ...                               ...  ...         ...        ...
95 1515379945000  0eae23651443e8dd5f6dc2ebe1949f97ed831185afc269...  ...  82942944770      NaN
96 1514969657000  905ece0f4a3f19fb4bea4191b780ea93a8a389f751323d...  ...  82942944770      NaN
97 1515688368000  f6cab98c80c37bc7a198ea40fb953de4e9bfb7802ea128...  ...  82942944770      NaN
98 1515272047000  27c2b5b29255fd33657a41786eebdcc90bbfec01cdceb7...  ...  82942944770      NaN
99 1514800240000  19ef07ecaca9687ef2af723c1d9c1601ede1c7ea4b0da1...  ...  82942944770      NaN

[100 rows x 11 columns]
```

In above screen dataset loaded and now click on 'Parse & Build BTN Petrinet Simulation' button to extract transaction address and build Petrinet object


A Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation
A Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation
Upload Blockchain Transaction
Parse & Build BTN Petrinet Simulation
Run Pattern Matching Rules Algorithm
Extension Rules Matching from Cache
Withdraw Transaction Graph
Deposit Transaction Graph
Propose vs Extension Graph
Type here to search
11:31
15-08-2022

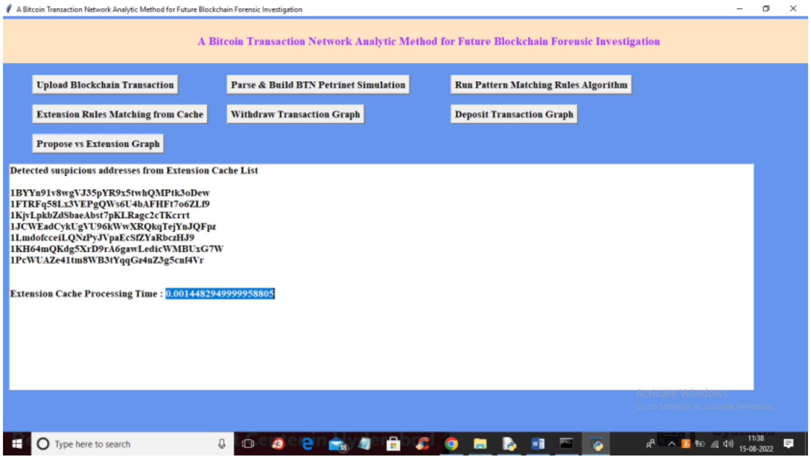
```
Parse transaction transition of petrinet: b78dd4052c5c19ed15bff7f7cbc072cb87601680165412cc4c30aaba5bdeb878 on time: 2009-05-07 10:49:46+00:00
Parse transaction transition of petrinet: bfcb4467092290da3bee702d5ffedfe1933c36a18b0e77e21e3764e04500b055 on time: 2009-12-27 21:04:31+00:00
Parse transaction transition of petrinet: a069017c031239357a6d325c7a10e6f4ed7cb722b1cb385875cda086e9bea984 on time: 2009-12-29 07:45:18+00:00
Parse transaction transition of petrinet: 9b9d3a70b70df897e2383fe16a09286502222f7ca06653d8279fb744105aaea4 on time: 2009-12-22 09:33:02+00:00
Parse transaction transition of petrinet: 3fe2d7fa73e776f591e075783bc24cbd3e2fff8d444c7249b2f6b2154062f8c7 on time: 2010-06-24 15:10:09+00:00
Parse transaction transition of petrinet: f64218969a07e366fcbd704ed5caa6f4776e6edf213bd4eabb615de86af56877 on time: 2009-12-27 08:48:50+00:00
Parse transaction transition of petrinet: f0d762a66d6348d0dd07ae7d3bf546b72a17f799fc6812db0be3fd0372959a48 on time: 2009-05-24 03:32:06+00:00
Parse transaction transition of petrinet: ad27eaee2dfc17ff0acf5ace2debec744056f091e44ca9c478d00e582a2f42ba on time: 2009-09-10 18:57:37+00:00
