Utrecht University

Fraudulent financial activity: graph analysis for fraud detection

Master Thesis

Piotr Stachyra

4889509

Under the supervision of Prof. Ioana Karnstedt-Hulpus and Vahid Shahrivari

Project's repository: https://github.com/p-stachyra/fraud_detection

Utrecht, July 2022

Abstract

This thesis aims to answer the question if graph-based methods can be employed on available financial datasets with the purpose of detecting illicit financial activities. The data was gathered from three separate data sets – one being a synthetic PaySim dataset, the second one provided by Vesta in cooperation with the Institute of Electrical and Electronics Engineers (IEEE) and the third one related to Bitcoin transactions. In all cases, exploratory analysis is applied to attempt to gain an initial overview of the data sets and presumably to identify certain characteristics which can serve to find additional methods for fraud detection. The data are analyzed using graph-based approaches which allows for retrieving centrality metrics for different classes of nodes indicating if they are involved in fraudulent activity or not. The outcomes were examined using goodness of fit analysis and descriptive statistics measures to determine if there are differences between groups of observations. At a general level of metrics distribution in different observation classes, Mann-Whitney U test was employed. Finally, Louvain modularity was used to gather information regarding dense communities which can constitute fraud rings. The results of this study suggest that some of the methods presented in this paper can be useful, however, precise, non-anonymized data must be provided to prove their efficacy. In all our experiments, the centrality metrics did not perform well for predicting fraud. Without additional information on the entity making a transaction it is not possible to flag potentially suspicious nodes accurately.

Keywords: financial network, fraud detection, graph properties, centrality, graph theory.

Contents

1 Introduction

Financial fraud is a wide-spread problem which generates costs of approximately 6% of GDP for countries such as Great Britain [1]. Novel techniques for fraud detection are needed, not only to improve performance of the flagging mechanisms, but also due to the fact that perpetrators attempt to evade detection using increasingly sophisticated methods. One possible solution is to use graphs which can be a reasonable alternative for a tabular data format. Graphs can tackle the problem of numerous relationships and interconnections between financial accounts and can lead to discovering patterns that are difficult to spot using other approaches. They can help identify fraud rings and other graph-related structures which are involved in fraudulent activities [2].

Such innovations and further exploration are necessary, as many anti-fraud systems are still based on thresholds and more sophisticated methods are needed to detect illicit financial activities effectively [3]. Considering these new possibilities, it is important to emphasize that a graph-based inferential study can provide useful insights about the significance of graph metrics, however, their further use in applications for fraud detection must be evaluated based on some appropriate performance measure. For instance, despite obtaining a relatively good recall score, the precision might be unsatisfactory, leading to flagging many legitimate transactions and generating additional expenses of a detailed evaluation [4]. These issues must be studied in further stages of developing the novel techniques based on additional graph metrics and among various models.

This research aims to answer the question if fraudulent transactions' characteristics differ from non-fraudulent transactions in terms of graph metrics. The analysis presented in this paper is *de facto* an exploratory process which is focused on inference.

2 Data

One of the biggest challenges for this thesis project was the lack of non-anonymized real-world datasets. Due to privacy issues, such datasets cannot be shared publicly. This led to conducting an analysis of datasets which are freely available on Kaggle, but each of them had its own limitations. Finally, three data sets were analyzed in this project.

PaySim data set

The first one is PaySim data set, which aims to address the issue of a limited number of publicly available financial transactions' data sets. It is a synthetic data set which was generated using PaySim data simulator of mobile transactions using agent-based modeling. The generative process was implemented using a sample of real-world transaction logs produced by assets from a particular region, which precise location was not disclosed [3]. Due to inconsistencies and synthetic nature of the data set it serves as a starting point of the analysis, mostly to determine what are the characteristics of an artificial data set.

IEEE-CIS Fraud Detection data set

The second one is the IEEE-CIS Fraud Detection data set, and it was published by IEEE Computational Intelligence Society (IEEE-CIS) on Kaggle, in cooperation with Vesta – a company which provided the data set. At the time of writing this thesis, it is used for a competition which aims to improve the current techniques for fraud detection [5]. It is suitable for tabular analysis, as it is record-oriented, providing various details on the transactions. The data were anonymized so that the identity of the entities cannot be restored. This is the biggest obstacle for the

process of extracting nodes, however, it can be overcome using another approach. The analysis was performed on the training set, due to obvious reason of the presence of labels for each transaction. The training set was split into two files: one named *train_identity.csv* and the other *train_transaction.csv*.

Elliptic data set

The last one is the Elliptic data set. It maps Bitcoin transactions to entities belonging to licit categories characterized by activity such as exchanges, providing wallets, mining, licit services, etc. and those which belong to illicit ones involved in an activity such as scams, malware propagation, terrorist organizations, ransomware groups, Ponzi schemes, etc. Elliptic data set represents a transactions' graph, and the data were collected from a Bitcoin blockchain [6].

2.1 Data exploration

As the data sets were provided in comma-separated values format (CSV), the data exploration process was performed using relational approaches. Characteristics such as the number of nodes, number of edges, the number of missing values, the range of time steps, the number of values in each of the transaction categories related to fraud and other details were studied prior to data preparation. Obtaining this information allows for proposing a coherent plan for creating a graph database and focuses the analysis on meaningful attributes.

2.1.1 Exploratory analysis of the PaySim data set

PaySim simulator was used to generate a relatively big data set and it consists of transaction-oriented records. The data set is available on Kaggle [7]. It contains 6,362,620 rows and 11 attributes. These 11 features correspond to: the time step (step), the type of financial activity (type), the amount (amount), the source account's signature (nameOrig), the balance for the source account before the transaction (oldbalanceOrg), the balance for the source account after the transaction (newbalanceOrg), the destination account's signature (nameDest), the balance for the destination account before the transaction (oldbalanceDest), the balance for the destination account after the transaction (newbalanceDest), an attribute for indicating if the transaction was fraudulent (isFraud) and if it was flagged as fraudulent one (isFlaggedFraud). The records correspond to the transactions recorded for a time period of 744 hours, or approximately 31 days [7]. As it is a synthetic data set, the records were populated with the data in such a way that there are no missing values in any of the attributes.

The graph structure of the PaySim dataset

One of the most important aspects of further graph analysis are vertices. In case of this data set no entity signatures for the nodes' IDs had to be extracted, as the CSV file contained an edge-list. Although, despite availability of the data on nodes and relationships between them, it did not ensure efficiency and suitability of a graph analysis for financial flow among nodes in a financial network.

Suppose that the records are unique transactions, made by unique entities which do not appear more than one time for the recorded time period. If that was the case, the data set contained $2n$ nodes, where n represents the number of rows. That would be 12,725,240 unique nodes in total, constituting a bipartite graph with

source accounts set and destination accounts set both of a cardinality 6,362,620. From the perspective of graph analysis for fraudulent activity, if the financial flow was recorded only between pairs of nodes from two separate, homogeneous sets and the directed graph contained 6,362,620 weakly connected components, that would be a highly unwanted case. It would make nearly all of graph's properties meaningless from the perspective of this research, as they would be record-specific and they would show no general pattern across the entire network. Fortunately, in the case of the PaySim data set, the records contain duplicates for entities which interacted with the others, which allows for examining the financial flow and for obtaining meaningful properties of groups of nodes actively managing their funds. The actual number of nodes is 9,073,900 and the number of edges is equal to the number of records in the CSV file *paysim_dataset.csv*: 6,362,620. There are two categories of transactions encoded in a binary format – 1 for fraudulent transactions and 0 for non-fraudulent transactions. There are 6,354,407 non-fraudulent ones and 8213 fraudulent ones, constituting approximately 0.0013 of the number of all records in the data set.

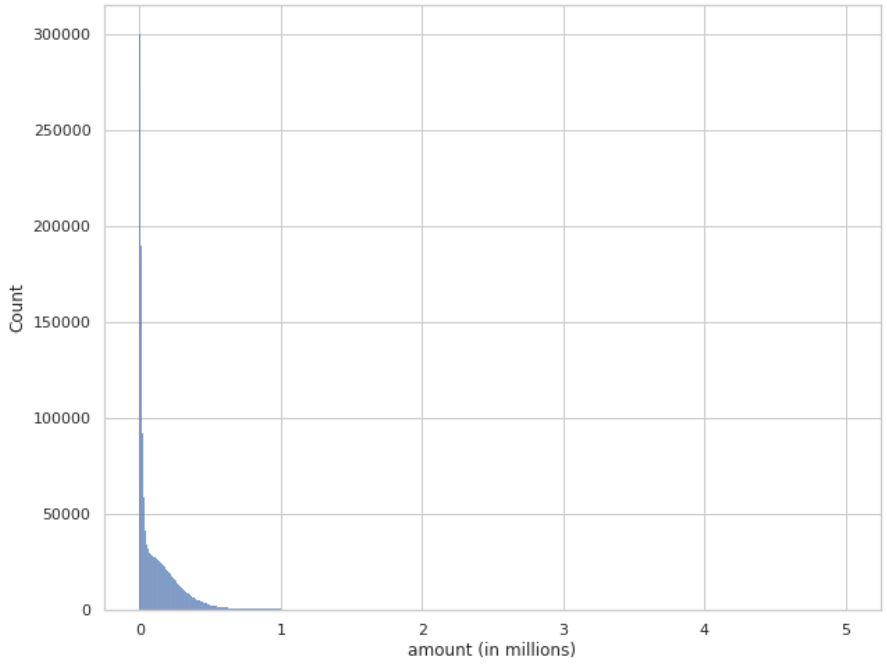
One of the interesting aspects to focus on could be the amounts transferred. The maximum value for amounts is 92,445,516.64, the recorded minimum is 0, the mean and median are 179,861.9 and 74,871.9 respectively, the standard deviation is 603,858.23. Therefore, we can conclude that the distribution of the amounts in the data set is positively skewed, with quite a considerable spread of the values. These measures are listed in Table 1.

Table 1: Descriptive statistics for the PaySim data set amounts' distribution.

measure	value
minimum	0.0
maximum	92,445,516.64
mean	179,861.9
median	74,871.9
standard deviation	603,858.23

Considering these characteristics, the data set was split on the threshold of 5,000,000, which resulted in more than 6,300,000 records below that amount and just 11,515 records above or equal to it. Based on this finding, we can plot a histogram for the group below that threshold, so that we can see a general shape of the distribution for the majority group, constituting more than 99% of the records. Obviously, if we plotted the histogram including the high-amounts group, it would have a long tail reaching up to 92,445,516.64 with very low frequencies for these observations. Such a focus at the majority group stands as an alternative to a log-log plot which may lead to an incorrect interpretation of the pattern. The histogram is shown in Figure 1.

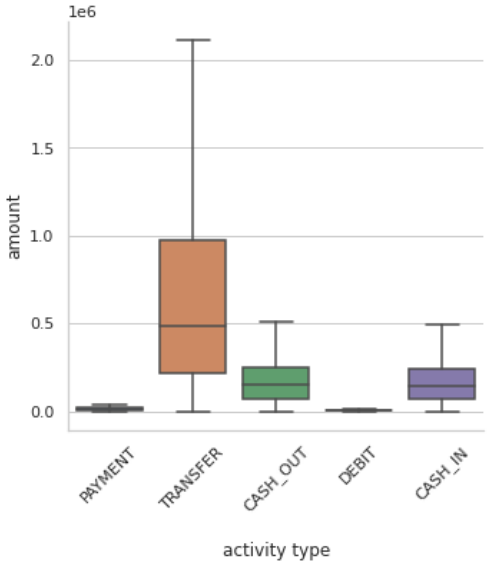
Figure 1: The distribution of the PaySim data set amount below 5,000,000.



A histogram showing the distribution of amounts for a group of observations which transferred less than 5,000,000. Such a stochastic approach can lead to filtering a majority group, tacking the problem of a highly-skewed distribution and a long tail of the outliers. In this plot, it is quite evident that most entities manage funds of less than 1 million in this financial network.

Another interesting aspect of this data set is to study the amount characteristics in different financial activity categories (payments, transfers, cash-outs, debits, cash-in). PAYMENT and DEBIT categories contain observations with the smallest interquartile ranges (IQR) compared to other activity types. There are similar distributions of amounts for CASH_OUT and CASH_IN categories and the biggest spread of values was noted for observations from TRANSFER category. Boxplots for these categories are shown in Figure 2.

Figure 2: Boxplots for amounts in different categories.

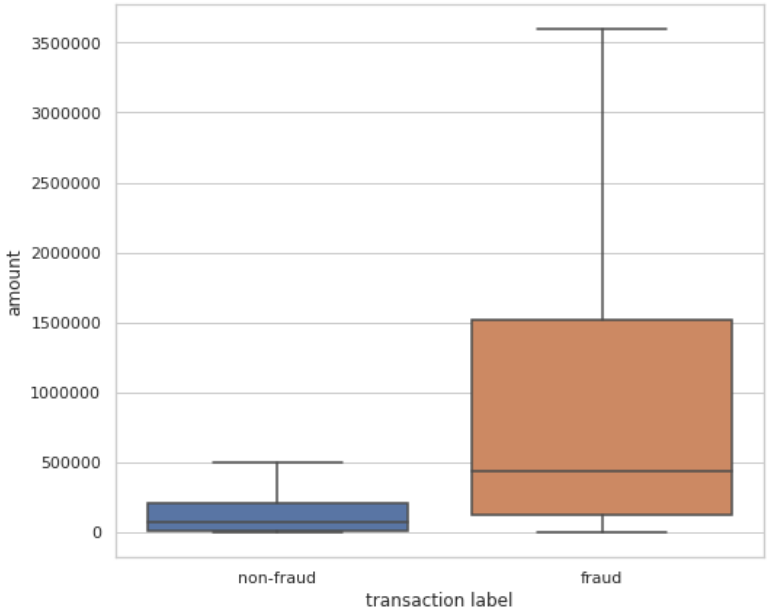


The boxplots show the details for the distributions among different financial activity categories. It is quite evident that whereas PAYMENT and DEBIT observations are characterized by small amounts, TRANSFER records have a wide range of amounts with the upper boundary of more than 2 million. CASH_IN and CASH_OUT have similar distributions for the amounts which seem quite consistent.

