The CSI Journal on Computer Science & Engineering

The CSI Journal on
Computer Science and Engineering
Vol. 16, No. 2, 2019
Pages 1-13
Regular Paper

A new framework for combining supervised and semi-supervised methods in fraud detection

Abdollah Eshghi Mehrdad Kargari

Tarbiat Modares University, Industrial and systems engineering, Tehran, Iran

Abstract

Every year a large amount of money is lost due to fraud in financial institutions. Detecting frauds is a complicated task and limiting fraud detection systems to certain kinds of detection methods like supervised or unsupervised methods does not seem efficient. In this paper, a combined framework for fraud detection systems, consisting of both supervised and semi-supervised methods in three main components namely rule-based component, trend-analysis-based component and, a scenario-based component is proposed. The rule-based component is the supervised part of the framework and a decision tree, which is a cost-insensitive classification algorithm, is used for this component. In the trend-analysis-based component, which is the semi-supervised part of our proposed framework, the normal behavior of users are modeled and the extent of dissimilarities of newly arrived transactions are calculated. Finally, in the scenario-based component, which is another semi-supervised part of the proposed framework, the extent of similarities of the sequence of transactions with the known fraud scenarios are calculated. The final result is obtained through combining the results of all these three components using a bagging method. Combining the outputs of all these components together using the proposed bagging model, rather than detecting more frauds, the results are more stable and the number of false alarms is reduced remarkably.

Keywords: Fraud detection, supervised methods, Semi-supervised methods, Trend analysis, Bagging Fraud detection, supervised methods, Semi-supervised methods, Trend analysis, bagging.

1. Introduction

The number of frauds has been increasing with electronic card transactions rising every day. There are different statistics reported on the number or volume of frauds in different countries. For instance, billions of dollars of revenue are lost each year as a result of credit card fraud [1]. Also, according to a report published by the European Central Bank in 2014 [2], fraud has increased about 14.8% in comparison with 2011 and again according to Nilson Report [3] losses caused by fraud reached 21.84 billion dollars in 2015, which showed a twenty-percent increase in comparison with 2014. The amount of losses caused by fraud are increasing each year and, as stated in [4], the organizations in USA lose about 7% of their revenues to fraud. Hence, fraud detection methods are required to improve parallel with the increase in electronic banking transactions.

Fraud detection systems, especially in the banking industry, are always facing several challenges and solving them all is a complicated task. Some of the challenges which fraud detection systems face are as follows: there is still no best practice algorithm or method for fraud detection; the behavior of the entities involved in a fraud detection system is constantly changing [5]; the number of frauds, in comparison with non-frauds, is very rare [6], which will deteriorate the precision of data mining algorithms results; there exists no standard dataset to compare different algorithms and methods, and in many cases, there is no labelled data for analysis; on the one hand, decision making must be done in real time, on the other hand, the number of false alarms must be minimized. Fraud detection systems must be dynamic and adapt to the changes in users' and fraudsters' behavior. Users' behavior changes according to the changes in environmental variables and fraudsters' behavior changes according to the changes in users and fraud detection systems. Fraudsters always make their best efforts to have their behavior look legitimate, which

makes the process of fraud detection more complicated [7]. Modelling the fraudulent behavior is a complex task since these types of behavior are more and more sophisticated and seriously threaten the security and trust of online banking businesses [8].

There are three main categories of techniques in data mining which are applied to fraud detection; these are supervised methods, semi-supervised methods and unsupervised methods [9]. While supervised methods are straight-forward to fraud detection and their results are more understandable, they have two critical challenges if applied solely to fraud detection. First, in most cases, especially in banking sectors, there is no labelled data to apply supervised algorithms on them and, second, it is not an easy task to find distinctive labels because of the uncertainties and ambiguities in the supervision or labels [10]. Hence, using unsupervised or semi-supervised methods besides supervised methods is inevitable in fraud detection systems.