Parse transaction transition of petrinet: b20235fa6c098d67f3533a5e381e6818c2df9f0550605cdc51adee33b1f58262 on time: 2010-03-28 09:40:06+00:00
Parse transaction transition of petrinet: a9b8edff795dfd950b97801785ff7c602092c5fb2ecdf96d9f3909077e2c748a on time: 2009-05-29 08:56:32+00:00
Parse transaction transition of petrinet: db11872b3f95c453f188841b7a3351384223a021a494988d64ad51c4abfa08ce on time: 2009-03-08 02:05:11+00:00
Parse transaction transition of petrinet: ad100c232a19228eb3455c8b654bbd202545a3d288254a8b1d2e18e1ddb9ff78 on time: 2010-04-28 22:45:51+00:00
Parse transaction transition of petrinet: 0f104817b73619da934dc0dc2c5ebe3a45148b5ca623dc7818b68af8d02d7040 on time: 2009-05-30 20:08:19+00:00
Parse transaction transition of petrinet: 376556966d7f2b603c9254387623925a4c0e051745b5c3b8824d2a9d287db009 on time: 2010-04-18 18:46:36+00:00
Parse transaction transition of petrinet: b3de3287f26f59804b80cd604b70f0767de6c430891d28942e83d9b09d4249bd on time: 2010-06-04 06:06:26+00:00
Parse transaction transition of petrinet: d6a4b6e254628ac411b625786a37527a3bcb041315a984c82ffa2ac2b628c364 on time: 2010-06-22 11:09:02+00:00
Parse transaction transition of petrinet: e1e59f7e5d6e2d98b4bb60f1ab8c407ad11ca360b9125d4c65f9daf944fa05f4 on time: 2010-04-29 23:57:04+00:00
Parse transaction transition of petrinet: f13988c8d3ed2cbfbc45b2d88c013ddd93cb04ca43b18bc4932081466dfeda5c on time: 2009-06-03 04:38:30+00:00
Parse transaction transition of petrinet: 1837d8b7b8aa93cd49fe256f0aea81a70d781b9164b77544d99dc8d38e5de868 on time: 2009-01-16 12:01:28+00:00
Parse transaction transition of petrinet: 3a5472952b3840b06fd5d80275aca40f90fe92704886dff75b8660b8d47bc1dc on time: 2010-04-06 05:35:39+00:00
```

In above screen all transaction addresses extracted with time details and now click on 'Run Pattern Matching Rules Algorithm' button to detect illegal payment addresses by matching rules and get below output


A Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation
A Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation
Upload Blockchain Transaction
Parse & Build BTN Petrinet Simulation
Run Pattern Matching Rules Algorithm
Extension Rules Matching from Cache
Withdraw Transaction Graph
