

Graph Mining Approach to Suspicious Transaction Detection

Krzysztof Michalak, Jerzy Korczak

Institute of Business Informatics

Wroclaw University of Economics, Wroclaw, Poland

Email: {krzysztof.michalak, jerzy.korczak}@ue.wroc.pl

Abstract—Suspicious transaction detection is used to report banking transactions that may be connected with criminal activities. Obviously, perpetrators of criminal acts strive to make the transactions as innocent-looking as possible. Because activities such as money laundering may involve complex organizational schemes, machine learning techniques based on individual transactions analysis may perform poorly when applied to suspicious transaction detection.

In this paper, we propose a new machine learning method for mining transaction graphs. The method proposed in this paper builds a model of subgraphs that may contain suspicious transactions. The model used in our method is parametrized using fuzzy numbers which represent parameters of transactions and of the transaction subgraphs to be detected. Because money laundering may involve transferring money through a variable number of accounts the model representing transaction subgraphs is also parametrized with respect to some structural features. In contrast to some other graph mining methods in which graph isomorphisms are used to match data to the model, in our method we perform a fuzzy matching of graph structures.

I. INTRODUCTION

FINANCIAL institutions such as banks are legally obliged to monitor activities of their customers and to report events that may indicate involvement in a criminal act. In the case of bank transactions monitoring, one of the goals is to detect money laundering activities i.e. activities aimed at concealing the origin of illegally-obtained money.

There are several difficulties in money laundering detection. People involved in money laundering obviously try to conceal the real purpose of money transfers used in this process. Therefore, one can expect that individual transactions will not clearly stand out from amongst other bank transfers. The probability of a fraud depends not only on parameters of individual bank transfer but also on relations with other transfers and the entities that send them.

The volume and value of transactions reported as suspicious are very high. For example, the value of transactions reported to the anti-money laundering watchdog by Russian financial institutions in the first nine months of 2010 was 120 trillion roubles (2.44 trillion pounds, 3.8 trillion dollars) [9]. 5.6 million filings were made by banks, insurance companies and financial service companies in this period.

In the area of fraud detection it is typical that events to be detected are in considerable minority compared to the overall amount of data. For example, in 2010 there were

593,819 fraudulent credit card transactions detected in Australia worth in total \$145,854,208 [7]. In the same year over 1.4 billion credit card transactions were made for a total amount of more than 195 billion dollars (values calculated using statistics from [8]). The same sources report 241,063 fraudulent credit card transactions totalling \$85,215,615 in 2006 and a total number of over 1.1 billion transactions worth in total over 155 billion dollars. In the above-cited cases fraudulent transactions constitute only 0.02 – 0.04% of all transactions which means that data manifest extreme class imbalance. This in itself creates a significant challenge for many machine learning methods.

Typically, the money laundering process is divided into three stages: placement, layering and integration [5]. Each of the stages involves various schemes of money transferring between bank accounts. These schemes can be identified as subgraphs in the transaction graph. Figure 1 shows two simple examples of subgraphs which represent transferring money via a number of intermediaries.

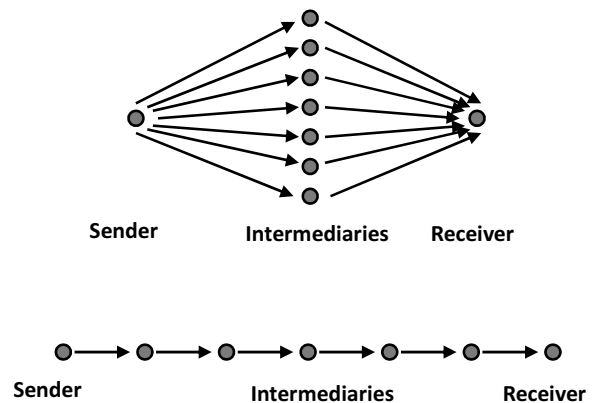


Fig. 1. Examples of subgraphs which may indicate money-laundering activities

In the first case a larger amount is split to smaller transfers in order to decrease the probability that individual transactions will be reported as suspicious. The second case serves the purpose of obscuring the connection between the sender and the receiver. The entire money laundering operation may involve many such schemes, so identification of a suspicious subgraph may help in uncovering much larger network of ille-

gal transactions. Money laundering operations are intertwined with many other transactions, including legal ones. Figure 2 shows a small subgraph of transactions (in which vertices are shaded in gray) that may raise suspicion because funds are transferred from one account to another via many independent transaction chains. It is clear that accounts participating in what may turn out to be illegal transaction structuring may as well be engaged in other, probably harmless activities.