Fraudulent transactions and amounts transferred

When boxplots are created for the amounts of fraudulent and non-fraudulent activities, the distributions seem unalike when the outliers exceeding the boxplot's upper boundary value are not plotted. This is shown in Figure 3.

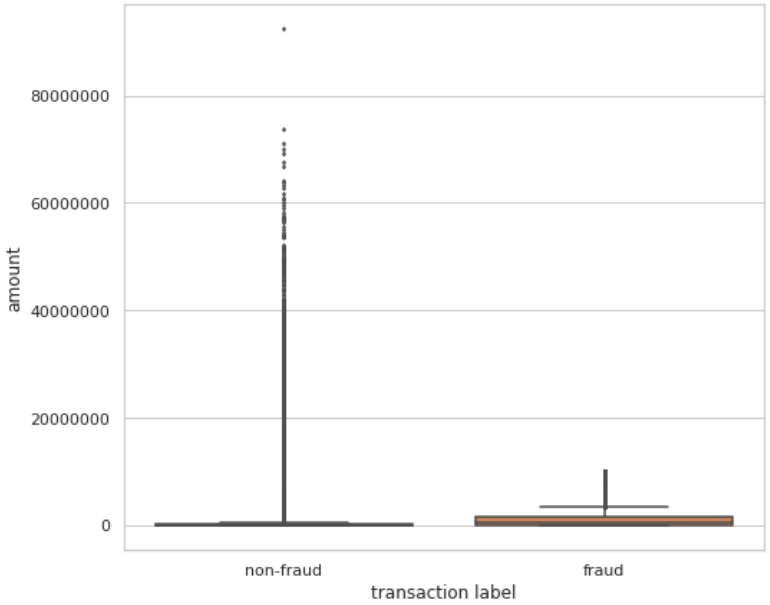
Figure 3: The distributions of amounts in non-fraud versus fraud categories.



Not plotting fliers in boxplots can lead to a wrong assumption on significantly different distributions for fraud and non-fraud categories.

Such a visualization indicates that non-fraudulent transactions have a much smaller range of amounts and that values exceeding the limit of 500,000 can be flagged as fraud. *Nota bene*, according to the obtained boxplots, it is approximately the median of the amounts related to fraudulent activity. Plotting fliers, however, it is quite evident that whereas fraudulent activity contains outliers exceeding the upper boundary of the IQR, they are located within a range of around 3,500,000 to 10,000,000. A plot which includes the fliers is presented in Figure 4.

Figure 4: The distributions of amounts in non-fraud versus fraud categories with outliers.



The fliers plotted for these boxplots show that practically, it would be almost impossible to flag potential fraudulent transactions based on a simple mechanism relying on an amount-based filter.

As seen in Figure 4, the non-fraudulent activity is characterized by many more outliers with a much greater spread for their values. In order to obtain the exact number of records which can be considered outliers, an interquartile range can be used again. It is because normalization such as Z-score normalization and labeling values located 3 standard deviations from the mean as anomalies is an established technique in case of normally distributed values, when 68%, 95%, 99.7% of the data lay respectively within a range of 1, 2 and 3 standard deviations from the mean [8]. In case of this attribute's values, such a pattern would not be found, therefore a method which does not require these assumptions is needed. Just as with the visualizations presented before, one possible solution is to use a boxplot technique proposed by Tukey in 1977, which can be applied to data of skewed distributions. It utilizes the concept of interquartile range (IQR) and fences or

upper and lower boundaries which reach the distance of 1.5 IQR below the first quartile (Q1) and 1.5 IQR above the third quartile (Q3) [8]. The interquartile range (IQR) can be computed using the following formula (Equation 1).

$$\text{IQR} = \text{Q3} - \text{Q1} \tag{1}$$

Where Q3 is the third quartile and Q1 is the first quartile. The lower boundary (L) can be obtained subtracting the IQR value multiplied by 1.5 from the Q1 value (Equation 2) and the upper boundary (U) by adding the IQR value multiplied by 1.5 to the Q3 value (Equation 3).

$$\text{L} = \text{Q1} - 1.5 \times \text{IQR} \tag{2}$$

$$\text{U} = \text{Q3} + 1.5 \times \text{IQR} \tag{3}$$

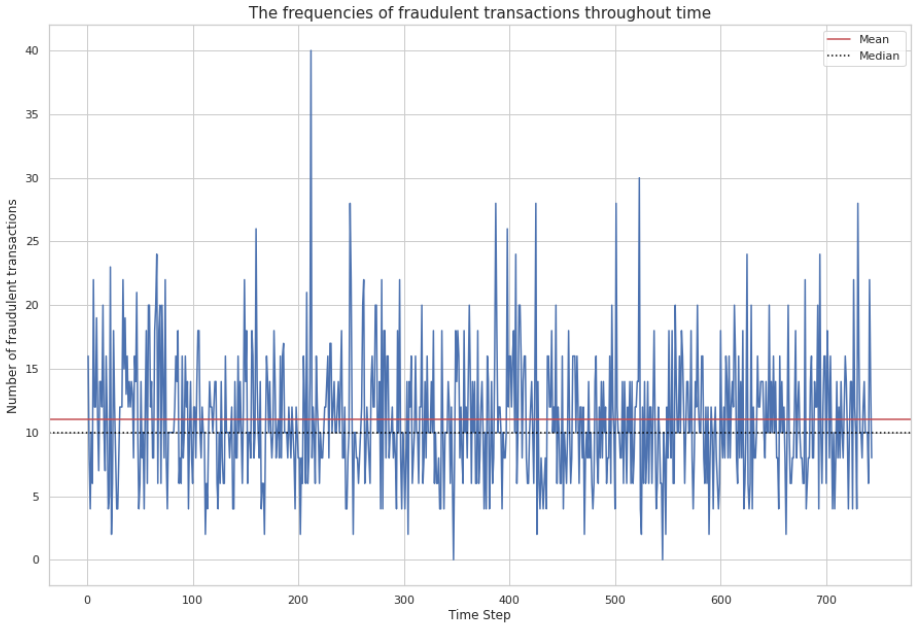
In the context of the entire data set, there are 338,078 observations which are anomalies located above the upper boundary and 0 observations laying below the lower boundary. For the fraudulent activity this is 998 observations and for non-fraudulent activity: 335,347 observations. Fraudulent activity was recorded for TRANSFER and CASH_OUT types of transactions, with 4097 and 4116 observations of illicit characteristic respectively. The other categories did not contain any fraudulent actions. These findings are presented in Table 2.

Table 2: Fraudulent and non-fraudulent transactions in each financial activity category.

type	isFraud	count
CASH_IN	0	1399284
CASH_OUT	0	2233384
CASH_OUT	1	4116
DEBIT	0	41432
PAYMENT	0	2151495
TRANSFER	0	528812
TRANSFER	1	4097

An interesting facet would be to discover that fraudulent activity is characterized by some temporal pattern. Unfortunately, the analysis of fraudulent activity throughout time cannot confirm the existence of any particular pattern which could lead to conclusions that in the context of the entire data set, the temporal component is significant factor in terms of further analysis. It is shown in Figure 5.

Figure 5: The number of fraudulent activities in each time step.



The visualization of the number of fraudulent activities in each time step does not prove that any temporal pattern exist for the illicit activity. Mean and median number of fraudulent activity are plotted as red and dashed blue vertical lines respectively.

The key findings of the exploratory analysis are that it is possible to create a graph which can show financial flow among different vertices and which would contain over 9 million nodes and 6 million relationships. The data set is imbalanced and contains less than 1% of observations classified as an illicit financial activity. The amounts managed by entities constitute a highly positively skewed distribution of observations and at a statistical level it is possible to distinguish certain patterns among fraudulent and non-fraudulent distributions. The median amount for fraudulent transactions was 441,423.44 and for non-fraudulent activity: 74,684.72. Nevertheless, the number of outliers makes it quite difficult to detect fraud from a practical perspective, as among 338,078 anomalies, 998 were fraudulent. An illicit

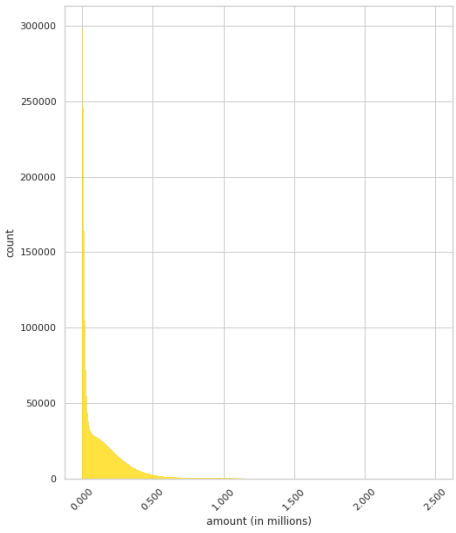
activity was detected in TRANSFER and CASH-OUT categories, whereas other categories contained observations of lawful activities only.

Limitations

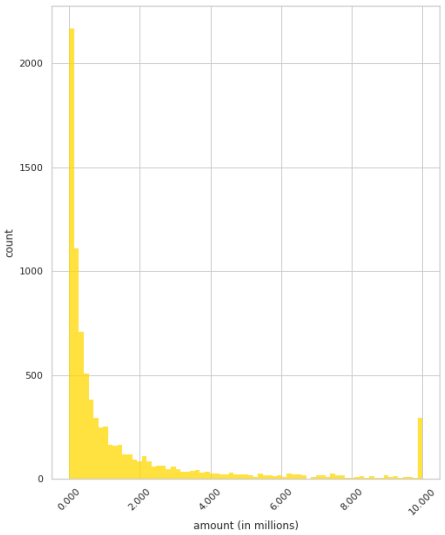
The further data analysis of the PaySim data set was focused on a limited number of records, due to issues related to the amount of random-access memory (RAM) required to perform computations on the entire graph which would include all timesteps. Therefore, the graph creation was based on a sample of the data set for time steps from 1 to 3 inclusively. Based on the results obtained in the exploratory analysis, the sample can be considered representative for the whole data set. The fraction of fraudulent transactions in the original data set was 0.0013, in case of the sample of time steps 1-3 it is 0.0066. The distributions of the amounts for financial activity are highly positively skewed for both data sets. It is shown in Figure 6 and can be compared for the sample distributions in Figure 7.

Figure 6: The distribution of amounts for the entire data set in non-fraudulent and fraudulent class.

(a) Amounts: non-fraudulent class



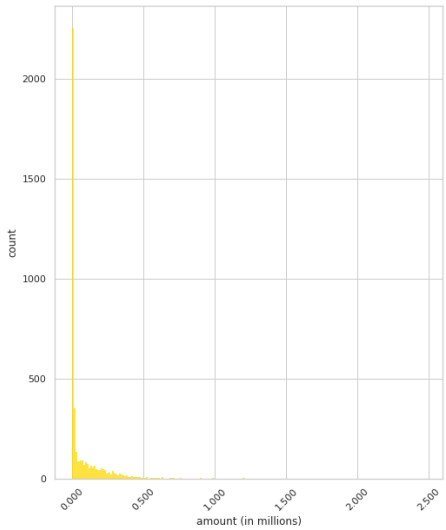
(b) Amounts: fraudulent class



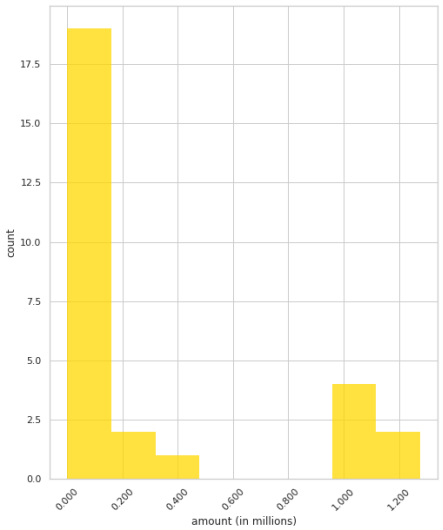
Both classes are characterized by positively-skewed distributions. The fraudulent class distribution contains an anomaly at the end of the distribution's tail with more than 300 activities for amounts greater than 10 million.

Figure 7: The distribution of amounts for the sample of time steps from 1 to 3 in non-fraudulent and fraudulent class.

(a) Amounts: non-fraudulent class



(b) Amounts: fraudulent class



Just as in Figure 6, both classes are characterized by positively-skewed distributions. Again, for the fraudulent class of observations, the end of the distribution's tail contains higher number of activities for the amounts of more than 1 million.

Moreover, the fraudulent activity was related to the same activity categories in both time periods. This can be seen comparing the entries in Table 3 for the entire data set with the records of Table 4.

Table 3: Illicit and lawful financial activity in each category - the entire data set.

type	isFraud	count
CASH_IN	0	1399284
CASH_OUT	0	2233384
CASH_OUT	1	4116
DEBIT	0	41432
PAYMENT	0	2151495
TRANSFER	0	528812
TRANSFER	1	4097

Fraud was present only in CASH_OUT and TRANSFER categories. For the CASH_OUT it constituted less than 0.2% of operations and for the transfers, less than 0.8%.