Deposit Transaction Graph
Propose vs Extension Graph
Type here to search

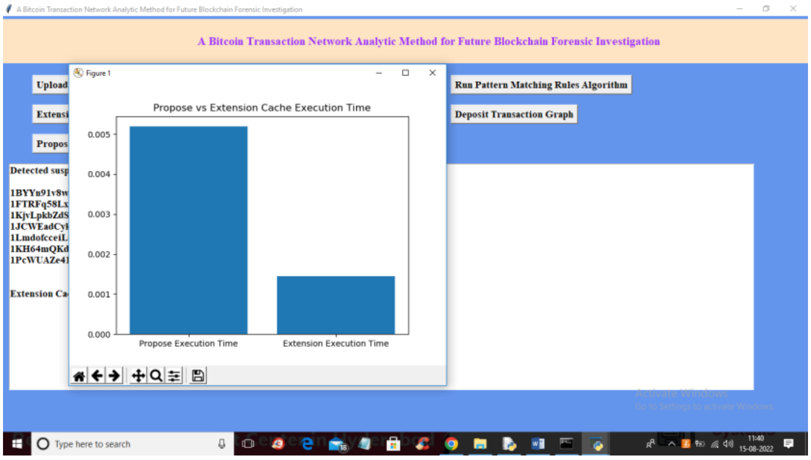
```
Detected suspicious addresses

1BYYn91v8wgVJ35pYR9x5twhQMPtk3oDew
1FTRFq58Lx3VEPgQWs6U4bAFHFt7o6ZLf9
1KjvLpkbZdSbaeAbst7pKLRagc2cTKcrrt
1JCWEadCykUgVU96kWwXRQkqTejYnJQFpz
1LmdofcceiLQNzPyJVpaEcSfZYaRbczHJ9
1KH64mQKdg5XrD9rA6gawLedicWMBUxG7W
1PcWUAZe41tm8WB3tYqqGz4nZ3g5cnf4Vr


Propose Algortihm Processing Time : 0.005190812999998684
```

In above screen propose algorithm displaying all suspected illegal payment addresses and to run propose algorithm system took 0.00519 seconds and now click on ‘Extension Rules Matching from Cache’ button to avoid already detected illegal payment addresses and scan only new transaction and get below output

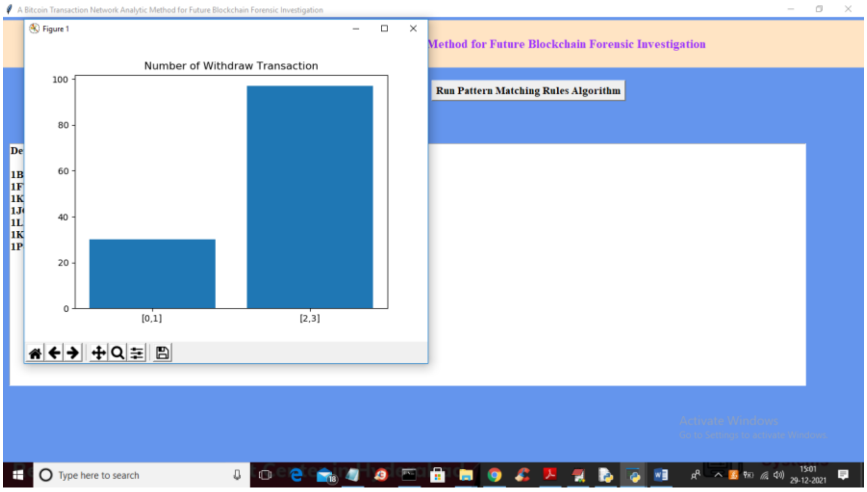

A Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation
A Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation
Upload Blockchain Transaction
Parse & Build BTN Petrinet Simulation
Run Pattern Matching Rules Algorithm
Extension Rules Matching from Cache
Withdraw Transaction Graph
Deposit Transaction Graph
Propose vs Extension Graph
Type here to search

```
Detected suspicious addresses from Extension Cache List

1BYYn91v8wgVJ35pYR9x5twhQMPtk3oDew
1FTRFq58Lx3VEPgQWs6U4bAFHFt7o6ZLf9
1KjvLpkbZdSbaeAbst7pKLRagc2cTKcrrt
1JCWEadCykUgVU96kWwXRQkqTejYnJQFpz
1LmdofcceiLQNzPyJVpaEcSfZYaRbczHJ9
1KH64mQKdg5XrD9rA6gawLedicWMBUxG7W
1PcWUAZe41tm8WB3tYqqGz4nZ3g5cnf4Vr


Extension Cache Processing Time : 0.0014482949999958805
```

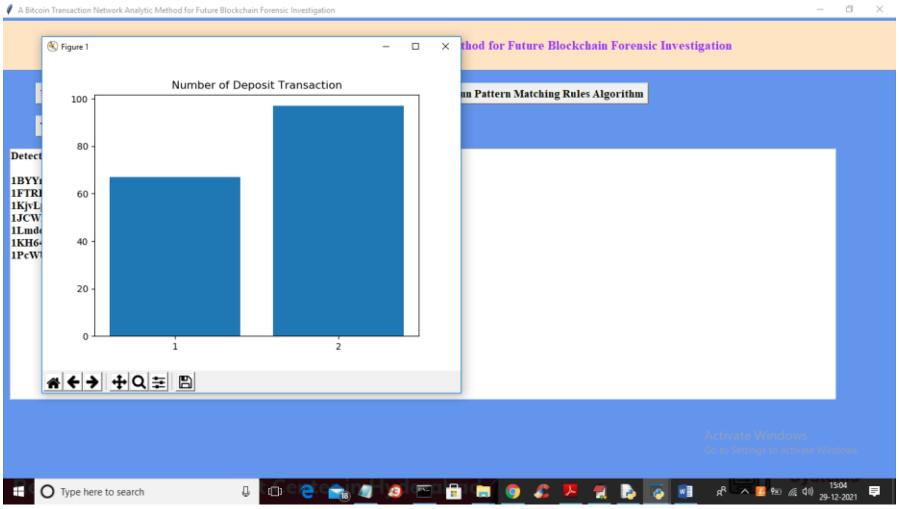
In above screen with extension Cache it took 0.0014 seconds to processes all transaction and now click on ‘Propose vs Extension Graph’ button to get below execution time graph



In above graph x-axis represents technique name and y-axis represents execution time and in above graph we can see Extension Cache technique took less execution time so it’s faster than proposed algorithm


A Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation
Figure 1
Method for Future Blockchain Forensic Investigation
Run Pattern Matching Rules Algorithm
Number of Withdraw Transaction
100
80
60
40
20
0
[0,1]
[2,3]
Type here to search
15:01
29-12-2021