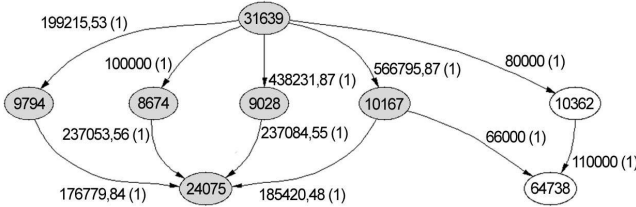


Fig. 2. Suspicious transaction graph connected with other, possibly legal activities. Vertex labels are account identifiers, edge labels contain number of transactions between the two accounts and the total amount transferred.

II. PROPOSED METHOD

Money laundering is hard to detect because the occurrences are very infrequent in the graph of all transactions and individual bank transfers involved in a money laundering scheme are structured so that they appear as legitimate. These obstacles are especially hard to overcome if the detection process is based only on features of individual transactions. Therefore, graph mining methods seem to be interesting, because they can detect complex dependencies between transactions. It is also possible to take into account properties and relations of entities involved in sending and receiving the transfers.

We present a method for graph structure learning using a model which can be trained on a previously annotated graph of transactions and then can be matched against a graph of unannotated transactions in order to select a number of transactions for a more thorough checking by human expert. This is in agreement with the mode of operation of financial watchdog institutions which employ experts to scrutinize suspicious transactions. Because of a huge number of transactions the work of an expert can be made significantly more efficient if a computer system is able to suggest a limited number of transactions to be checked for indications of possible money laundering. The results of the expert's work may in turn be used to train the system again.

The work presented in this paper is a part of a larger research project carried out at our Institute aimed at developing various money laundering detection methods. One of the research topics in this project is money laundering detection in data warehouses [3], [4]. Suspicious transactions discovered in data warehouses can be used as training data for the method described in this paper.

As described in the introduction, organizational schemes involved in money laundering may vary to a great extent with respect to the number of transactions used to conceal the nature of the activity. Therefore, we propose a model which can

adapt to training data with respect not only to transaction and account parameters but also with respect to the graph structure.

Proposed model represents subgraphs similar in structure to the graph presented in Figure 3.

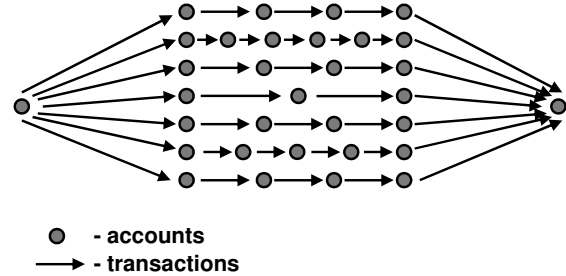


Fig. 3. General structure of a detected subgraph

In order to model such subgraphs, hierarchical three-level patterns are generated. Some of the parameters of the model are polygonal fuzzy numbers [1]. They are denoted using the hat ($\hat{\cdot}$) symbol. In the proposed method we use simplified polygonal fuzzy numbers which use only 6 real numbers x_1 , x_2 , x_3 , x_4 , m_2 and m_3 . Membership function $\mu_{\hat{x}}$ of such fuzzy number \hat{x} is presented in Figure 4 and is calculated as follows:

$$\mu_{\hat{x}}(x) = \begin{cases} 0 & \text{for } x \leq x_1, \\ m_2 \cdot \frac{x-x_1}{x_2-x_1} & \text{for } x \in (x_1, x_2], \\ m_2 + (m_3 - m_2) \cdot \frac{x-x_2}{x_3-x_2} & \text{for } x \in (x_2, x_3], \\ m_3 \cdot \frac{x_4-x}{x_4-x_3} & \text{for } x \in (x_3, x_4), \\ 0 & \text{for } x > x_4. \end{cases} \quad (1)$$

where:

$\hat{x} = \langle x_1, x_2, x_3, x_4, m_2, m_3 \rangle$ - polygonal fuzzy number.