In unsupervised or semi-supervised methods, patterns of users' behavior are extracted according to several features and then new behaviors are compared with these extracted patterns, and according to the extent of its deviation from the extracted patterns, the fraud detection system (FDS) decides to whether alert it as a fraudulent behavior or not.

In this paper, we introduce a framework for fraud detection, which combines supervised models with semi-supervised models and has the capability to detect new frauds while making lower false alarms. It can start its work with just legal dataset and during the time its accuracy will improve. By combining the results of supervised and semi-supervised methods rather than detecting new types of fraud, it can decrease the alarm rates too. This framework consists of three main components, namely rule-based component, trend analysis component and scenario-based component. In the rule-based component, any supervised algorithms can be used; here, a decision tree algorithm is used. In the trend analysis component, a semi-supervised method for modelling the normal behavior of users in different time periods is employed and the degree of dissimilarity is estimated by comparing the newly-arrived transactions to their historical trends. Finally, in the scenario-based component, known scenarios of frauds are utilized according to previous experiences and the cases mentioned in the literature.

The remainder of the paper is organized as follows: Section 2 is the literature review. In section 3 we present our proposed methodology to create the combined model. In section 4 the experiments and evaluation is presented. The results are in section 5 and finally, discussion is presented in section 6.

2. Literature Review

While most of the real-life fraud detection approaches adopt black box models [11] and practical implementations are rarely reported [7], the number of academic studies is remarkable. Most of the proposed fraud detection techniques apply a concept usually referred to as Outlier Detection. An outlier is an observation that is so different from other observations and seems to have been generated by a different mechanism [12] and outlier detections refer to the process of finding outlier cases [13].

There is increasing demand for robust and intelligent user profiling technologies [5], especially in the banking and in fraud detections systems. Making profiles for user behaviors is a key technology to respond to the needs of real-time fraud detection [5] because according to [14], real users may gradually change their behavior over a longer period of time. In [15], the profiling method has been used for credit card fraud detection. The statistical representation of user behaviors is carried out through user profiles. Profile-based fraud detection systems always employ thresholds for modelling. Although deploying such systems are easy and straightforward for preventing so many false alarms which are prevalent in such systems, due to the lack of support for different behavioral profiles between monitored accounts [16], it is necessary to calculate and assign profile thresholds for each user; thus, a proactive fraud detection system will be achievable [5]. According to [17], in order to decrease false alarm rates, the thresholds for different profiles of users must vary by time as both legitimate and fraudulent user behaviors change over time (e.g. interest rates, seasonal/monthly variations, new fraud attacks etc.) [5].

Cahill [17] lists the key elements of modern fraud detection systems as event-driven processing, memory, learning and, self-initializing. Event-driven processing facilitates the detection of fraud as it is happening, rather than at fixed points in time and unrelated to account activity [5]. The memory means involving all past data in profile processing for a user (but not necessarily all with the same weights); learning is the ability of the fraud detection system to adapt with new customer behaviors; and finally, self-initializing is the ability to have meaningful profile thresholds for newly opened accounts. Such issues can be solved to some extent by using profile-based methods.

As mentioned in [5], profile-based fraud detection methods can be categorized into two main processing models: time-oriented and action-oriented or event-driven. In time-oriented models, transactions are accumulated over a specified time period (for example hours, days, etc.), and then batch processing will be done over them. In the event-processing model, new transactions are examined as soon as they arrive. Despite the fact that action-oriented models seem to perform better results in comparison to time-oriented models [18], the time-consuming process of reading and writing current profiles and thresholds from and into data storage justifies the time-oriented approach, particularly for applications in which time is not critical. However, in banking applications in which time is critical, action-oriented methods may produce more false alarms.