In above screen x-axis represents total withdraw from account 0 to 1 and vice versa and y-axis represents number of gather addresses for that withdrawal. Now click on ‘Deposit Transaction Graph’ button to get below graph


A Bitcoin Transaction Network Analytic Method for Future Blockchain Forensic Investigation
Figure 1
thod for Future Blockchain Forensic Investigation
un Pattern Matching Rules Algorithm
Number of Deposit Transaction
100
80
60
40
20
0
1
2
Type here to search
15:04
29-12-2021

In above screen x-axis represents number of account ID and y-axis represents number of deposit transaction made by that account

5. CONCLUSION

The primary emphasis of this effort was placed on the construction of the BTN-CPMP. In the beginning, the dataset is preprocessed so that any missing symbols or unfamiliar characters in the forensic blockchain dataset may be located and accounted for. The preprocessed dataset is then subjected to a Petri-Net model application, which detects attributes such as the time stamp, transaction id, work tera hash, and work error. The Petri-Net model was primarily used in order to construct and parse the BTN model. The PMR criteria needed to extract the transaction addresses together with the time stamp data are then generated. Therefore, PMR is able to identify illicit payment addresses by comparing the known data with illegal addresses (spam addresses). In addition, a CPMR is used in order to identify fraudulent transactions. This PMR keeps a record of all unlawful payment addresses that have been identified in the past. Therefore, for every new transaction, CPMR will disregard all of those previously recorded (detected) unlawful payment addresses. This will protect the integrity of the network. This phenomenon produces a decrease in the amount of time needed to identify fraudulent transactions, resulting in a speedup of the processing. According to the results of the simulations, the proposed method led to a reduction in the amount of time required for the processing of transactions i.e., TPT, the amount of time required to detect fraudulent transactions i.e., FTDT, and an improvement in the FTDA when compared to the conventional methods. Further, this work can be extended with deep learning models for improved performance.