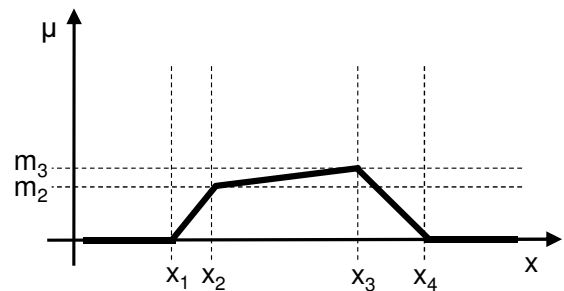


Fig. 4. Polygonal fuzzy number with components x_1 , x_2 , x_3 , x_4 , m_2 and m_3

The lowest level of the model is a *TR* pattern that describes a single transaction:

$$TR = \langle \hat{a}, r(\cdot), s(\cdot) \rangle, \quad (2)$$

where:

\hat{a} —polygonal fuzzy number representing transaction amount,

$s(\cdot)$ —function assigning weights to classes to which entities sending transfers belong,

$r(\cdot)$ —function assigning weights to classes to which entities receiving transfers belong.

The middle level is a *SER* pattern that describes transaction chains in which transactions are connected in series:

$$SER = \langle \hat{m}, \hat{\delta} \rangle, \quad (3)$$

where:

\hat{m} —polygonal fuzzy number representing the number of transactions in a chain,

$\hat{\delta}$ —polygonal fuzzy number representing the ratio of amount transferred in the last transaction to the amount transferred in the first transaction.

The top level is a *PAR* pattern that describes parallel transaction chains (which are described by the *SER* pattern)

$$PAR = \langle \hat{n}, \hat{\Delta}, \tau \rangle, \quad (4)$$

where:

\hat{n} —polygonal fuzzy number representing the number of transactions in a chain,

$\hat{\Delta}$ —polygonal fuzzy number representing the ratio of the sum of amounts received by the receiving account to the sum of amounts sent from the sending account.

τ —acceptance threshold used for deciding which transaction subgraphs match the pattern.

A complete pattern contains one set of parameters for each of the three levels.

Using such patterns, transaction subgraphs are evaluated in the following manner. First, weights are assigned to individual transfers using parameters of the *TR* pattern. A transfer T of amount a sent by an entity belonging to a class c_s to an entity belonging to a class c_r is assigned a weight w_{TR} which is calculated as:

$$w_{TR}(T) = \mu_{\hat{a}}(a) \cdot s(c_s) \cdot r(c_r), \quad (5)$$

where:

$TR = \langle \hat{a}, s(\cdot), r(\cdot) \rangle$ - pattern to which the transaction is matched,

$\mu_{\hat{a}}(\cdot)$ —membership function of the fuzzy number \hat{a} .

Transaction chains are evaluated using parameters of the *SER* pattern. A transaction chain L of length m in which the ratio of amount transferred in the last transaction to the amount transferred in the first transaction equals δ is assigned a weight w_{SER} which is calculated as:

$$w_{SER}(L) = \frac{\sum_{T \in L} w_{TR}(T)}{m} \cdot \mu_{\hat{m}}(m) \cdot \mu_{\hat{\delta}}(\delta), \quad (6)$$

where:

$SER = \langle \hat{n}, \hat{\delta} \rangle$ - pattern to which the transaction chain is matched,

TR —pattern used to match individual transactions,

T —transaction which belongs to the transaction chain L ,

$\mu_{\hat{m}}(\cdot)$ —membership function of the fuzzy number \hat{m} ,

$\mu_{\hat{\delta}}(\cdot)$ —membership function of the fuzzy number $\hat{\delta}$.

Subgraphs consisting of parallel paths are evaluated using parameters of the *PAR* pattern. A subgraph P containing n parallel paths in which the ratio of the sum of amounts received by the receiving account to the sum of amounts sent from the sending account equals Δ is assigned a weight w_{PAR} which is calculated as:

$$w_{PAR}(P) = \frac{\sum_{L \in P} w_{SER}(L)}{n} \cdot \mu_{\hat{n}}(n) \cdot \mu_{\hat{\Delta}}(\Delta), \quad (7)$$

where:

$PAR = \langle \hat{n}, \hat{\Delta}, \tau \rangle$ - pattern to which the transaction subgraph is matched,

SER - pattern used to match transaction chains,

L - transaction chain which belongs to the subgraph P ,

$\mu_{\hat{n}}(\cdot)$ - membership function of the fuzzy number \hat{n} ,

$\mu_{\hat{\Delta}}(\cdot)$ - membership function of the fuzzy number $\hat{\Delta}$.