Table 4: Illicit and lawful financial activity in each category - the sample of time steps from 1 to 3.

type	isFraud	count
CASH_IN	0	854
CASH_OUT	0	529
CASH_OUT	1	15
DEBIT	0	244
PAYMENT	0	2240
TRANSFER	0	379
TRANSFER	1	13

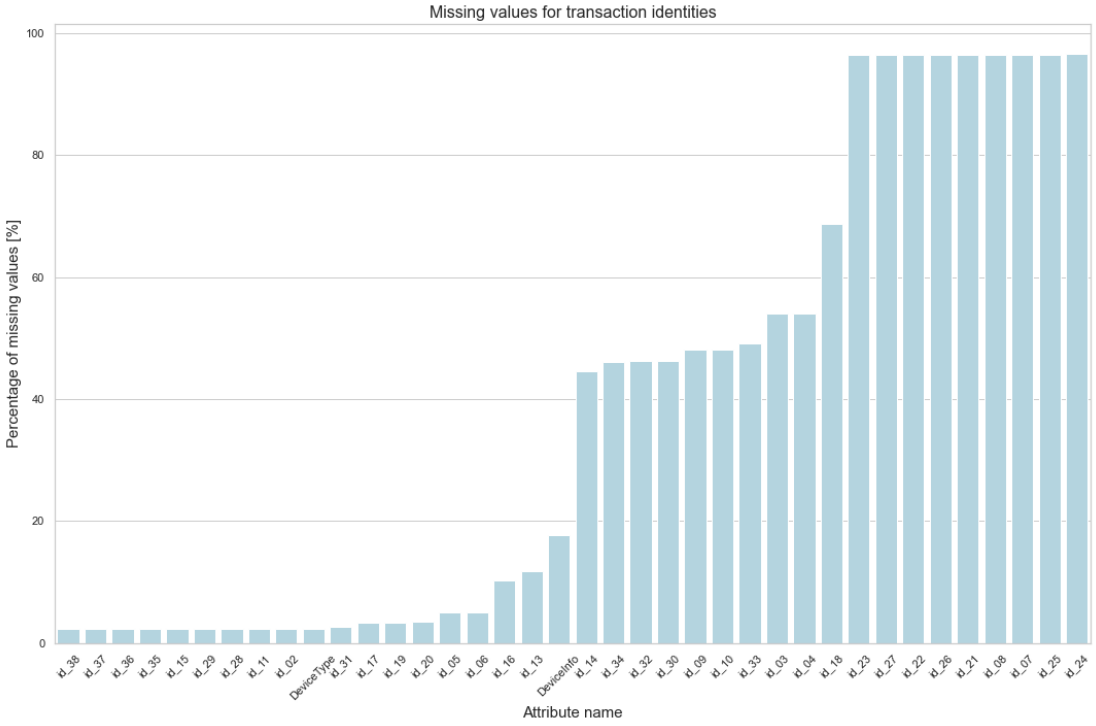
Similarly to Table 3, fraud was detected only in CASH_OUT and TRANSFER categories. For the CASH_OUT it constituted less than 3% of operations and for the transfers, about 3.3%.

Considering these aspects, it is acceptable to study PaySim data set from the perspective of the selected sample. The graph of that sample consists of 6876 nodes and 4274 arcs.

2.1.2 Exploratory analysis of the IEEE-CIS data set

Just as mentioned before, the data set is split into two files. One corresponds to the records related to transactions features – such as transaction ID, information about the device used for making a transaction and numerous other attributes for which no description was provided and which were anonymized. The data frame of that CSV file contains 144,233 records and 41 columns. The other CSV file contains data on the transactions themselves such as the amount, label - if the transaction is fraudulent or not, the time step, card information, asset information and payment service type. Obviously, sensitive information such as card details or addresses were anonymized. It contains 590,540 rows and 394 columns. Most of the attributes in the data set had to be discarded, as no descriptions were provided for most of them. The dataset contained many missing values, especially for the transaction identity data frame. It is shown in Figure 8.

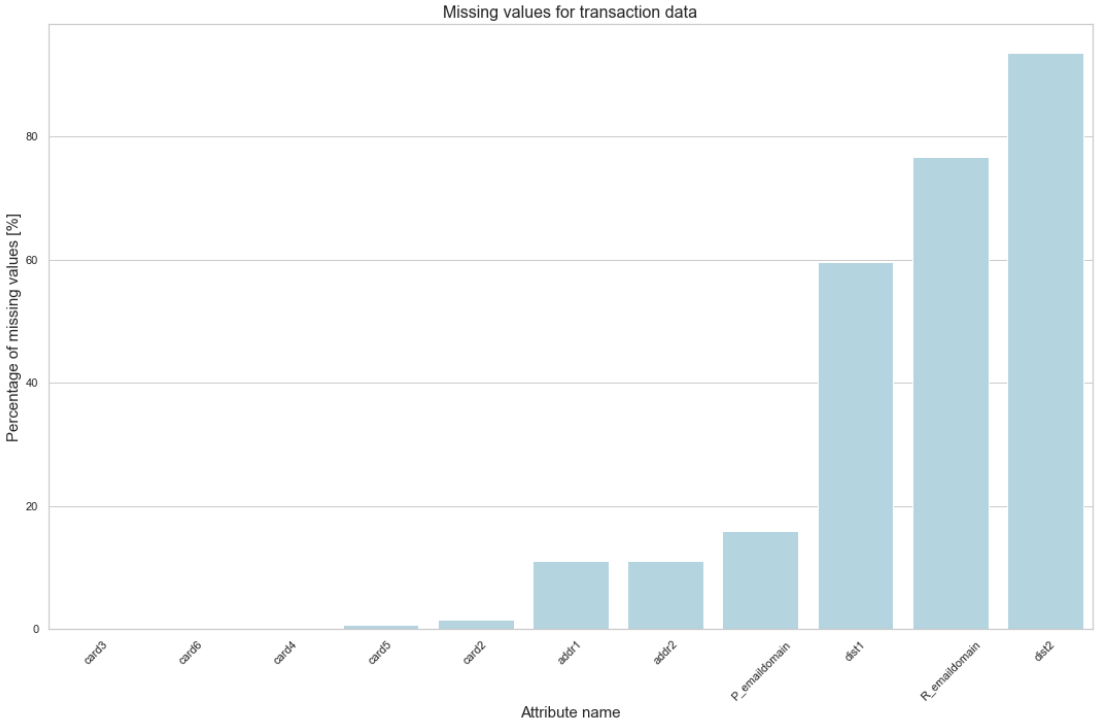
Figure 8: The percentages of missing values in attributes of transaction identities CSV file.



The plot shown that data missingness in transaction identities data set can be a potential problem for including some of the identity attributes. For nine of them, the missingness achieves the level of more than 90% and for ten of them - more than 40%.

In the case of another data frame related to the transaction data, there were fewer missing values, however, among attributes which constitute a considerable part of the data set. In result, potential problems for entity reconstruction in the process of graph preparation may occur. This is shown in Figure 9.

Figure 9: The number of missing values in attributes of transaction data CSV file.



Again, some attributes contain more than 60% of missing values, however, despite the concerning missingness among three variables, in case of this data frame, the quality of the data set in terms of missing values, can be considered somehow better than for the data frame referred in Figure 8.

Considering the issue of missingness presented in Figures 8 and 9, a solution avoiding imputation was chosen. It is because the most important factor of this data set's analysis is retrieving nodes' signatures in order to construct the financial graph. Moreover, most attributes do not contain descriptions which could lead to meaningful inferential analysis results. The data set was narrowed to the following attributes, with all records containing missing values being dropped. For the transaction data these were: *TransactionID*, *isFraud*, *TransactionDT*, *TransactionAmt*, *ProductCD*, *card1*, *card2*, *card3*, *card4*, *card5*, *card6*, *addr1*, *addr2, dist1, dist2, P_emaildomain, R_emaildomain* and for transaction identities the selected

columns were: *TransactionID, DeviceType, DeviceInfo.*

Table 5 presents fraudulent and non-fraudulent transactions distributions in *ProductCD* attribute. It represents financial service types. Table 6 shows the same determinants, but for the *DeviceType* attribute.

Table 5: Fraudulent and non-fraudulent financial activity for different financial service types.

ProductCD	isFraud	count
C	0	54552
C	1	7640
H	0	31337
H	1	1571
R	0	36125
R	1	1423
S	0	10901
S	1	684

The data set contains 4 categories of products (C, H, R, S) which can refer to a financial service and not to a purchased product or service. More specifics were not disclosed. As seen in this table, the product category *C* was characterized by the biggest ratio of fraudulent transactions which constituted more than 12% of all transactions for this service type. Further conclusions could be formed if more details were available on these categories.

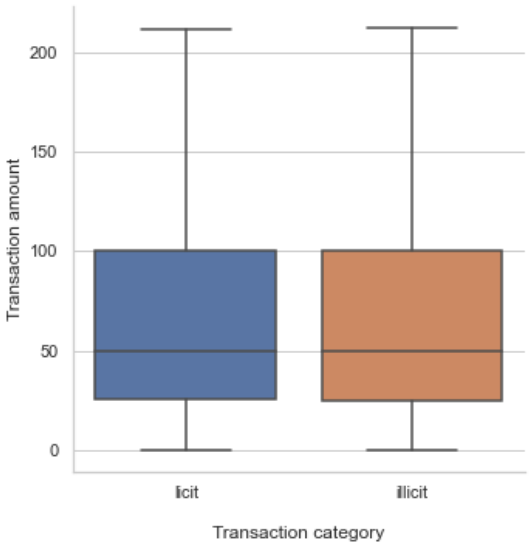
Table 6: Fraudulent and non-fraudulent financial activity for desktop and mobile devices.

DeviceType	isFraud	count
desktop	0	79611
desktop	1	5554
mobile	0	49988
mobile	1	5657

For the transactions made via desktop devices the fraudulent ones constituted 6.52% of them, whereas in the case of mobile devices this was more than 10% of transactions.

The distribution of amounts for fraudulent and non-fraudulent transactions were the same, thus no assumptions about thresholds could be made. This is shown in Figure 10.

Figure 10: The distribution of amount for licit and illicit financial activities.



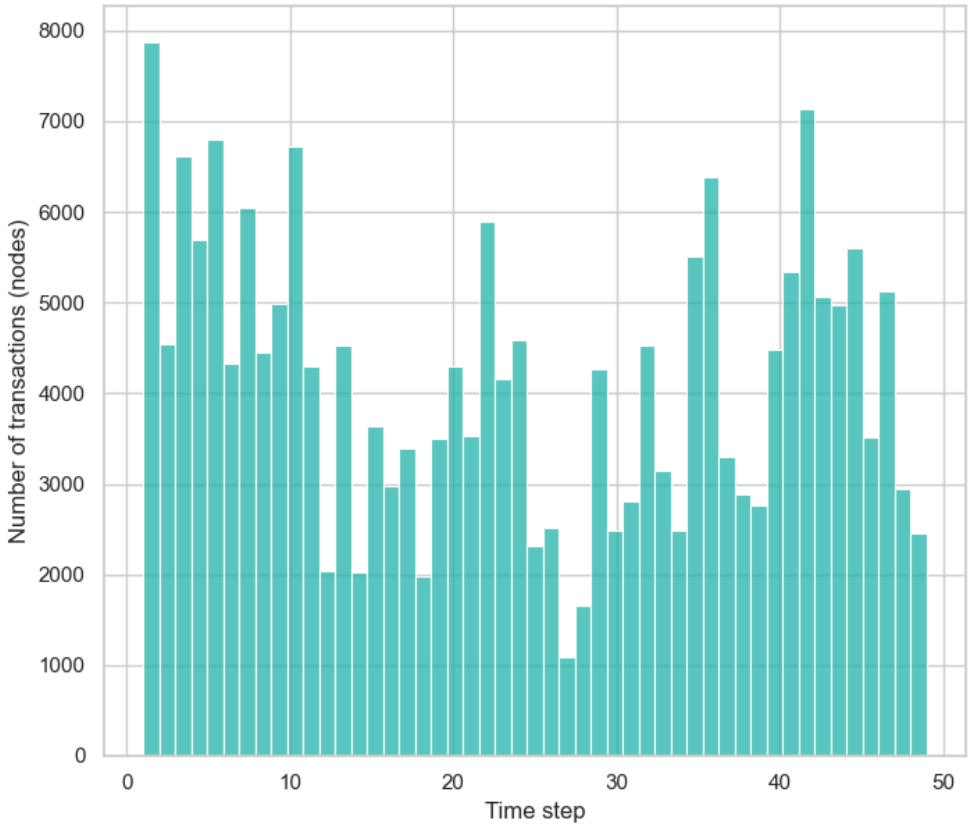
No amount-threshold can be set to differentiate between potential fraudulent and non-fraudulent transactions.

2.1.3 Exploratory analysis of the Elliptic data set

The data set contains 203,769 nodes and 234,355 edges. It was recorded in the range of 49 time steps and between each evenly spaced time step there is an interval of around 2 weeks. The whole data set is split into three files: one for an edge list, one for classes of each transaction, and one for transactions and the time step they belong. Additionally, in the last file, there are 162 numeric attributes, however, due to intellectual property issues, their descriptions could not be provided, therefore there is little use for them in the further inferential analysis [6]. It is worth noting that in fact the data set holds 49 directed graphs. This is because, if initially

a network is constructed from the transaction of all the time steps, there are 49 weakly connected components – each corresponding to its time step. In such a context, the giant component consists of the transactions from the first time step and it contains 7880 nodes. The smallest one is from the time step 27 and contains 1089 nodes. The graph sizes for each time step are shown in Figure 11.