In [18], a method has been proposed for calculating profiles. In this method, a window time of the fraud less account activity is regarded as a base for calculating the users' profiles. Bolton and Hand [19] have proposed a model in which breakpoint analysis is used for detecting the trend of spending changes in customer behavior. The aggregation of customer behavior over a series of feature variables is proposed. In [20], an experimental comparison between several algorithms has been done and some questions like "what algorithm should be selected for fraud detection?" and "what window time length is appropriate for updating the model?" have been answered. In [8], user profiles are calculated by building a contrast vector for each transaction based on its customer's historical behavior sequence and an algorithm has been proposed for mining contrast patterns and distinguishing fraudulent patterns from genuine ones. In [21] and [22], a sequence of customer actions is regarded as a profile for the customer and fraud cases are

detected using the Hidden Markov Model method. In [23], by combining anomaly detection and misuse detection models, a hybrid model has been proposed, and the similarity of an incoming sequence of transactions to both fraudulent and non-fraudulent ones is determined through utilizing segment alignment.

In [24] a framework for transaction aggregation is proposed, attempting to hold customers behavior trend in aggregated features and using classification algorithms for fraud detection. Bahnsen et al. [25] have expanded the transaction aggregation strategy introduced by Whitrow et al. [24] and proposed a method to create a new set of features based on analysing the periodic behavior of the time of a transaction using the von Mises distribution. A real-life system with the capability of action-oriented systems and with a reasonable low rate of false alarm is the necessary requirement of fraud detection systems. As it was induced from [6] and [7], there are only 10 studies to date claiming to have been implemented practically. Our proposed framework is a general one, which different algorithms can be applied to each component and it is applicable in a real-world environment.

3. The proposed methodology

As depicted in Figure 1, the proposed framework is comprised of three main components namely rule-based component (RBC), trend-analysis component (TAC) and scenario-based component (SBC). The output of RBC is "Fraud" or "Not fraud". Decision Tree which is a cost-insensitive classification algorithm and is used by Bahnsen et al [25] is used for this component. The output of TAC is a risk number which is between 0 and 1. By determining a threshold, the output of this component can be divided into "Risky" or "Not Risky" transactions. If the calculated dissimilarity is less than a pre-determined threshold (θ), then the result is "Not Risky"; otherwise, it is "Risky" (equation 1).

$$TAC = \begin{cases} Risky & if\ risk \geq \theta \\ Not\ Risky & if\ risk < \theta \end{cases} \quad (1)$$

By extracting different trends of customer's behavior in different periods, the extent of dissimilarity of each transaction according to its relevant trend is calculated. Then after assigning weights to each dissimilarity, the weighted average of all dissimilarities is calculated and using equation (1), the result of TAC component is gained. While a transaction is passing the RBC and TAC steps, it will also be examined against the SBC. A repository of fraud scenarios was formed and the extent of the similarity of the new sequence to fraud scenarios will be estimated by holding a sequence of transactions in a buffer. The output of SBC is as equation 2:

$$SBC = \begin{cases} Risky & if\ any\ scenario\ occurs \\ Not\ Risky & if\ no\ scenario\ occurs \end{cases} \quad (2)$$

The final result of a transaction (Fraud or Not-Fraud) is estimated by combining the outputs of SBC, RBC, and TAC. Combining the outputs is done with a bagging model as described in this paper.

3.1. RBC

According to SAS's 2015 report of fraud detection systems, one of the main components which is needed in any fraud detection system is a rule-based component. Any classifier algorithm can be used for this component. Bahnsen et al in [22] have used three cost-insensitive classification algorithms namely decision tree (DT), logistic regression (LR) and random forest (RF) algorithm and have trained the algorithms using different sets of features including only raw features, only aggregated features and both raw and aggregated features. Their results showed that training algorithms using both raw and aggregated features will gain more preferable results. Here the DT algorithm which has the best result in [22] compared to LR and random FR is used and the algorithm is trained with both raw and aggregated features. The typical raw features employed in this paper are as summarized in Table 1.