The entire *PAT* pattern includes all the parameters required to match graph elements at each of three levels:

$$PAT = \langle TR, SER, PAR \rangle, \quad (8)$$

$$PAT = \langle \hat{a}, r(\cdot), s(\cdot), \hat{m}, \hat{\delta}, \hat{n}, \hat{\Delta}, \tau \rangle. \quad (9)$$

Fuzzy parameters \hat{a} , \hat{m} , $\hat{\delta}$, \hat{n} and $\hat{\Delta}$ can be described by 6 real numbers each. Functions $r(\cdot)$ and $s(\cdot)$ have discrete domains and thus are adequately represented by discrete sets of weights. The entire *PAT* pattern is thus described by $31 + 2k$ real numbers, where k is the number of entity classes.

Model parameters ($31 + 2k$ real numbers describing a *PAT* pattern) have to be adjusted based on a training data set containing transactions annotated by an expert. For optimization of parameters of *PAT* patterns we propose to use a genetic algorithm with the following properties.

Specimen—a set of $31 + 2k$ real numbers interpreted as *PAT* pattern parameters,

Mutation—each of the $31 + 2k$ real numbers in each of the specimens is mutated with equal probability P_{mut} .

Mutation of the numbers that represent fuzzy number components x_1 , x_2 , x_3 and x_4 is controlled by additional parameters Δ_x , m_x and M_x defined separately for each fuzzy parameter of the model (i.e. \hat{a} , \hat{m} , $\hat{\delta}$, \hat{n} and $\hat{\Delta}$). First, a random value d is drawn with uniform probability from the range $[-\frac{\Delta_x}{2}, \frac{\Delta_x}{2}]$. Then, value of the component x_i is modified by adding d . As this modification may disrupt the order of fuzzy number components x_1 , x_2 , x_3 and x_4 and may also lead to violation of constraints $m_x \leq x_1$ and $x_4 \leq M_x$ a check (and possibly also a correction) must be performed. This correction is performed as follows:

- if $x_1 < m_x$ then $x_1 \leftarrow m_x$
- for $i = 1, 2, 3$: if $x_{i+1} \leq x_i$ then $x_{i+1} \leftarrow x_i + 0.001$
- if $x_4 > M_x$ then $x_4 \leftarrow M_x$
- for $i = 3, 2, 1$: if $x_i \geq x_{i+1}$ then $x_i \leftarrow x_{i+1} - 0.001$

Mutation of the numbers that represent fuzzy number parameters m_2 and m_3 , weights assigned to entity classes and the value of acceptance threshold τ is performed by adding a value drawn with uniform probability from the range $[-0.005, 0.005]$ and ensuring that the result is in the range $[0, 1]$.

Selection—a standard roulette-wheel selection procedure [6] is used.

Crossover—a standard single-point crossover operator [2] is used. Probability of a crossover being performed on any two specimens is controlled by the parameter P_{cross} .

Evaluation function—the evaluation function for a given specimen S is calculated in the following way:

- from the specimen S a pattern $PAT(S) = \langle TR(S), SER(S), PAR(S) \rangle$ is constructed using $31 + 2k$ real numbers as parameters,
- a set \mathcal{P} is constructed containing those subgraphs G that match the pattern $PAT(S)$ and have a weight $w_{PAR(S)}(G) > \tau$,
- for each subgraph $G \in \mathcal{P}$ a total number of transactions $t_n(G)$ in this subgraph and a sum of weights of transactions $t_w(G)$ in this subgraph are calculated. Weights of transactions are based on annotations made by the expert. For example, transactions annotated as "illegal" may have a weight 1.0, transactions annotated as "legal" a weight 0.0 and transactions annotated as "not classified" a weight 0.1.
- evaluation of the specimen is calculated as:

$$F(S) = \frac{\sum_{G \in \mathcal{P}} w_{PAR(S)}(G) \cdot t_w(G)}{\sum_{G \in \mathcal{P}} t_n(G)}. \quad (10)$$

The specimen (or specimens) achieving the highest values of evaluation function F may be used to identify suspicious transactions in previously unseen data.

III. EXPERIMENTS

For experiments training and testing data sets containing transactions annotated by an expert are required. Because such data sets are hard to obtain due to confidentiality of banking data, the experiments performed so far were based on artificially-generated data. In order to build a data set containing transactions similar to those encountered in real-life we used a model which represents transactions in a "mini-economy" during a period of one year. In this model, three classes of economic entities are defined: companies, individual persons and offices (tax offices and social security offices). Economic entity classes are characterized by probability distributions which are used to generate parameters for instances belonging to each class.

Companies are characterized by the following probability distributions:

- distribution of the number of employees: $N(m_E, \sigma_E)$,
- distribution of salary: $N(m_S, \sigma_S)$,
- distribution of the number of goods sold per year: $N(m_G, \sigma_G)$,

- distribution of prices of goods: $N(m_P, \sigma_P)$.