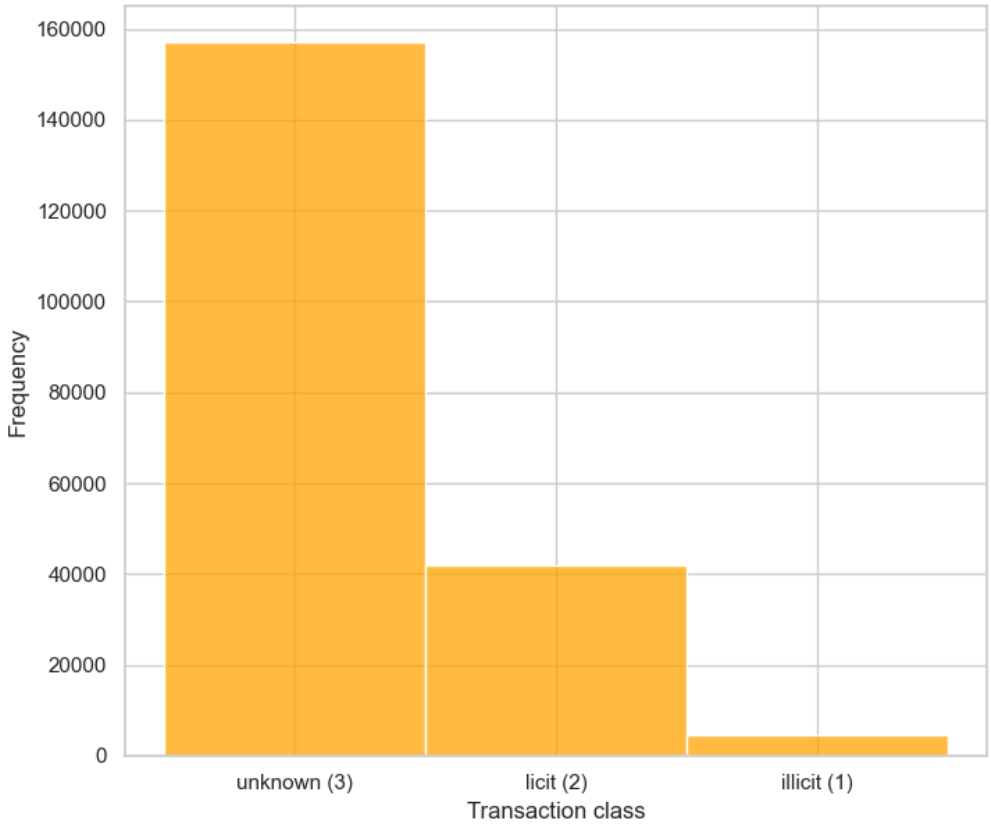
Figure 11: The sizes of the financial network in different time steps.



The graph contained the most nodes and therefore, the activity was the greatest in the first time step.

There are three transaction classes: unknown, licit and illicit. Their shares in the data set are 77.2%, 20.6%, 2.2% respectively. They are presented in Figure 12.

Figure 12: The distribution of transaction classes.

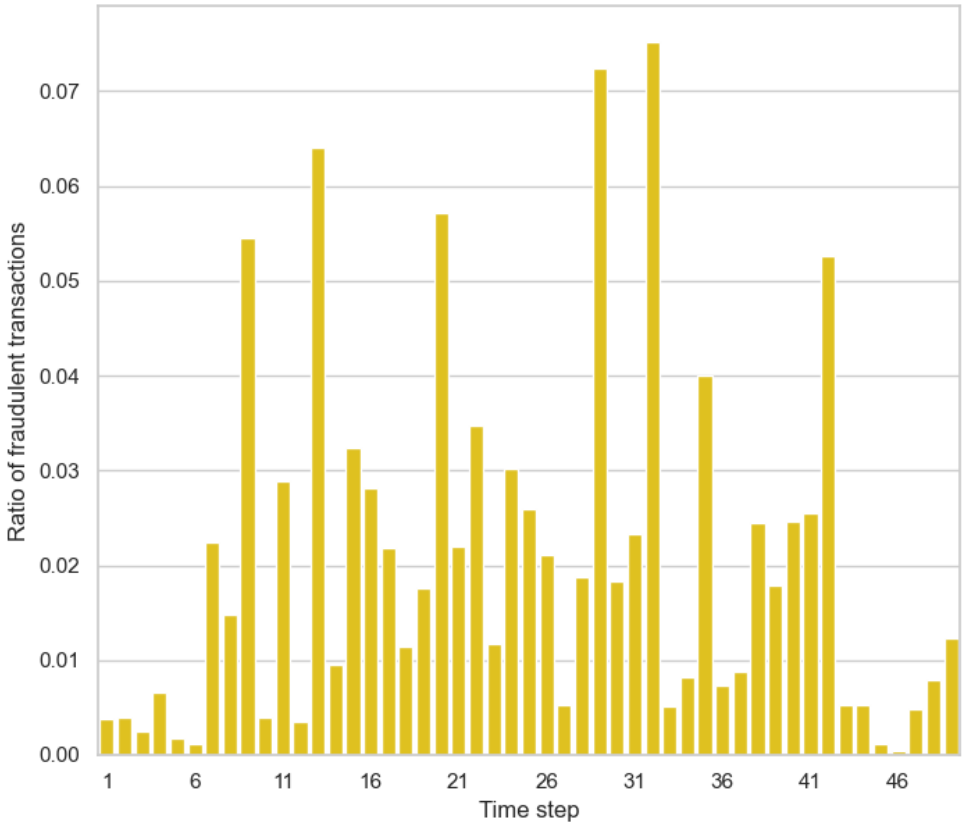


The majority group for transaction classes is the unknown class, which can lead to potential problems due to lack of interpretability for these records.

The biggest number of fraudulent transactions was recorded at the 32nd timestep with 342 illicit transactions which constituted approximately 7.5% of all transactions for that time step. Considering that the component, or rather the graph, contained 3306 nodes at that time step, fraudulent transactions constituted ap-

proximately 10% of all transactions during that period. The ratio of fraudulent transactions to all transactions for each time period is shown in Figure 13.

Figure 13: The ratio of the number of fraudulent transactions to the number of all transactions throughout time.



As seen in the figure, the biggest ratio of fraudulent transactions to all transactions was recorded for the time step number 32.

From the standard perspective of data set cleanness, it could be considered clean. There are no empty records, containing N/A or NaN values. Although, the issue is quite subtle and for the purpose of further analysis – mostly not to interfere

with the graph creation process – it will be neglected. That issue is the presence of transaction entities labeled as unknown, which as mentioned before, constitute approximately 80% of the data set. The other attributes do not contain missing or undefined values.

2.2 Financial graph preparation

The data were prepared in such a way to be suitable for further steps of the analysis – employing usage of graphs for each of the data sets considered in this study. The data stored in tabular format in CSV files were preprocessed and stored in Neo4j graph databases. Neo4j is a NoSQL database which stores the data of nodes and edges. It was written in Java and Scala and provides ACID-compliant, transactional standard of the database [9]. Before creating a graph database, the data had to be prepared in such a way that the nodes and their properties are organized within one CSV file and the edges and their attributes in another one. Following points present the data preparation process for splitting values related to nodes and edges before populating the database with them:

1. Extract unique nodes from the data set and assign attributes to them
2. Prepare an edge list and attributes which will be assigned to the edges
3. Save the data on nodes to a CSV file: node ID, node label and its other attributes.
4. Save the data on edges and their attributes to a CSV file: the source node ID, other edge attributes, the destination node ID, the type of relationship.
5. Save headers for these data sets in separate CSV files.

A correct format for the nodes header CSV file are the following attributes: *id:ID*, other attributes, *label:LABEL*. A correct format for edges header CSV file is as follows: *:START_ID*, other attributes, *:END_ID, :TYPE*. Each attribute name in the header file is separated from its declared data type with a colon, thus if cost attribute is provided for a relationship, it can be represented as *cost:int* if it is of integer type.

The resulting CSV files can then be used to create a graph in Neo4j using admin import – a tool which can be used for loading large amounts of data from CSV files into an unused non-existing database [10]. Upon successful import, the database can be used to perform computation on the graph as soon as it is started. The DBMS uses bolt protocol, a lightweight protocol used for databases, which by default operates on the port number 7687 [11].

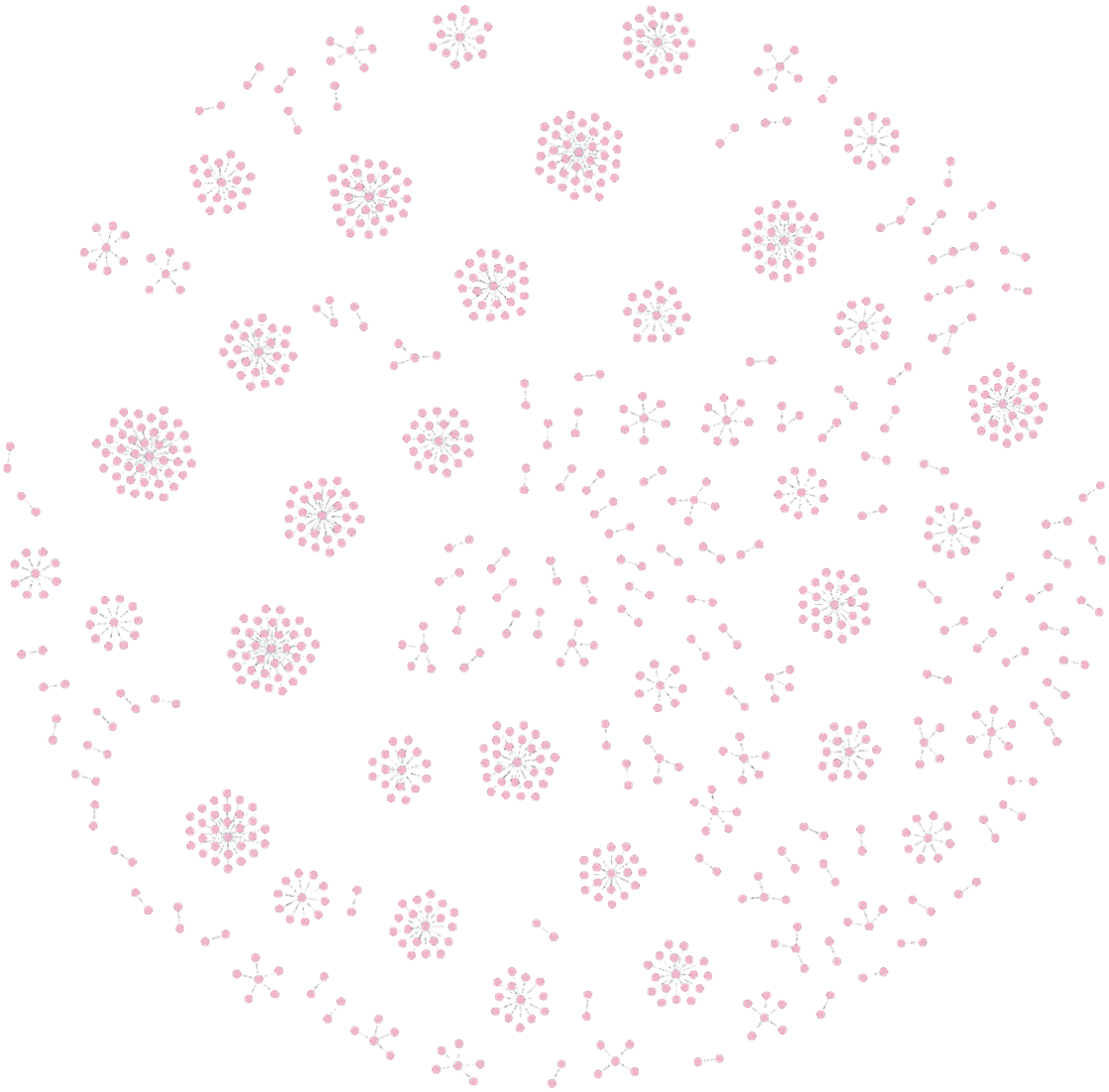
Further analysis is based on the graph analysis methods and it is supported by the relational approach using the original CSV data sets' files. Moreover, as a form of an alternative, for some computations, not only Neo4j graphs, but also Networkx graphs were used. Networkx is a Python library for creation, manipulation, and study of graphs [12].

2.2.1 The PaySim data set graph

As it was described before, the data set is suitable for graph analysis, however, only a sample of the entire data set could be considered. A class attribute indicating if a node was involved in fraudulent activity had to be assigned to each vertex, as the analysis of centrality metrics employed studying the characteristics of two distinctive sets of nodes: fraudulent versus non-fraudulent ones. Figures presented below show the organization of CSV files which were used to import the data to Neo4j database. After successful import of the data, the graph consisted of 6876

nodes and 4274 relationships. A visualization of 1000 nodes for this network is shown in Figure 14.

Figure 14: A visualization of 1000 nodes and their edges of the PaySim graph.



As seen in the figure, the graph for PaySim data set is relatively sparse. There are numerous individual interactions between two nodes and a few structures of nodes having high in-degree score with their neighbours having just one out-going edge.

2.2.2 The IEEE-CIS data set graph

Reconstructing nodes

The IEEE-CIS data set was prepared in such a way to allow for a tabular analysis, however, it is not straightforwardly suitable for a graph analysis. The key issue was that there were no data allowing for recreation of an edge list representing transaction flow in the financial network. Each row contains data on entities making transactions and on features related to the transactions themselves. The first step of data preparation so that it can be analyzed from a financial graph's perspective, was to attempt to reconstruct unique entities, so that source nodes can be obtained. It was performed using customer's card attributes placed in the data set's columns *card1, card2, card3, card4, card5, card6*. These are features which correspond to the customer's details on their payment cards. They can be considered to be precise enough to reconstruct particular entities, at the same time keeping an anonymous format. Values from these six attributes allowed for retrieving 8404 unique account signatures. They were treated as nodes representing entities making transactions.