Table 1. Typical raw features used in this paper

Attribute Name	**Description**
Transaction ID	Transaction Identifier
Account number	Customer Identifier
Card number	Card Identifier
Transaction date	Time and date of the transaction
Amount	Amount of the transaction
Channel type	ATM, POS, Internet, Mobile …
Merchant Code	Merchant Identifier
Merchant Group	Business of the merchant
Account Balance	Remaining amount in customers account
Gender	Female or male
Age	Card holder age
Bank	Issuer Bank

According to the order of combination of raw features and the previous time periods, several aggregated features can be extracted. For extracting the aggregated features, we used the method which is used by Bahnsen et al. in [25], they set the period of times for aggregating the features to 1, 3, 6, 18, 24, 72 and 168 hours; we also regard 720 hours in this paper. Figure 2 shows the order of the combination of the amount, merchant group and channel type for making aggregated features. According to this figure, 5×4×8=160 new aggregated features can be extracted. For example, the sum of the amount of transactions in the last 24 hours is a new aggregated feature, or the sum of the amounts of transactions in the last 24 hours which are made in the same merchant group is another integrated feature. All the other integrated features can be induced using Figure 2.

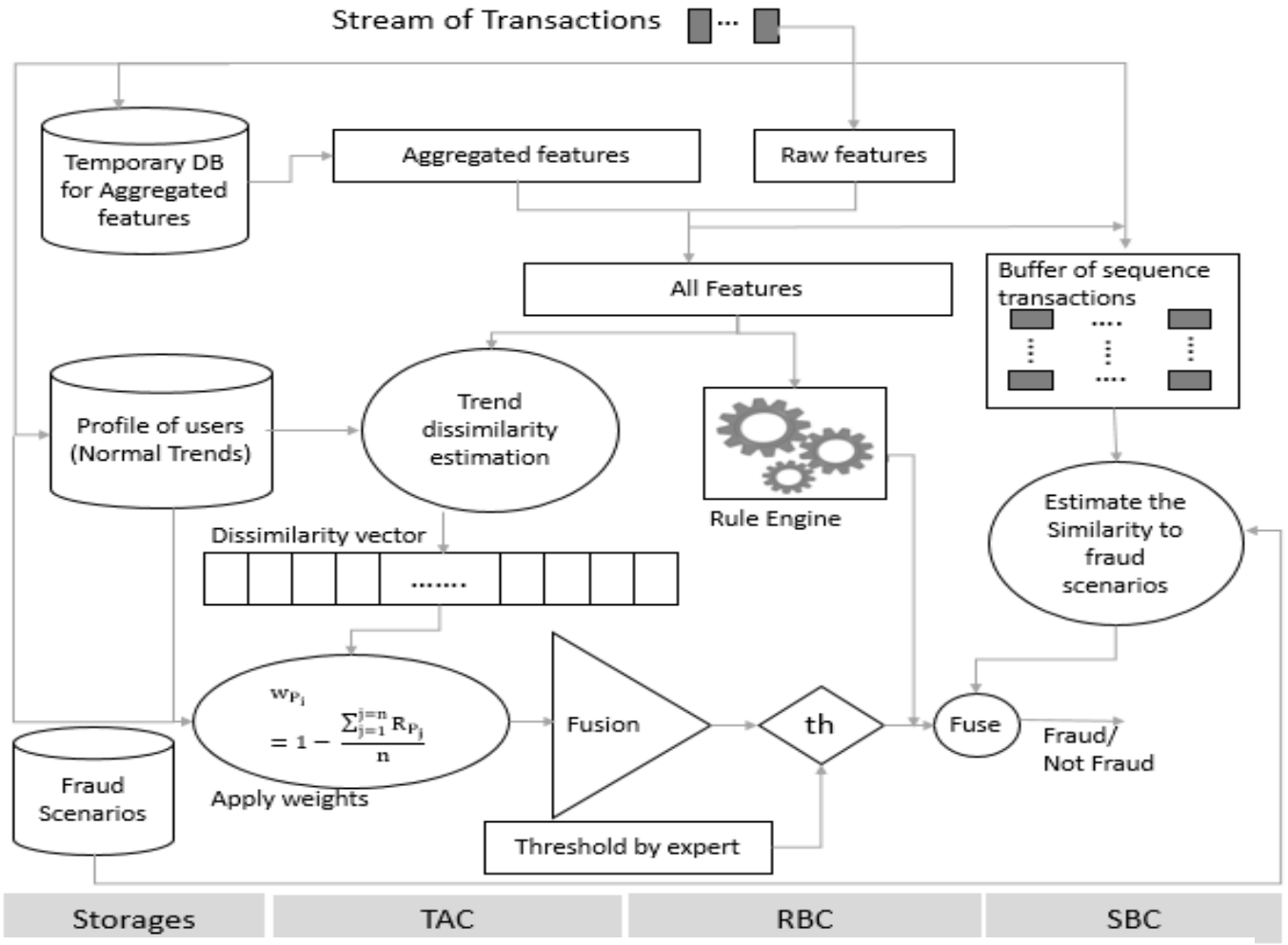

Figure 1. The big picture of the proposed fraud detection framework

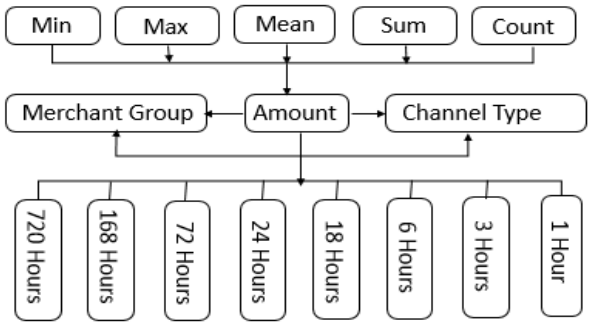

Figure 2. Aggregated features in different time periods on amount of transaction

3.2. TAC

As mentioned before, TAC is the semi-supervised component of the proposed frameworks and in this component, just the none-fraud transactions of customers are gathered and their historical normal trends are extracted. Next, the dissimilarity of any new transaction is estimated according to these historical trends. The historical transactions of customers for two main raw features namely "amount of transaction" and "daily time of transaction" against two other raw features namely "channel type" and "merchant