A predefined number of companies n_c is generated. For each company a number of employees n_E is drawn from the Gaussian distribution $N(m_E, \sigma_E)$. Then, n_E persons are added to the model. For each of the 12 months in a year a salary a_s is drawn from the Gaussian distribution $N(m_S, \sigma_S)$ and a transaction representing the payment (with the amount a_s) is generated.

Next, buying of goods is simulated. For each company a number of goods sold during the simulated year n_G is drawn from the Gaussian distribution $N(m_G, \sigma_G)$. For each good a price a_p is drawn from the Gaussian distribution $N(m_P, \sigma_P)$. A buyer is selected at random from all employees of all companies and a new transaction (with the amount a_p - a payment for the good) is added.

Generation of offices is controlled by the parameters n_T - the number of tax offices and n_F —the number of social security offices. Offices are also characterized by two probability distributions:

- distribution of tax rate: $N(m_T, \sigma_T)$,
- distribution of social security fee rate: $N(m_F, \sigma_F)$.

One tax office and one social security office are assigned at random to each company. The sum of payments C_p received by the company for goods sold in each month is calculated and a tax rate α_T is drawn from the Gaussian distribution $N(m_T, \sigma_T)$. Tax amount a_T is calculated as $a_T = C_p \cdot \alpha_T / 100$ and a transaction sent by the company to the tax office account is generated. Social security fee a_F is calculated as $a_F = C_s \cdot \alpha_F / 100$ based on the sum of salaries in each month C_s and a social security fee rate α_F which is drawn from the Gaussian distribution $N(m_F, \sigma_F)$.

The steps described above produce a set of transactions representing the usual activities observed in economy. To this set of transactions n_{ML} money laundering schemes are added. Each of these schemes consists of a sender, a number n_B of intermediaries and a receiver. Transfers are sent from the sender to one intermediary and then to the receiver. Each parallel path goes through one intermediary only, so n_B parallel paths are created. The generation of the money laundering schemes is characterized by the following probability distributions:

- distribution of the amount sent from the sender to one intermediary: $N(m_Q, \sigma_Q)$,
- distribution of the number of intermediaries (equal to the number of parallel paths): $N(m_B, \sigma_B)$,
- distribution of the fraction of the amount received by the intermediary that is forwarded to the receiver: $N(m_\Delta, \sigma_\Delta)$.

Generated transactions are annotated in the following manner:

- legal—tax and social security fee transactions,
- illegal—transactions belonging to the generated money laundering schemes,
- unknown—all the remaining transactions (salaries and payments for goods).

Using data generation method described above, we have generated four data sets: $SMALL_A$, $SMALL_B$, $LARGE_A$, $LARGE_B$. All data sets were generated using the same parameters for offices: $n_T = 5$, $n_F = 5$, $m_T = 15$, $\sigma_T = 1.5$, $m_F = 20$, $\sigma_F = 2.0$. Also, parameters of money laundering were the same: $m_Q = 5000$, $\sigma_Q = 1000$, $m_B = 40$, $\sigma_B = 10$, $m_\Delta = 1.0$, $\sigma_\Delta = 0.1$. These data sets contain three classes of companies that can be briefly characterized as large (L), medium (M) and small (S). The $SMALL$ and $LARGE$ data sets differ in the number of companies in each class. Parameters controlling generation of companies for data sets $SMALL_A$ and $SMALL_B$ and for data sets $LARGE_A$ and $LARGE_B$ are summarized in Tables I and II respectively.

TABLE I

PARAMETERS CONTROLLING GENERATION OF COMPANIES FOR DATA SETS $SMALL_A$ AND $SMALL_B$

Parameter	Company class		
	large	medium	small
n_C	2	4	25
m_E	5 000	500	50
σ_E	1 000	100	20
m_S	6 000	5 000	4 000
σ_S	1 500	1 200	1 000
m_G	100 000	1 000	100
σ_G	30 000	300	30
m_P	50	500	500
σ_P	10	100	100

TABLE II

PARAMETERS CONTROLLING GENERATION OF COMPANIES FOR DATA SETS $LARGE_A$ AND $LARGE_B$

Parameter	Company class		
	large	medium	small
n_C	2	8	100
m_E	5 000	500	50
σ_E	1 000	100	20
m_S	6 000	5 000	4 000
σ_S	1 500	1 200	1 000
m_G	1 000 000	10 000	1 000
σ_G	300 000	3 000	300
m_P	50	500	500
σ_P	10	100	100

The number of accounts and transactions of each type in each of the data sets is summarized in Table III.