Representing the data as customer-product bipartite graph

Another fundamental issue not allowing for an unchallenging shift from tabular data analysis to graph analysis was the lack of destination nodes for the transactions. One of the possible ideas was to use a bipartite graph, containing two disjoint, but internally homogeneous node set, as usually represented in customer-merchant financial networks [13]. In this case, the customers would constitute the nodes, whose signatures were obtained in the previous step, and the merchants, due to a lack of destination account characteristics, would be replaced with a type of product or service which was purchased by the customer. This approach allows

for identifying odd transactions, which edges lead to completely different products or services' category and can be flagged immediately. However, later it became clear that the products in the data refer to financial services, and not to goods or services that were paid for through the corresponding transactions. Having just five transaction categories as a service feature did not introduce relevant dependencies among vertices for the purpose of graph creation.

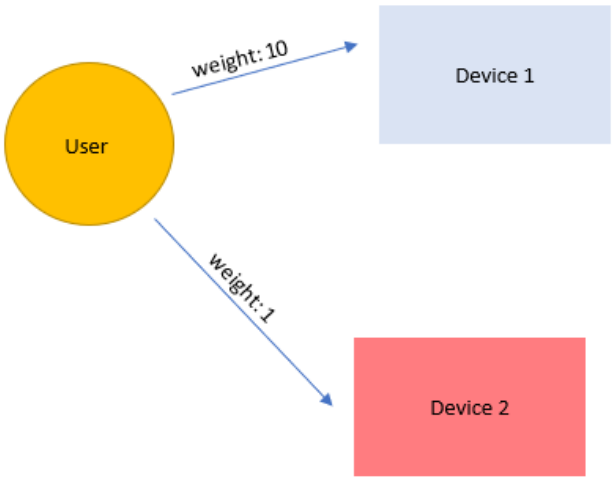
Representing the data as customer-device bipartite graph

An alternative idea was to map individual accounts to devices they use. Such a solution mimics a technique which could be implemented if sensitive details were provided – such as physical device addresses (MAC) or Internet Protocol (IP) addresses related to the activity of the customers. In this case the feature of interest was an attribute *DeviceInfo* which provided certain details regarding the system which was used to make a transaction. This way, a bipartite graph represents customer-device relationship. If a customer uses a device frequently, it can be considered a trusted device, however, in case some transactions are performed from a rarely used device, from which only one or few transactions were made, the activity can be flagged as suspicious – indicating a potential fraud. A similar method regarding the devices was introduced by Google a few years ago to protect its users. Google services' authentication mechanism requires an extra step constituting a two-factor authentication mechanism upon a logon from a new device. It is expected from the user to accept a notification on their mobile in case a user logs in from an unknown device in order to prevent the attackers from performing a successful account hijacking [14].

The Figure 15 represents two relationships of a user and the devices they used for making transactions. One relationship has a weight assigned of a value 10 indicating a strong relationship – most probably that would be their primary

device. The second edge has a weight of 1, indicating a weak relationship and possibly fraudulent activity from a system controlled by criminals.

Figure 15: A conceptual interaction of a user and the devices they use for financial transactions.



An example graph consisting of three heterogeneous nodes among which two belong to devices set of nodes and one represents a user node type. The edges are interactions between the user and the devices. Potentially, a suspicious interaction would constitute an outlier with a low weight indicating an infrequent use of a device.

This approach assumes an illicit activity originating from devices of different characteristics, thus it does not address the issue of a full compromise of a trusted device and transferring the money from the hijacked node. Although, from the customer's perspective such a scenario is quite unlikely, as using a banking application or accessing banking services through a browser usually requires a graphical user interface (GUI) session with the attacked device. Such a functionality can be granted by tools allowing to establish a connection to the hijacked device using remote desktop protocol (RDP) or other protocols offering GUI connection. Despite being quite unchallenging to use, logging using RDP requires obtaining user credentials to the system first and it logs out a user which actively operates on

the system during the attack, thus it may lead to suspicions [15]. Moreover, the attackers usually use spoofing or impersonation techniques stealing user's banking credentials and attempting to log in using a device already controlled by them. Extorting sensitive data is based on social engineering attacks and can be considered less challenging than jailbreaking the device and gaining an administrative access to it. *Nota bene*, many banking apps do not detect jailbreaking leaving the users exposed to possible interference with the app's functionality – especially when the device is easy to root due to availability of exploits allowing for privilege escalation [16].

Limitations of the IEEE-CIS data set

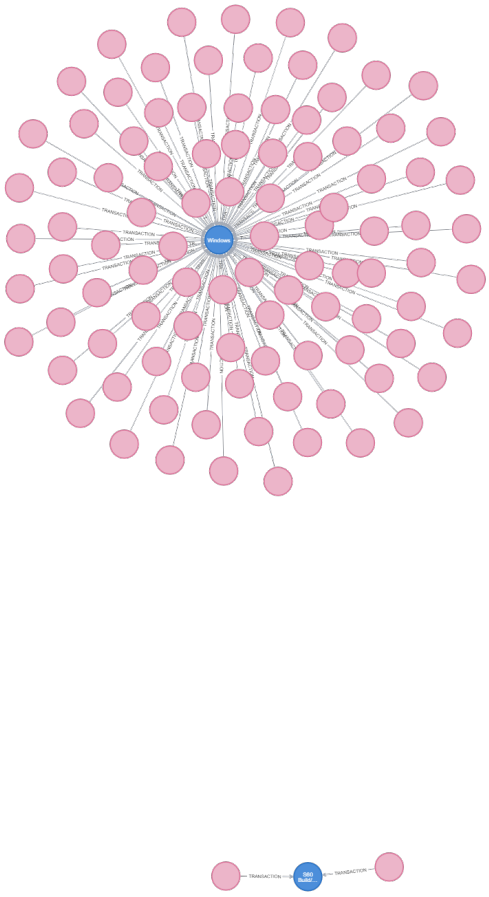
Despite many existing possibilities for anomaly detection mechanisms, the data set does not provide meaningful features allowing for practical reconstruction of individual devices, therefore the database instance created for this particular analysis simulates a general situation in which it is possible to map customers to the devices they use for financial transactions. An attempt to reconstruct nodes using other features of *df_train_identity.csv* dataset (*id_01* to *id_38*) produces too distinctive values, indicating two possibilities. The first one is that the features assigned to each device involved in the transaction are very detailed and do not allow for narrowing the size of the set to a significantly smaller number of devices which could allow for spotting a pattern among users and devices. The second one is that the time range was too short and just few devices in the dataset are duplicates. Therefore, it is a problem of too distinctive device signature which in practice leads to a conclusion that in fact most of the devices in the data set are unique. In such a case, further analysis would be senseless, as barely any trusted devices could be discovered. In result, only the attribute DeviceInfo is considered. It is worth noting that its values are in fact quite noisy, containing agent informa-

tion scraped from User-Agent headers. An example of such a header can look as follows: *Mozilla/5.0 (iPhone; CPU iPhone OS 13_5_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/13.1.1 Mobile/15E148 Safari/604.1* [17]. If the generation of the data set implemented some String pattern matching, noisy values are inevitable. In fact, additional information on device connection allows for retrieving a more precise device identity, thus noise can be considered helpful in this situation – for instance combining information on the operating system used with the version of the browser. Still, it only provides general overview of the software characteristics and does not allow for extracting unique device identity in a reliable way.

Building a graph database for the IEEE-CIS data set

Four CSV file were prepared to load the data into Neo4j database, similarly as in case of PaySim database two CSV header files for nodes and edges data and two CSV files containing the data for nodes and edges. A visualization of 100 relationships in the constructed graph is shown in Figure 16.

Figure 16: A visualization of IEEE-CIS graph structure for a sample of 100 relationships.



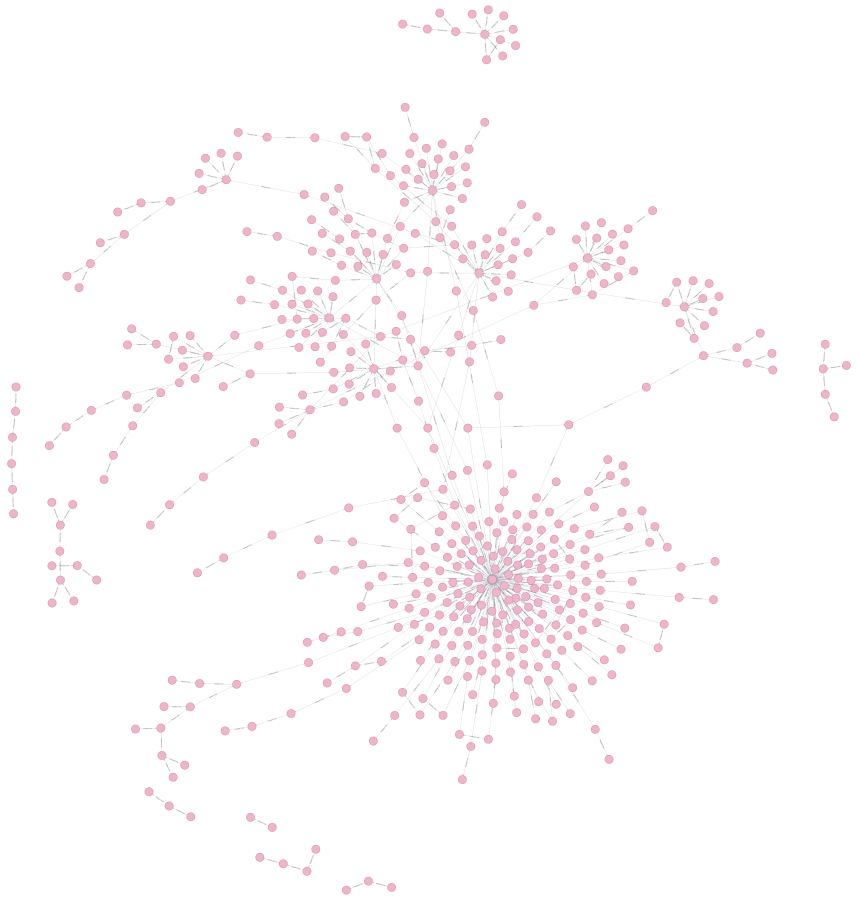
The nodes shown in blue are device nodes – upper one being Windows OS, bottom one representing a categorical data related to one of Android builds. The pink nodes are financial account nodes and they are connected to devices by weighted arcs.

2.2.3 The Elliptic data set graph

Considering that the data set is very clean, the only operations at the initial phase of data preparation were to discard all of the uninterpretable attributes and to replace unknown values in the transaction class attribute with numeric 3. Thereafter, the data set contained three classes with the following values: 1, 2 and 3, representing illicit, licit, and unknown transaction entities respectively. Following

that operation, the data type for the class value was optimized, converting the data type of the array's elements to 8-bit unsigned integers. The data set contained initially three files: one containing data of nodes and their features, another with an edge list and the third one with classes of nodes in the graph. A module responsible for preparing new CSV files which will be used to populate the graph database with data, was written in such a way to allow for extracting nodes, edges and their properties only for a specified time step. Such an approach allows for quick creation of a new database containing a weakly connected graph from a particular time step. This functionality was merged with the ability to quickly build a base for constructing a disconnected graph, however, one that contains the data from all the recorded time steps. A sample considered in the further parts of the analysis was the data from the first time step. A visualization of 1000 nodes for this graph is shown in Figure 17.

Figure 17: A visualization of Elliptic graph structure for a sample of 1000 nodes.



Compared to the PaySim data set graph, this network is denser. The financial flow is characterized by many interconnections between various graph's partitions.

3 Methods

3.1 Graph's large-scale structure

The methods employed in this research are mostly graph-oriented. The large-scale structure of the network is studied in terms of the following characteristics: the type of the graph – if it is a directed or undirected one, a number of nodes, a number

of edges, node's mean degree, a fraction of the size of the largest component to the size of the entire graph, a mean of the shortest paths, an exponent of a power law for its degree distribution, and a mean clustering coefficient. Having the number of nodes and the number of edges, a mean degree of nodes can be computed (Equation 4).

$$\mathrm{c} = \frac{2\mathrm{m}}{\mathrm{n}} \tag{4}$$

Where m is the number of edges and n is the number of nodes. Moreover, it is possible to compute the density, also known as connectance (Equation 5).

$$\rho = \frac{m}{\binom{n}{2}} = \frac{2m}{n\,(n-1)} = \frac{c}{n-1} \tag{5}$$

Where m is the number of edges, n is the number of nodes and c is the mean degree. The maximum number of possible edges in a graph is $\frac{1}{2}n(n-1)$, so the density of a graph indicates a fraction of existing edges to all possible edges. A graph in which $\rho \to 0$ is sparse [18].

Directed graphs have strongly and weakly connected components. A strongly connected component is a part of a directed graph in which for each pair of vertices u and v, there is a path from u to v and from v to u [19]. In case of a directed graph, a weakly connected component is a concept close to weakly connected components in a undirected graphs. It is such a part of a graph in which a set of nodes is considered connected when for each pair of nodes u and v, there is a directed path either from u to v or from v to u. Usually, in directed networks, the largest component corresponds to a relatively big weakly connected component [18].

A fraction of the number nodes in the largest component to the number of nodes in

the entire graph will serve as a metric for assessing how well connected its structure is.