group" in different periods is modeled. As mentioned before, TAC is the semi-supervised component of the proposed frameworks and in this component, just the none-fraud transactions of customers are gathered and their historical normal trends are extracted. Next, the dissimilarity of any new transaction is estimated according to these historical trends. The historical transactions of customers for two main raw features namely "amount of transaction" and "daily time of transaction" against two other raw features namely "channel type" and "merchant group" in different periods is modeled.

3.2.1 Extracting normal behavior trends

Time behavioral and amount behavioral trends are extracted as depicted in Figure 3. Here, the trend of times of transactions of cards during the day and trend of transactions amounts of cards against channel type and merchant group, during four different time periods of "last month", "last 3 months", "last 6 months", and "last 12 months" are extracted. Also, regarding the merchant group and channel type, 16 new aggregated time features and 16 new aggregated amount features are extracted (4×4=16: the first 4 is for "transactions' times/amounts", "transactions' times/amounts" & "same merchant group", "transactions' times/amounts" & "same channel type", "transactions' times/amounts" & "same merchant group" & "same channel"; the second 4 is for the time periods). When a new transaction arrives in TAC, about 32 normal trends will be retrieved and the extent of deviation of the transaction will

be calculated against them.

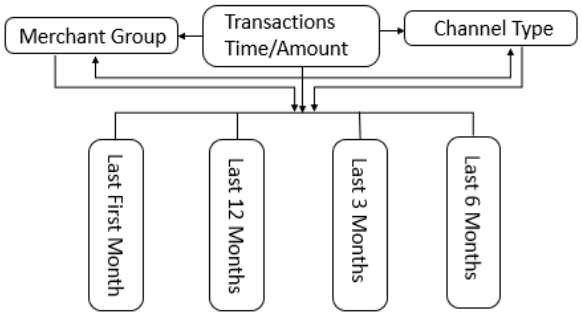

Figure 3. Aggregated time/amount features extraction

3.2.1.1. Extracting trends of times

Arithmetic mean is not a good idea for extracting the trend for the time of transactions, as it does not take the periodic behavior of the time feature into account. In [25] the von Mises distribution, which is a distribution of a wrapped normal distributed variable across a circle, is employed [26]. Here, we used our function for estimating the degree of dissimilarity of a transaction time to its previous trends.