In the experiments, one of data sets in a $LARGE/SMALL$ pair was used for training and the other one for testing. A population of $N_{pop} = 20$ specimens was trained for $N_{gen} = 20$ generations of genetic algorithm. Crossover and mutation probabilities were set to $P_{cross} = 0.1$ and $P_{mut} = 0.01$ respectively. Parameters controlling mutation of x_i components of fuzzy numbers are summarized in Table IV.

Specimen evaluation requires that transaction annotations are converted to numerical weights. In the experiments a weight 1.0 was assigned to “illegal” transactions,

TABLE III

THE NUMBER OF ACCOUNTS AND TRANSACTIONS OF EACH TYPE IN EACH OF THE DATA SETS

Object type	Number of objects			
	$SMALL_A$	$SMALL_B$	$LARGE_A$	$LARGE_B$
Accounts	11 238	11 401	19 261	21 270
companies	31	31	110	110
offices	10	10	10	10
personal	11 197	11 360	19 261	21 150
Transactions	294 972	383 463	2 854 965	2 625 671
legal	744	744	2 640	2 640
unknown	289 336	377 207	2 848 435	2 619 049
illegal	4 892	5 512	3 890	3 982
annot. ratio	0.0191	0.0166	0.0023	0.0025

TABLE IV

PARAMETERS CONTROLLING MUTATION OF x_i COMPONENTS OF FUZZY NUMBERS

Fuzzy number	Parameter		
	Δ_x	m_x	M_x
\hat{a}	200	range not limited	
\hat{m}	x_i not mutated, fixed at 0, 1, 2 and 3, only m_2 and m_3 are mutated		
$\hat{\delta}$	0.1	0.5	1.5
\hat{n}	2	3	100
$\hat{\Delta}$	0.1	0.5	1.5

a weight 0.0 to transactions annotated as “legal” and a weight 0.1 to transactions annotated as “not classified.”

Tests were performed in 10 independent iterations for each pair of $SMALL$ and $LARGE$ data sets. In each iteration, after the training has been completed, the best specimen (with the highest value of the evaluation function) was selected from the population and it was used for selecting suspicious transaction subgraphs from the testing data set. Only one, the best specimen, was used, because we wanted to limit the number of transactions marked as suspicious. In the real-life environment, transactions marked as suspicious are reviewed by human expert which obviously imposes limitations on the number of transactions that can be processed. The number of transactions that were actually legal, illegal and of unknown status (according to annotations) was used to measure the quality of the detection. Results are summarized in Tables V-VIII.

During the experiments execution time of test iterations was recorded. Measured values are summarized in Table IX.

A meaningful comparison of execution times is only possible for tests performed on the same machine. Therefore, only three tests that were performed on the same computer are included in the table. For comparison, the number of accounts and the number of transactions in each data set are also presented in the table.

TABLE V

THE NUMBER OF "LEGAL", "NOT CLASSIFIED" AND "ILLEGAL" TRANSACTIONS MARKED AS SUSPICIOUS IN THE EXPERIMENT WITH TRAINING DATA SET *SMALL_A*, TEST DATA SET *SMALL_B* AND EACH ITERATION PERFORMED INDEPENDENTLY. PERCENTAGE OF "NOT CLASSIFIED" TRANSACTIONS AMONG TRANSACTIONS MARKED AS SUSPICIOUS: 15.98%

Iteration	Number of transactions		
	<i>legal</i>	<i>unknown</i>	<i>illegal</i>
1	0	0	26
2	0	9	9
3	0	8	8
4	0	0	30
5	0	0	26
6	0	0	26
7	0	0	46
8	0	11	11
9	0	7	7
10	0	0	30
TOTAL	0	35	219

TABLE VI

THE NUMBER OF "LEGAL", "NOT CLASSIFIED" AND "ILLEGAL" TRANSACTIONS MARKED AS SUSPICIOUS IN THE EXPERIMENT WITH TRAINING DATA SET *SMALL_B*, TEST DATA SET *SMALL_A* AND EACH ITERATION PERFORMED INDEPENDENTLY. PERCENTAGE OF "NOT CLASSIFIED" TRANSACTIONS AMONG TRANSACTIONS MARKED AS SUSPICIOUS: 13.88%

Iteration	Number of transactions		
	<i>legal</i>	<i>unknown</i>	<i>illegal</i>
1	0	10	10
2	0	0	34
3	0	10	10
4	0	0	30
5	0	0	32
6	0	0	36
7	0	0	34
8	0	10	11
9	0	9	9
10	0	0	36
TOTAL	0	39	242

IV. CONCLUSION

In this paper we presented a graph mining method intended for detection of suspicious transactions. Contrary to data mining methods based solely on transaction features the method proposed in this paper takes into consideration those dependencies between individual transfers that may be indicative of illegal activities. We expect this feature to be a significant advantage, because single transactions are often tailored by criminals in order to be as innocent-looking as possible.