The clustering coefficient quantifies the number of triangles in a network and can be especially important when studying a social network. The mean clustering coefficient is a probability that two neighbor of a certain node are also neighbors themselves. In social networks, such a situation is called a *closed triad* and to compute the clustering coefficient, the number of closed paths of length of 2 is divided by the number of all paths of length of 2 [18]. In the Equation 6, n_{cp} corresponds to the number of closed paths of length 2 and nap to all paths of length 2.

$$\mathrm{C} = \frac{n_{cp}}{n_{ap}} \tag{6}$$

In the context of this study, that would mean that for instance a payee v of an entity u interacts with another payee w of the same entity u. In case one of them is involved in illicit activity, the whole triad can be flagged as suspicious. In Neo4j it is not possible to compute clustering coefficient for a directed network, thus the assumption is that entities, when interacting in a directional way, have an undirected relationship related for instance to their common interests which cannot be expressed in a form of directed paths in a financial graph. The global clustering coefficient is estimated from local clustering coefficient scores for nodes using mean, a graph has to be projected to an undirected structure upon the start of computations [20].

These properties allow for gaining an overview of the large-scale structure of the graph analyzed and can lead to certain conclusions on similarities between cases analyzed and specific cases of graphs studied in other research projects. It is ad-

ditional information which can help making general assumptions about the characteristics of a graph. They can be compared based on a summary of the characteristics of different networks: biological, social, information and technological ones [18]. Perhaps financial networks can be found to have common values of some metrics with specific instances of these graphs.

3.2 Centrality

Nodes' centrality metrics used in this analysis include degree, PageRank, closeness, betweenness and HITS. Each of them produces a centrality metric emphasizing different aspects of the position of a vertex in the graph.
The degree is the simplest centrality metric which is the number of edges connected to a vertex. In case of a directed network, there is an in-degree and out-degree for each node. In case of financial networks, the in-degree corresponds to financial interactions of other nodes with a node u and the out-degree to financial interactions of a node u with other entities.

3.2.1 PageRank

PageRank was invented by a group of employees of Google as a metric allowing for estimating an importance of web pages. In general, PageRank algorithm uses a concept of a random surfer who follows the links embedded in web pages. Each web page is assigned a real number and the higher it is, the more important a page is. A random surfer traverses the graph visiting the pages at out-going links from pages it visited. By applying an iterative approach to this problem, it visits certain pages more frequently than others due to how the out-going links guide it, incrementing the assigned real number score for the pages visited. The problem of disconnected

components was solved by introducing a parameter for switching position in the network at random, which allows the random surfer to be located in disconnected components despite the lack of out-going links leading to these components [21]. For a financial graph, a high PageRank score for a certain node suggests, that it is an important asset from a perspective of financial flow, as interactions among nodes at some point of making transactions or performing other activity related to finances, lead to that specific node.

3.2.2 Closeness

Closeness is a measure of centrality which tells what the mean distance from a node to other nodes is. For financial transactions, it can tell how far from most entities vertices which are known to be involved in the illicit activity are located.

3.2.3 Betweeness

Betweenness is another centrality measure which captures the extent to which a node lies on paths between other vertices [18]. It can help detecting bridges which connect parts of graphs and are important for instance in some of technological networks, such as the Internet, in which connectivity must be maintained at all times.

3.2.4 HITS

HITS centrality metric has its name derived from hyperlink-induced topic search centrality algorithm. It assigns two scores for a node – hub and authority centrality scores. The former one indicates to how many nodes with high authority

centrality a node points to, whereas the latter one to how many nodes with high hub centrality a node is related to [18]. HITS has a recursive definition for understanding the centrality of a node in which one can conclude that an entity is an apparent hub if it points to apparent authorities and it is clearly an authority if it points to apparent hubs [21]. In case of financial networks, such a centrality metric can be helpful in terms of understanding the hierarchy of entities and to search for fraudulent nodes for the perspective of that hierarchy.
The results of centrality metrics computation, which are listed in section 4. *Explorative Analysis of the Financial Networks*, are evaluated using a goodness of fit analysis by plotting the distributions for licit versus illicit vertices classes.

Mann-Whitney U test

For the centrality scores, a non-parametric method for comparing samples' populations is used. Since the distributions are highly skewed for all metrics, parametric methods such as t-test, Z-test or F-test could not be implemented. The remedy to this problem is use of Mann-Whitney U test which allows for comparing skewed distributions. The assumptions for performing Mann-Whitney U test are that the independent variable is continuous or ordinal, the dependent variable is categorical and contains two independent groups, there is no dependence among observations in each groups and between groups and that the distribution is not binomial [22]. We can assume the following hypotheses for a two-sided Mann-Whitney U test:

H_0 *: the two populations have equal distributions*

H_1*: the two populations have unequal distributions*

For ordered observations x and y from populations defined by continuous cumulative distribution functions f and g, we can compute a U statistic for the ranks of x and y, so that the hypothesis $f = g$ can be validated [23].

$$\mathrm{U} = \mathrm{mn} + \frac{m(m+1)}{2} - \mathrm{T} \tag{7}$$

The U test is used for the statistic's computation (Equation 7), where m is the number of observations y, n is the number of observations x and T is Wilcoxon statistic which constitutes the sum of y's ranks in the ordered sequence of x's and y's. U quantifies how many times y precedes an x [23].

Point biserial correlation

In the case of the IEEE-CIS model a bipartite graph represents account nodes and devices nodes among which higher in-degree from a certain user indicated that the device is trusted. Therefore, a point biserial correlation (Equation 8) was used to determine if there is a correlation between the degree and fraud. It is used to compute a correlation between a Boolean variable and continuous variable.

$$r_{pb} = \frac{\bar{Y_1} - \bar{Y_2}}{\bar{s_y}} \sqrt{\frac{N_1 N_0}{N(N-1)}} \tag{8}$$

In the formula above (Equation 8), $\bar{Y_0}$ and $\bar{Y_1}$ are the means of the continuous values of the observations and N_0 and N_1 are the numbers of Boolean observations (encoded 0 and 1), N is the total number of observations, s_y is the standard deviation of the continuous observations [24].

3.3 Louvain modularity

Louvain method is a hierarchical optimization technique, which is used to extract graph communities. It is iteratively partitioning the graph into communities of densely connected nodes. Nodes belonging to other communities are sparely connected. The optimization is done based on the modularity scores. The modularity of a partition is a measure of the density of the links inside communities, as compared to links leading to outside structures. Its value ranges from -1 to 1 [25].
It can be used for evaluating the structure of social networks on platforms such as Twitter or Facebook. Moreover, it can be applied to financial graphs for the purpose of fraud detection, as there may be fraud rings which are characterized by greater modularity scores [2]. For the purpose of this analysis, its functionality was used to monitor financial flow in the detected communities to discover behaviors among nodes in communities where fraudulent activity was detected.

4 Explorative Analysis of the Financial Networks

In this section, the results of an exploratory analysis of the financial networks will be presented. For each data set, large-scale structure was studied, similarly as in the summaries presented by Newman, 2019. For the PaySim and Elliptic graphs, centrality metrics were computed, and Louvain communities were studied in terms of transactions' financial flow. In the case of IEEE-CIS graph, the result of examining the point biserial correlation between transaction class and degree of financial accounts is presented, as well as the ratio of the number of illicit transactions to licit ones in each weight category. The weight corresponds to the number of interactions with a certain device category by a financial account node.

4.1 Large-scale structure of the graphs

Table 7: Large-scale properties of the analyzed graphs.

Name	Type	n	m	c	S	l	C
PaySim	Directed	6876	4274	1.243165	0.006254	1.0	0.0
IEEE-CIS	Directed	9526	117386	24.64539	0.988873	1.0	N/A
Elliptic	Directed	203769	234355	1.150101	0.038671	125.2837	0.013762

The table summarized metrics for large-scale structure of the graphs. n corresponds to the number of nodes; m to the number of edges, c to the mean degree; S to the ratio of the number of nodes in the largest component to the number of all nodes; l to the mean distance between connected nodes pairs; C to the average local clustering coefficient.

The PaySim data set's graph

In terms of large-scale structure, PaySim data set turned out to be quite dissimilar from the graphs analyzed by Newman, 2019. The mean degree (c) varies for different networks and in this case, was closest to results obtained for software packages, email messages, peer-to-peer or student dating networks [18]. The fraction of nodes in the largest component (S) was incomparable with for instance social or technological networks, as it is significantly lower than scores of 0.5-1.0 [18]. The mean distance between nodes (l) is 1.0, which is also non-existent in the mentioned networks. The average clustering coefficient is zero (C), which also does not appear in the study conducted by Newman, 2019. Additional metrics such as graph density suggested that it is in fact very sparse, with the connectance of approximately 0.0002, the graph did not have any strongly connected components, it had 2602 weakly connected components and its diameter was 1. It can be considered reliable outcome, considering the visualization of the structure related to the first 1000 nodes in the PaySim graph from Figure 14.

The IEEE-CIS data set's graph

In case of the IEEE-CIS fraud detection data set, the mean degree (c) score is located between co-authorship networks and film actors networks for social graphs. It is higher than in PaySim and Elliptic, as IEEE-CIS graph is a specific, bipartite graph with big partitions constituted by popular operating systems and browser versions. They are characterized by high degrees of the device nodes. User nodes however, have fewer relationships, just as the less popular systems or systems with not popular browser versions. For the fraction of the size of the largest component to the size of the entire graph (S), a value of nearly 1.0 was found. This is mostly due to the popularity of Windows OS which constitutes the largest components of the graph. The mean distance (l) is 1.0 again, which is quite logical in this case, as it is a bipartite graph. Due to this reason, the mean clustering coefficient (C) cannot be included in the metrics for this case.

When it comes to additional scores, the graph is slightly denser than PaySim graph with connectance of around 0.003. It contains 50 weakly connected components and the diameter, similarly to PaySim, of 1.0.

The Elliptic data set's graph

Similarly to the PaySim data set's graph, the Elliptic data set is also characterized by the mean degree (c) scores similar to the software packages, email messages, peer-to-peer or student dating networks. As it is a network with many partitions for different entities, the fraction of the size of the largest component to the size of the entire graph (S) is relatively low, however, not as small as in case of the PaySim data set. Based on these two metrics and on the fidnings presented for the PaySim data set, we can conclude that standard structure of financial networks is characterized by relatively low mean degree scores (just above 1.0) and small number of nodes belonging to the largest component. The mean distance between

connected node pairs (l) is evidently higher than more the other analyzed networks - above 125. This leads to a conclusion that most likely there are many interconnections between entities in this network and the sparsity is somehow avoided - contrary to the PaySim graph. The clustering coefficient (C) is also higher than for the PaySim graph.

4.2 Centrality

4.2.1 The PaySim data set graph

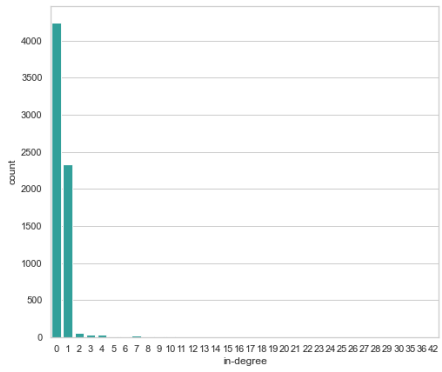
PaySim graph is characterized by positively skewed distributions for nearly all metrics of centrality with certain exceptions.

Degree

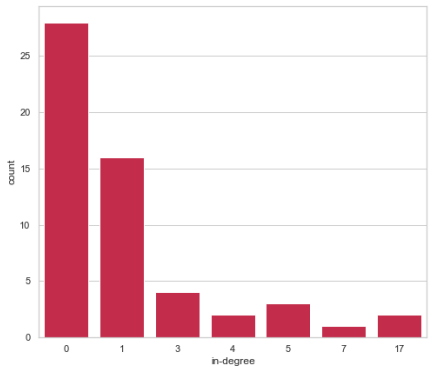
In the case of the simplest centrality measure, the in-degree is characterized by positive distributions for both fraudulent and non-fraudulent nodes (Figure 18). The fraudulent class distribution constitutes a partition of the non-fraudulent class distribution. The non-fraudulent class distribution has a long tail, so it is not possible to set a threshold of suspicious number of in-going links of individual nodes.

Figure 18: The distribution of in-degree for non-fraudulent and fraudulent class of transactions.

(a) Non-fraudulent class



(b) Fraudulent class

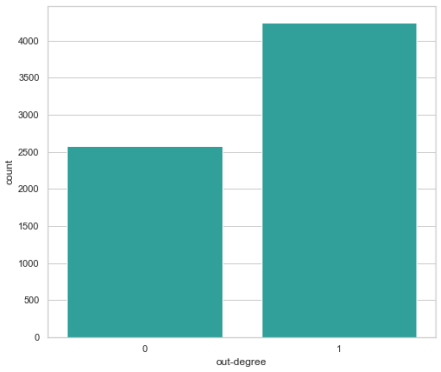


Both classes are characterized by positively-skewed distributions.

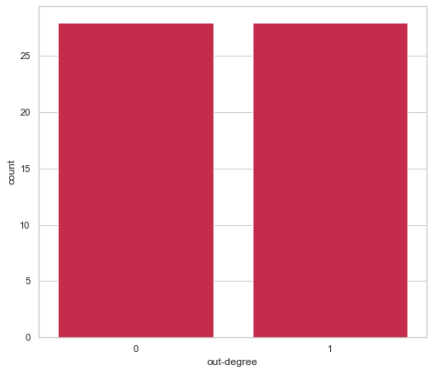
The out-degree distributions for fraudulent and non-fraudulent class of nodes differ. The former class is characterized by uniform, while the latter by negatively-skewed distribution. This is shown in Figure 19. Nevertheless, there are relatively few observations of the fraudulent nodes, therefore, again, the sample of fraudulent nodes' metrics melts into the non-fraudulent class at a transactions' population level.