In Figure 4, a sample of customer purchase daily time distribution is shown. As we can see, there are fewer transactions in hours 0 to 7 in comparison with other hours in the day. Consequently, the dissimilarity membership function (since we aim to detect abnormal behaviors, we need to estimate the degree of dissimilarity and not similarity) is depicted in Figure 5. The range between t1 and t2 is completely dissimilar to the customers' normal distribution time (for this case t1 is 0 and t2 is 6). For times between 'a' and t1 or between t2 and 'b', the degree of dissimilarity is in accordance with a slope line. The fuzzy dissimilarity membership function for time distribution trends is shown in Figure 5.

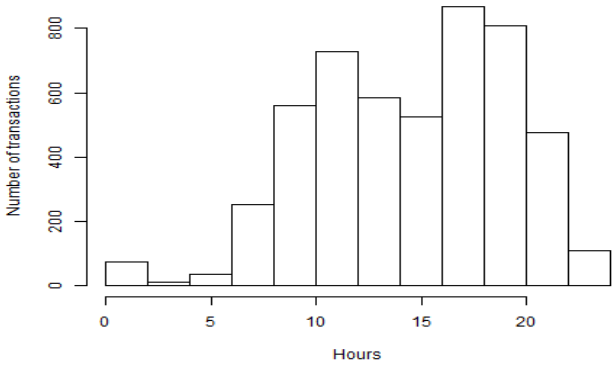

Figure 4. Customer purchase time distribution

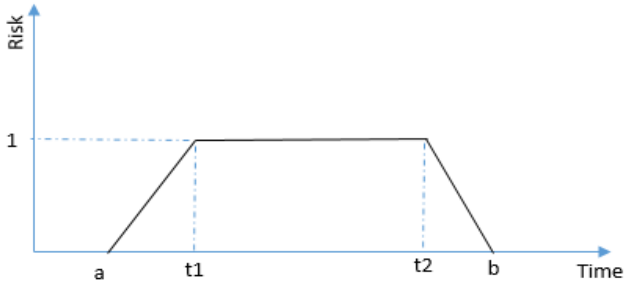

Figure 5. Time dissimilarity fuzzy membership function

$$time\_risk = \begin{cases} 1 & t1 \le t \le t2 \\ \frac{1}{t1-a}(t-a) & a \le t \le t1 \\ \frac{1}{t2-b}(t-b) & t2 \le t \le b \\ 0 & o.w \end{cases} \quad (1)$$

For determining the marginal points (a, t1, t2, b) the box plot rule which is a simple statistical technique for detecting univariate and multivariate anomalies [13] is used. With a box plot, we can graphically depict the data using summary attributes. Min is the smallest non-anomaly observation; Q1 and Q3 are lower and upper quartiles respectively, and Max is the largest non-anomaly observation. The quantity Q3 – Q1 is called the Inter Quartile Range (IQR). Applying the box plot, we can indicate the limits beyond which any observation will be treated as an anomaly (Figure 6).

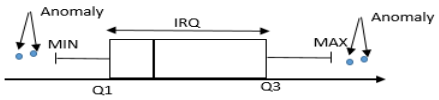

Figure 6. Box plot for detecting anomalies

A data instance which lies more than Q3 + 1.5*IQR will be treated as the soft outlier and the quantity Q3+1.5*IQR is called the soft threshold. Here, similarly, a data instance which lies at more than Q3 + 3*IQR will be treated as the hard outlier and the quantity Q3+3*IQR is called the hard threshold. As for the time distribution, we regard Q1 as the soft left threshold (a), Q3 as the soft right threshold (b), Q1+2 as the left hard threshold (t1) and Q3-2 as the right hard threshold (t2). A new transaction time is out of the confidence interval range with a degree between 0 and 1.