Results of the experiments suggest that the method proposed in the paper has several advantageous properties:

- in the experiments the proposed method managed to avoid marking as suspicious of any transactions that were annotated as "legal". Although it is not clear at this point to what extent this quality will persist in the case of other data sets, this is a desirable behaviour for a suspicious transaction detection method;

TABLE VII

THE NUMBER OF "LEGAL", "NOT CLASSIFIED" AND "ILLEGAL" TRANSACTIONS MARKED AS SUSPICIOUS IN THE EXPERIMENT WITH TRAINING DATA SET *LARGE_A*, TEST DATA SET *LARGE_B* AND EACH ITERATION PERFORMED INDEPENDENTLY. PERCENTAGE OF "NOT CLASSIFIED" TRANSACTIONS AMONG TRANSACTIONS MARKED AS SUSPICIOUS: 27.05%

Iteration	Number of transactions		
	<i>legal</i>	<i>unknown</i>	<i>illegal</i>
1	0	10	10
2	0	11	11
3	0	0	22
4	0	0	22
5	0	9	9
6	0	0	24
7	0	0	22
8	0	0	22
9	0	12	12
10	0	24	24
TOTAL	0	66	178

TABLE VIII

THE NUMBER OF "LEGAL", "NOT CLASSIFIED" AND "ILLEGAL" TRANSACTIONS MARKED AS SUSPICIOUS IN THE EXPERIMENT WITH TRAINING DATA SET *LARGE_B*, TEST DATA SET *LARGE_A* AND EACH ITERATION PERFORMED INDEPENDENTLY. PERCENTAGE OF "NOT CLASSIFIED" TRANSACTIONS AMONG TRANSACTIONS MARKED AS SUSPICIOUS: 21.05%

Iteration	Number of transactions		
	<i>legal</i>	<i>unknown</i>	<i>illegal</i>
1	0	0	26
2	0	9	9
3	0	0	28
4	0	9	9
5	0	0	28
6	0	10	10
7	0	0	24
8	0	0	26
9	0	10	10
10	0	10	10
TOTAL	0	48	180

- more than 2/3 of transactions marked as suspicious were actually involved in money laundering schemes;
- as shown in Table IX computation time doubled with the similar—twofold increase in the number of accounts. The difference in transaction number between the same two data sets was about 10 times. This is beneficial, because the volume of transactions in historic record increases more rapidly than the number of bank customers;

The fact that the proposed method did not mark any legal transactions as suspicious is promising. It is easy to understand that reporting transactions which are considered legal by human experts most probably means raising a false alarm. A much harder question is, how large percentage of transactions that are unannotated by human expert (or annotated as "not classified") should be marked as suspicious by the algorithm. In the case of artificial data used in this paper it is obvious that all "not classified" transactions were,

TABLE IX

EXECUTION TIME (IN SECONDS) OF TEST ITERATIONS. ALL TESTS PRESENTED IN THE TABLE WERE PERFORMED ON THE SAME MACHINE.

Training data set	<i>SMALL_B</i>	<i>LARGE_A</i>	<i>LARGE_B</i>
accounts	11 401	19 261	21 270
transactions	383 463	2 854 965	2 625 671
Test data set	<i>SMALL_A</i>	<i>LARGE_B</i>	<i>LARGE_A</i>
accounts	11 238	21 270	19 261
transactions	294 972	2 625 671	2 854 965
Iteration 1 time	1 108	2 110	2 204
Iteration 2 time	1 191	2 088	2 119
Iteration 3 time	1 197	2 027	2 067
Iteration 4 time	1 126	2 146	2 098
Iteration 5 time	1 162	2 109	2 173
Iteration 6 time	1 156	2 116	2 165
Iteration 7 time	1 189	2 060	2 451
Iteration 8 time	1 166	2 130	2 141
Iteration 9 time	1 185	2 128	2 409
Iteration 10 time	1 172	2 075	2 096
Average time	1 165	2 099	2 192

in fact, legal. Therefore, the percentage of "not classified" transactions among transactions marked as suspicious can be interpreted as a false positive ratio. In real-life scenario, however, some of the "not classified" transactions contained in the training data set will actually be illegal (i.e. involved in a money laundering schemes) because it is never possible to identify all illegal activities. Nevertheless, statistically, most of the "not classified" transactions are legal. It is thus hard to decide at this point, to what extent the learning model should be trained to avoid reporting transactions similar to "not classified" examples from the training set.