Figure 19: The distribution of out-degree for non-fraudulent and fraudulent class of transactions.

(a) Non-fraudulent class



(b) Fraudulent class



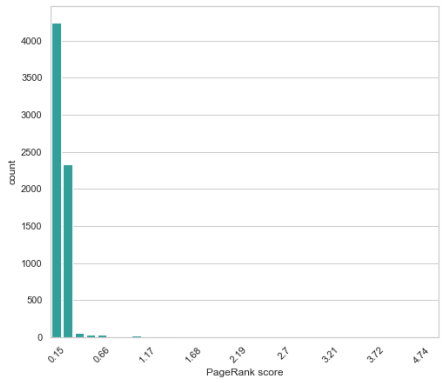
The out-degree distribution is negatively skewed in the case of non-fraudulent nodes and uniform for the fraudulent ones. Unfortunately, this is insufficient differentiation, as it occurs at a general, statistical level.

PageRank

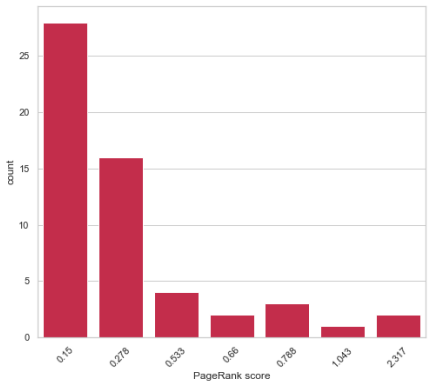
Similarly as in the case of in-degree, PageRank centrality scores also form a positively-skewed distributions for the fraudulent, as well as for the non-fraudulent class of nodes. It is shown in Figure 20. No significant differences at an individual level of observations - such as many outliers of fraudulent-class nodes could be found.

Figure 20: The distribution of PageRank centrality for non-fraudulent and fraudulent class of transactions.

(a) Non-fraudulent class



(b) Fraudulent class



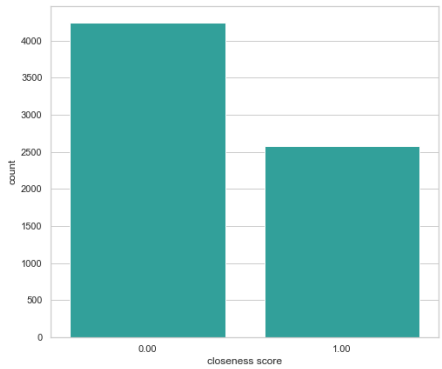
Both classes are characterized by positively-skewed distributions of the PageRank centrality scores.

Closeness

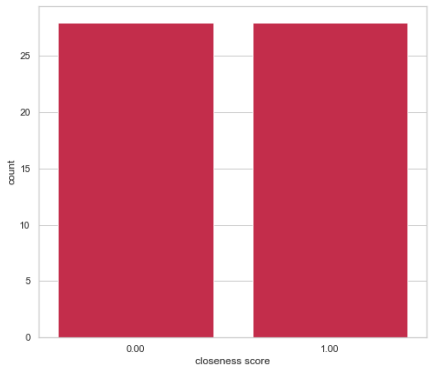
Closeness distributions differ between fraudulent and non-fraudulent class of nodes - the former being characterized by a uniform distribution and the latter by a positively-skewed distribution. This can be seen in Figure 26.

Figure 21: The distribution of closeness centrality for non-fraudulent and fraudulent class of transactions.

(a) Non-fraudulent class



(b) Fraudulent class



Similarly as for the out-degree distribution, the fraudulent nodes have a uniform distribution of closeness scores. Contrary to that example, the non-fraudulent nodes have a positively-skewed distribution.

Betweeness

For the betweenness scores, all of the observations obtained 0.0. Therefore, this metric emerged as meaningless. This is due to the structure of the network, which is not characterized by many interconnections among different partitions of the graph.

HITS

HITS centrality again did not provide interesting insights, as the fraudulent nodes' scores were contained within the range of non-fraudulent nodes and their distributions did not provide meaningful insights on the differences between these two groups of nodes. The authority centrality for fraudulent nodes as well as the hub centrality was 0.0 in all cases. As can be seen in Table 8 and Table 9 which show the authority and hub score distributions respectively, the fraudulent nodes would belong to the majority group of value 0.0 for both metrics.

Table 8: Authority score for the non-fraudulent nodes.

authority score	count
0	6814
0.001	3
0.046	1
0.026	1
0.999	1

The fraudulent nodes, having the authority score of 0.0 belong to the majority group for these scores.

Table 9: Hub score for the non-fraudulent nodes.

hub score	count
0	6763
0.154	42
0.007	36
0.004	35

Similarly as in the Table 8, the fraudulent nodes would belong to the majority group and are not easily distinguishable from the population of all nodes.

Mann-Whitney U tests

In the case of PaySim data set graph, the two-sided Mann-Whitney U tests proved that differences exist between the non-fraudulent and fraudulent classes of nodes for the distributions of in-degree, PageRank and closeness centrality. The null hypothesis assuming equality between these distributions was rejected due to statistically significant U test statistic values indicated by the p-value (assuming significance if p-value < 0.05). The tests' results are shown in Table 13.

Table 10: Two-sided Mann-Whitney U tests for centrality metrics of fraudulent and non-fraudulent class distributions.

Centrality	U test statistic	p-value
in-degree	226452.000	0.005
out-degree	167552.000	0.06
PageRank	380387.000	0.00
closeness	309848.000	0.00

The U test statistic was statistically significant for in-degree, PageRank and closeness centralities, thus the null hypothesis assuming equality between the distributions of the fraudulent and non-fraudulent class could be rejected.

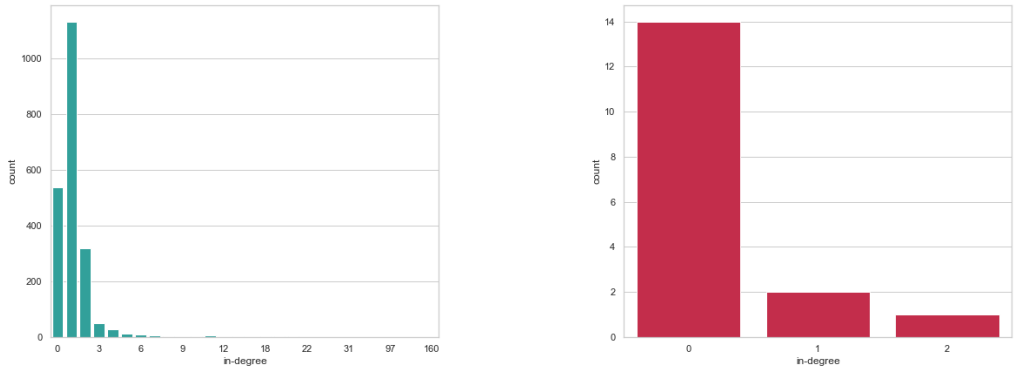
4.2.2 The Elliptic data set graph

Degree

As in the case of PaySim data set graph, the in-degree is characterized by positive distributions for both fraudulent and non-fraudulent nodes (Figure 22). Again, the fraudulent class distribution constitutes a partition of the non-fraudulent class distribution.

Figure 22: The distribution of in-degree for licit and illicit class of transactions.

(a) Licit class (b) Illicit class

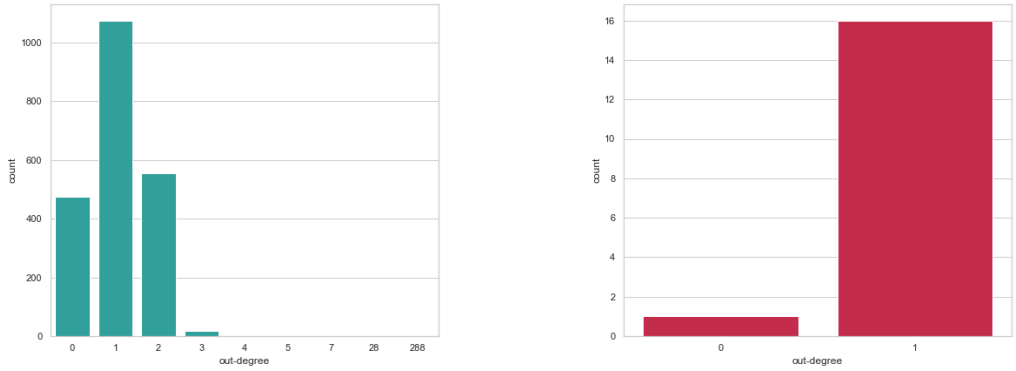


Both classes are characterized by positively-skewed distributions.

The out-degree distributions for illicit and licit class of nodes differ, again similarly to the case of PaySim data set. The licit class is characterized by slightly positively-skewed and the illicit by negatively-skewed distribution. This is shown in Figure 23. Just as before, there are relatively few observations of the illicit nodes, so the illicit transactions out-degree scores melt into the licit class of observations.

Figure 23: The distribution of out-degree for licit and illicit class of transactions.

(a) Licit class (b) Illicit class



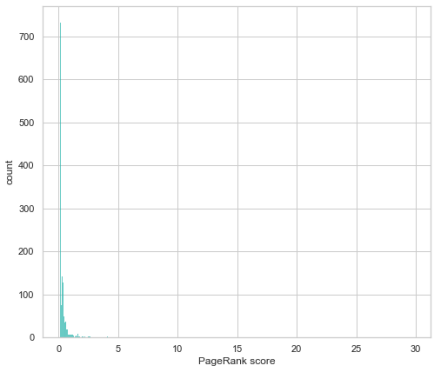
The licit class distribution is slightly positively-skewed and the illicit class - negatively skewed.

PageRank

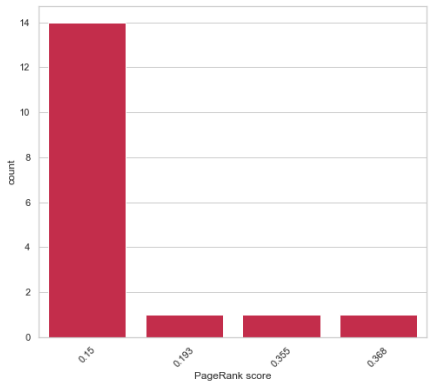
Similarly as in the case of PaySim data set, PageRank centrality scores also form a positively-skewed distributions for the illicit, as well as for the licit class of nodes. It is shown in Figure 24. Again, the illicit observations belong to the range of licit observations' values.

Figure 24: The distribution of PageRank centrality for licit and illicit class of transactions.

(a) Licit class



(b) Illicit class



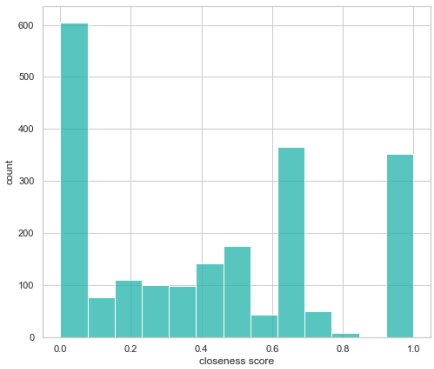
The licit class of transactions is strongly positively-skewed and the observations belonging to the illicit class constitute a part of the licit class centrality scores' range.

Closeness

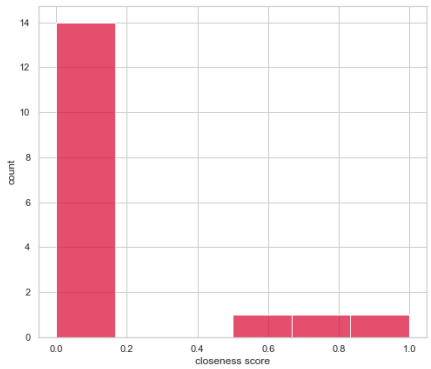
Closeness distributions differ between fraudulent and non-fraudulent class of nodes - the former being characterized by a uniform distribution and the latter by a positively-skewed distribution. This can be seen in Figure 25.

Figure 25: The distribution of closeness centrality for licit and illicit class of transactions.

(a) Licit class



(b) Illicit class



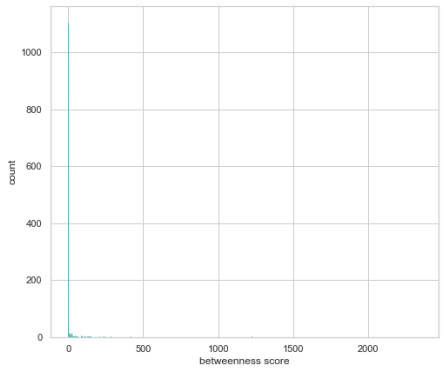
Similarly as for the out-degree distribution, the fraudulent nodes have a uniform distribution of closeness scores. Contrary to that example, the non-fraudulent nodes have a positively-skewed distribution.

Betweeness

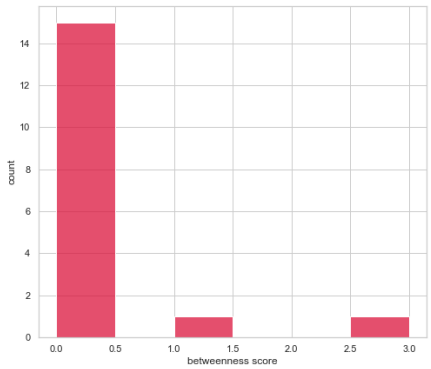
Contrary to the PaySim data set's example, there were different values of the betweenness scores for the nodes of Elliptic data set. The illicit nodes are characterized by low betweenness scores, thus they cannot be considered important *bridges* of that network. It is quite apparent that the structure of Elliptic data set's graph is more interesting than the one of the PaySim data set in terms of the financial flow and interconnections among accounts of that network.