Future scope

This research presented an innovative methodology for the investigation of the Bitcoin transaction network. In this architecture, Bitcoin transactions are formalised as an expanded version of the Safe Petri net, which is referred to as BTN. Static and dynamic aspects of a Bitcoin transaction may be understood by its structure and the semantic qualities it has. The DNA feature of Bitcoins may be used for the investigation of the movement of Bitcoins. There are many other transaction patterns that may be defined based on the qualities that have been stated. It is possible to determine which addresses correspond to certain patterns. Based on a review of real-world case studies, the approach that was developed has been shown to be an effective instrument for use in future forensic investigations of Bitcoin transactions.

In our studies, the pattern expressions are created by hand programming. In the subsequent stage, a compiler will be constructed to automatically compile patterns into programmes. This will be the first step. The data included in the Bitcoin Blockchain is enormous; thus, it is possible to reduce the amount of time spent on doing a recent case study by saving intermediate states at various periods in time. Following that, we will proceed to look at further ways that intermediate states of BTN might be kept safe. In our research, the Bitcoin Blockchain is dissected with the help of an open-source programme called Bitcoin Database Generator. However, it does not retrieve information about blocks and transaction orders, which might negatively impact the analysis performance. Another one of our study goals for the future is to investigate this tool's downside, so stay tuned for that! Forensic examination of bitcoin will be aided in the future by our implementation of a fully integrated bitcoin analysis platform.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] G. White. UK company linked to laundered Bitcoin billions, BBC, (2018). Available: https://www.bbc.com/news/technology-43291026

[3] N. J. Ajello, "Fitting a Square Peg in a Round Hole: Bitcoin, Money Laundering, and the Fifth Amendment Privilege Against SelfIncrimination," Brooklyn Law Review, vol. 80, p. 4, 2015.

[4] F. Reid and M. Harrigan, "An Analysis of Anonymity in the Bitcoin System," in proceedings of 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, 2011, pp. 1318-1326.

[5] S. Göbel, A Polynomial Translation of Mobile Ambients Into Safe Petri Nets: Understanding a Calculus of Hierarchical Protection Domains: Springer, 2016.

[6] D. Ron and A. Shamir, "Quantitative Analysis of the Full Bitcoin Transaction Graph," in proceedings of International Conference on Financial Cryptography and Data Security Berlin, Heidelberg, 2013, pp. 6-24.

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

[8] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating User Privacy in Bitcoin," in proceedings of International Conference on Financial Cryptography and Data Security, Berlin, Heidelberg, 2013, pp. 34-51.

[9] C. Zhao and Y. Guan, "A GRAPH-BASED INVESTIGATION OF BITCOIN TRANSACTIONS," in proceedings of Advances in Digital Forensics XI, Cham, 2015, pp. 79-95.

[10] D. D. F. Maesa, A. Marino, and L. Ricci, "Uncovering the Bitcoin Blockchain: An Analysis of the Full Users Graph," in proceedings of 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2016, pp. 537-546.

[11] A. Pinna, "A Petri Net-based Model for Investigating Disposable Addresses in Bitcoin System," in proceedings of Knowledge Discovery on the WEB, 2016, pp. 1-4.

[12] J. V. Monaco, "Identifying Bitcoin users by transaction behavior," in proceedings of SPIE Defense + Security, 2015, p. 15.

[13] M. Möser, R. Böhme, and D. Breuker, "An inquiry into money laundering tools in the Bitcoin ecosystem," in proceedings of APWG eCrime Researchers Summit, 2013, pp. 1-14.

[14] G. D. Battista, V. D. Donato, M. Patrignani, M. Pizzonia, V. Roselli, and R. Tamassia, "Bitconeview: visualization of flows in the bitcoin transaction graph," in proceedings of IEEE Symposium on Visualization for Cyber Security (VizSec), 2015, pp. 1-8.

[15] N. Christin, "Traveling the silk road: a measurement analysis of a large anonymous online marketplace," presented at the Proceedings of the 22nd international conference on World Wide Web, Rio de Janeiro, Brazil, 2013.