Further work may concern improving the precision of the detection (in order to avoid too many legal transactions being submitted to human experts for evaluation) but also improving the completeness of the results (ensuring that as many illegal transactions as possible are marked as suspicious). Also, improvement in computational speed may be important, especially because computation time may be a factor limiting the possibility of searching for more complex subgraph patterns.

In order to achieve the above mentioned goals further work may be conducted in some of the following directions:

- training a population of positive and negative patterns. Currently, specimens represent only such patterns that identify suspicious subgraphs. Learning patterns that represent legal transaction subgraphs may help in reducing the false positive ratio;
- using decision rules designed by experts to improve detection quality. Apart from knowledge extracted from annotated graphs, rules, for example excluding tax transfers from suspicion, may help in reducing the search space that must be processed;
- interacting with human expert, for example using the number of transactions confirmed by the expert to be

illegal as a criterion in model training. Another possibility is to allow the expert who uses the system to adjust some parameters by hand. For example, parameters controlling fuzzy number mutation (Δ_x , m_x and M_x) may be tuned in order to ensure that the model obtained during training satisfies certain constraints;

- using Genetic Multi-Objective Optimization (GMOO) in order to balance conflicting requirements (maximizing the number of detected frauds, but at the same time limiting the number of transactions suggested for review by human expert);
- implementing computationally-intensive parts of the algorithm (for example, evaluation function calculation) using CUDA architecture in order to take advantage of massive parallelism available on modern GPUs;
- allowing more complex subgraph patterns to be used. One of the possible approaches can be to implement evolving graph structure (not only evolutionary optimization of subgraph model parameters);
- alternative methods for searching for matching subgraphs. Currently all matching subgraphs are evaluated at each level of the hierarchical pattern. Using non-deterministic methods may be hard to avoid, especially if more complex subgraph patterns will be used.

Another important aspect of further research is to obtain real-life data from financial institutions and test the proposed method on such data.

REFERENCES

- [1] Th. Fetz, J. Jager, D. Koll, G. Krenn, H. Lessmann, M. Oberguggenberger and R. Starkm "Fuzzy Models in Geotechnical Engineering and Construction Management" *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, no. 2, pp. 93–106, 1999.
- [2] O. Hasancebi and F. Erbatır, "Evaluation of crossover techniques in genetic algorithm based optimum structural design" *Computers & Structures*, vol. 78, no. 1-3, pp. 435–448, 2000.
- [3] J. Korczak, W. Marchelski and B. Oleszkiewicz, "A New Technological Approach to Money Laundering Discovery using Analytical SQL server," in: *Advanced Information Technologies for Management AITM 2008*, J. Korczak, H. Dudycz and M. Dyczkowski (eds.) *Research Papers no. 35*, pp. 80-104, Wrocław University of Economics, 2008.
- [4] J. Korczak, B. Oleszkiewicz, "Modelling of Data Warehouse Dimensions for AML Systems," in: *Advanced Information Technologies for Management AITM 2009*, J. Korczak, H. Dudycz and M. Dyczkowski (eds.) *Research Papers no. 85*, pp. 146-159, Wrocław University of Economics, 2009.
- [5] E. M. Truman and P. Reuter, "Chasing Dirty Money: The Fight Against Anti-Money Laundering", Peterson Institute for International Economics, 2004
- [6] J. Zhong, X. Hu, J. Zhang and M. Gu, "Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms," in: *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-2 (CIMCA-IAWTIC'06) - Volume 02*, pp. 1115–1121, IEEE Computer Society, 2005.
- [7] "Debit and Credit Card fraud requires vigilance" <http://www.moneyreview.com.au/debit-and-credit-card-fraud-requires-vigilance/>
- [8] "Reserve Bank of Australia - Payments Data" <http://www.rba.gov.au/payments-system/resources/statistics/index.html>
- [9] "Russia reports \$3.8 trillion in suspect transfers: report," <http://www.reuters.com/article/2010/12/06/us-russia-economy-money-idUSTRE6B516Z20101206>