Figure 26: The distribution of betweenness centrality for licit and illicit class of transactions.

(a) Licit class



(b) Illicit class



The illicit observations' scores tend to be low values, whereas the entire population is characterized by positively-skewed distribution with a long tail.

HITS

Similarly as in the case of the PaySim data set, HITS centrality did not provide interesting insights. The scores were within the range of the licit nodes belonging to the largest groups, thus the group of illicit transactions cannot be differentiated based on this metric. The authority centrality for illicit transaction nodes was 0.0 in all cases and 0.0 or 0.004 for the hub score. As can be seen in Table 11 and Table 12 which show the authority and hub score distributions respectively, the fraudulent nodes would belong to the majority groups.

Table 11: Authority score for the licit nodes.

authority score	count
0.000	1861
0.059	228
0.001	22
0.061	6
0.060	6
0.002	2
0.063	2
0.064	1
0.004	1
0.062	1

The illicit transactions, having the authority score of 0.0 belong to the majority group for these scores.

Table 12: Hub score for the illicit nodes.

hub score	count
0.000	7526
0.004	248
0.003	97
0.007	6
0.008	1
0.997	1
0.019	1

Similarly as in the Table 11, the fraudulent nodes would belong to the majority groups and cannot be flagged based on some anomalous value.

Mann-Whitney U tests

In the case of Elliptic data set graph, the two-sided Mann-Whitney U tests proved that differences exist between the licit and illicit classes of transaction nodes for the distributions of in-degree, closenes and betweenness centrality metrics. The null hypothesis assuming equality between these distributions was rejected due to statistically significant U test statistic values indicated by the p-value (assuming

significance if p-value < 0.05). The tests' results for this data set are shown in Table 13.

Table 13: Two-sided Mann-Whitney U tests for centrality metrics of fraudulent and non-fraudulent class distributions.

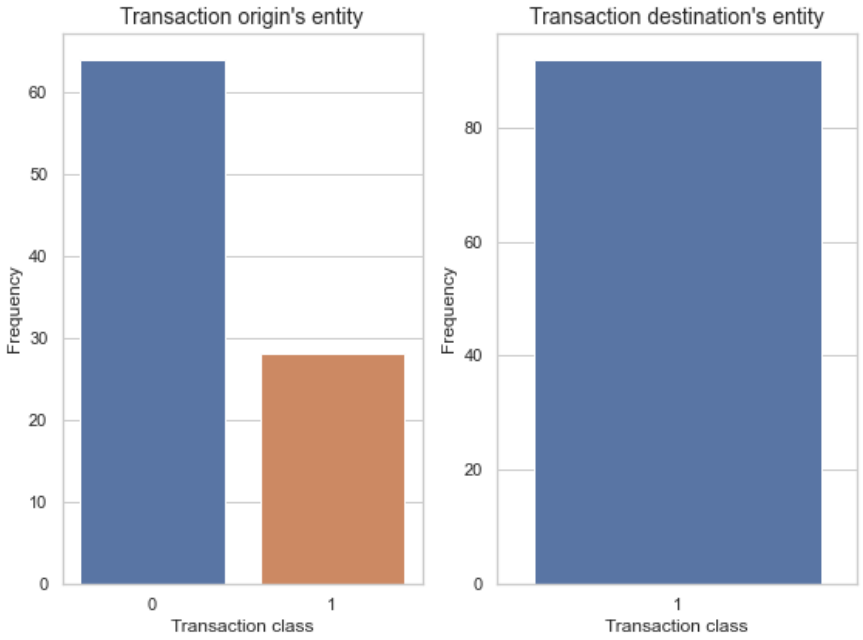
Centrality	U test statistic	p-value
in-degree	7807.000	0.00
out-degree	16454.000	0.479
PageRank	14590.500	0.164
closeness	11803.000	0.012
betweenness	10071.000	0.001

The U test statistic was statistically significant for in-degree, closeness and betweenness centralities, so the null hypothesis assuming equality between the distributions of the illicit and licit classes could be rejected.

4.3 Louvain communities

Louvain modularity allowed for obtaining dense communities and they were filtered in such a way to find communities with nodes which are involved in fraudulent activity. In these suspicious communities, a financial flow was studied, expecting that the general tendency will be to pass the amount from fraudulent nodes in such a way to eventually launder the money. At the structural level this means that one could expect a chain of several transactions of class fraudulent which ends with a non-fraudulent transaction. Interestingly, in case of PaySim graph, it was an opposite situation, and at the end of time step 3, all of the interactions between nodes became illicit.

Figure 27: The financial flow in Louvain communities for the PaySim data set.



In the case of PaySim data set, quite an illogical flow of the transactions was found. The nodes which are known to be involved in illicit activity do not launder the money, instead at the end of the final time step, there are more fraudulent activities (encoded as 1) than at the beginning. This may be due to sampling, however, it is a synthetic data set, thus the inconsistencies are expected.

For Elliptic Bitcoin transactions data set, this tendency was more logical, as it is quite evident that the funds coming from illicit transactions were laundered and at the end of the time step, more transactions were labeled as licit for these communities. Therefore, in short, one can expect that within these dense communities characterized by illegal activities, the number of illicit transactions will decrease, whereas the number of licit ones will increase. This is shown in Figure 28.

Figure 28: The financial flow in Louvain communities for the Elliptic data set.

In the case of Elliptic data set, there was a tendency to launder the money, as the number of illicit transactions decreased at the end of the time step, while the number of licit ones increased.

4.4 Fraud detection for suspicious devices - IEEE-CIS data set

The point biserial correlation was applied to check if higher degree scores for nodes are related to fraud. This method was used for IEEE-CIS graph, to determine if accounts which are connected to more devices are more likely to be affected by fraud. The result of this test is that could not confirm this hypothesis. The score obtained was 0.01599 and it can be considered statistically significant, due to the p-value which was approximately 0.00.

5 Conclusion and discussion

The real-world data sets were found to have certain similarities with large-scale networks analyzed by Newman, 2019, mostly in terms of the mean degree and the fraction of the number of nodes in the giant component to the total number of nodes in a graph. This means that Elliptic and IEEE-CIS graphs can be another interesting cases for further study of financial, large-scale networks. At the same time, a graph created from the data of an artificial data set - PaySim had little similarity to the large-scale real-world networks' properties. For all of the centrality metrics' distributions, no distinctive pattern can be found using goodness of fit analysis. All of the values in the ranges of fraudulent nodes occur also in the ranges of non-fraudulent nodes and cannot be considered anomalies neither in terms of their values, nor in terms of frequencies. The Mann-Whitney U tests allowed for stating general conclusions regarding the existence of differences between distributions of different metrics for fraudulent and non-fraudulent nodes classes, however, the results do not allow for telling precisely which nodes are fraudulent based on their properties. The obtained outputs only specify general characteristics of the differences between populations' distributions. Louvain modularity which was used to retrieve dense communities did not allow for automatic detection of fraud rings and it only provided some general characteristics of the financial flow between nodes. Interestingly, whereas in the real-world data set – Elliptic data set, the nodes seemingly aimed to manipulate the transactions in such a way that eventually the transferred funds will be flagged as licit activity, in PaySim data set, it was an opposite phenomenon. Perhaps some amount was removed from the financial graph after the money had been withdrawn by fraudsters, however, a few records indicated that the amounts were transferred further to some other illicit nodes.

The IEEE-CIS graph could not be analyzed using the same methods as PaySim and Elliptic graphs due to the characteristics of that data set. The nodes could not be labeled according to if they were involved in fraud or not, as some operating systems would have to be flagged as fraudulent in general. For instance, due to the fact that Windows had the most in-going relationships, among which some were fraudulent, it would automatically mean that presumably the most important node in the data set must be categorized as fraudulent, even though, most transactions performed using the devices with Windows OS installed, were licit. As mentioned before, construction of such a graph was intended to have experimental purpose, however, without sensitive information regarding the system or network used by each customer, it is impossible to determine which activity is normal and which is abnormal and should be flagged.

Apart from the need of sensitive information related to devices used by customers, other confidential data could be used to implement an efficient mechanism for flagging potentially suspicious account nodes. One of methods which could be applied to more precise data set is the use of Jaccard set similarities for sets of personally identifiable information (PII) such as phone number, social security numbers, e-mail addresses of account holders. Having the pair-wise scores for each pair of records, the highest Jaccard similarity scores can be examined, as duplication of records may indicate the existence of synthetic identities. Overall, the major problem for these analyses was access to sensitive information allowing for retrieving unique identities of accounts, devices and other entities. If such data could be obtained, some of the proposed methods could in fact be at least moderately effective.

References

[1] J. Gee and M. Button, *The latest data from around the world*. [Online]. Available: http://www.crowe.ie/wp-content/uploads/2019/08/The-Financial-Cost-of-Fraud-2019.pdf.

[2] A. Hodler, *The 1 Platform for Connected Data Financial Fraud Detection with Graph Data Science How Graph Algorithms Visualization Better Predict Emerging Fraud Patterns The 1 Platform for Connected Data Financial Fraud Detection with Graph Data Science*. 2021. [Online]. Available: https://go.neo4j.com/rs/710-RRC-335/images/Neo4j-Financial-Fraud-Detection-GDS-white-paper-EN-A4.pdf.

[3] E. Lopez-Rojas, A. Elmir, and S. Axelsson, *PAYSIM: A FINANCIAL MOBILE MONEY SIMULATOR FOR FRAUD DETECTION*. 2016. [Online]. Available: http://www.msc-les.org/proceedings/emss/2016/EMSS2016_249.pdf.

[4] M. Needham and A. E. Hodler, *Graph algorithms : practical examples in Apache Spark and Neo4j*. Beijing O'reilly, 2019, pp. 190–224, ISBN: 9781492047681.

[5] IEEE Computational Intelligence Society, *IEEE-CIS Fraud Detection*, 2019. [Online]. Available: https://www.kaggle.com/competitions/ieee-fraud-detection/data.

[6] Elliptic, *Elliptic data set*, 2019. [Online]. Available: https://www.kaggle.com/datasets/ellipticco/elliptic-data-set.

[7] E. Lopez-Rojas, *Synthetic financial datasets for fraud detection*, 2017. [Online]. Available: https://www.kaggle.com/datasets/ealaxi/paysim1.

[8] S. Seo, *A review and comparison of methods for detecting outliers in univariate data sets*, Aug. 2002. [Online]. Available: http://d-scholarship.pitt.edu/7948/.

[9] neo4j, *What is a graph database and property graph — neo4j*, Oct. 2019. [Online]. Available: https://neo4j.com/developer/graph-database/.

[10] ——, *Neo4j admin import - operations manual*, 2022. [Online]. Available: https://neo4j.com/docs/operations-manual/current/tutorial/neo4j-admin-import/.

[11] Neo Technology, *Bolt protocol*, 2019. [Online]. Available: https://boltprotocol.org/.

[12] Networkx developers, *Networkx — networkx documentation*, 2022. [Online]. Available: https://networkx.org/.

[13] C. B. Bruss, A. Khazane, J. Rider, R. Serpe, A. Gogoglou, and K. E. Hines, "Deeptrax: Embedding graphs of financial transactions," *arXiv:1907.07225 [cs, stat]*, Jul. 2019. [Online]. Available: https://arxiv.org/abs/1907.07225.

[14] L. Tung, *Google now gives you android notifications when new devices log into your accounts*, 2016. [Online]. Available: https://www.zdnet.com/article/google-now-gives-you-android-notifications-when-new-devices-log-into-your-accounts/.

[15] P. Stachyra, *Active directory: What do ctf environments teach us about attacking domain controllers?* Apr. 2020. [Online]. Available: https://medium.com/@hyphens443/attacking-domain-controllers-a45b9cb9651c.

[16] A. Kellner, M. Horlboge, K. Rieck, and C. Wressnegger, "False sense of security: A study on the effectivity of jailbreak detection in banking apps," *2019 IEEE European Symposium on Security and Privacy (EuroSP)*, Jun. 2019. DOI: 10.1109/eurosp.2019.00011.

[17] MDA Contributors, *User-agent - http — mdn*, 2022. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent.

[18] M. E. J. Newman, *Networks*. Oxford Oxford University Press, 2018, ISBN: 9780198805090.

[19] C. Stein, *Strongly Connected Components*. 2011. [Online]. Available: http://www.columbia.edu/~cs2035/courses/csor4231.F11/scc.pdf.

[20] neo4j, *Local clustering coefficient - neo4j graph data science*, 2022. [Online]. Available: https://neo4j.com/docs/graph-data-science/current/algorithms/local-clustering-coefficient/.

[21] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge Cambridge University Press, 2020, ISBN: 9781108476348.

[22] L. Statistics, *Mann-whitney u test in spss statistics — setup, procedure interpretation — laerd statistics*, 2013. [Online]. Available: https://statistics.laerd.com/spss-tutorials/mann-whitney-u-test-using-spss-statistics.php.

[23] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, Mar. 1947. DOI: 10.1214/aoms/1177730491.

[24] D. Kornbrot, "Point biserial correlation," *Wiley StatsRef: Statistics Reference Online*, Apr. 2014. DOI: 10.1002/9781118445112.stat06227.

[25] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, P10008, Oct. 2008. DOI: 10.1088/1742-5468/2008/10/p10008. [Online]. Available: https://doi.org/10.1088/1742-5468/2008/10/p10008.