A matrix of customer's time trends must be held so that we can estimate the extent of dissimilarities of newly arrived transactions (equation (4)).

$$\text{ttv(i)} = \left\{\left(a_i^{j,k}, t_{1,i}^{j,k}, t_{2,i}^{j,k}, b_i^{j,k}\right)\right\}_{j=1,k=1}^{N,M} \quad (2)$$

N is the number of merchant groups and M is the number of channel types. According to the amount of spending, we extracted 12 merchant groups, so for each customer N is less than or equal to 12. The channel types are internet channel, physical point of sell channel, and mobile channel; hence, M is always less than or equal to 3. For each customer and time period (last first month, last three months, last 6 months and last 12 months) at most 3×12=36 trends must be held as time distribution profiles. Thus, there exist at most 4×36=144 time profiles for each user. Each trend consists of four marginal numbers (a, t1, t2, b) which must be placed in equation (3) to calculate the dissimilarity.

3.2.1.2. Extracting trend of amounts

According to a customer's habit of the amount of spending in different periods or via different channels, his/her outlier amounts will be specified and the thresholds will be assigned. The dissimilarity fuzzy function for amount is similar to Figure 7. The dissimilarity of amounts smaller than the soft threshold equals zero and the dissimilarity of amounts between the soft and hard threshold is in accordance with a

line and their values are between 0 and 1, while the dissimilarity of amounts larger than hard thresholds is 1.

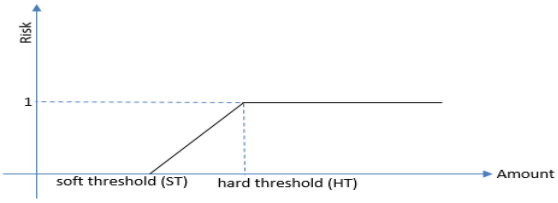

Figure 7. Amount dissimilarity fuzzy membership function

Soft and hard thresholds are the border of outliers in customers' spending behavior. These have different values for different customers. Even for one customer, the thresholds are different for different periods of time or for different channels or businesses. Since low amounts (amounts far less than Q1) are not important for fraud detection, we did not consider them in the amount of fuzzy membership function. The region between Q1 −1.5*IQR and Q3 +1.5*IQR contains 99.3% of the observations [13], hence for our case the normal region is between 0 and Q3 + 1.5*IQR which contains about 99.6% of the observations. The amount of risk for the fuzzy membership function for "Amount Risk" is calculated as equation (5).

$$amnt\_risk = \begin{cases} 1 & trAmount > HT \\ \dfrac{\text{trAmount} - \text{ST}}{\text{HT} - \text{ST}} & \text{ST} \leq \text{Amount} \leq \text{HT} \\ 0 & o.w \end{cases} \quad (3)$$

In the above formula, the trAmount is the amount of the transaction, and ST and HT are soft threshold and hard threshold, respectively. If the transaction amount is more than the hard threshold, then its risk is 1. For amounts between the hard and soft thresholds, the risk is in accordance with the line, which connects the two points with coordinates (0, ST), and (HT, 1). In other cases, the risks will be zero. It is natural that the steep of the line and also the ST and HT are different for different users and even for a user in different channels, businesses or different periods. A matrix of customers' amount trends must be held so that we can estimate the extent of dissimilarities of newly arrived transactions (equation (6)).

$$atv(i) = \left\{\left(ST_i^{j,k}, HT_{1,i}^{j,k}\right)\right\}_{j=1,k=1}^{N,M} \quad (4)$$

N is the number of merchant groups and M is the number of channel types. For each customer and each time period, at most 3×12=36 trends of amounts must be held as amount profiles. So, for each user, there is at most 4×36=144 amount profiles. Each trend consists of two marginal numbers (ST, HT) which must be placed in equation (5) to calculate the dissimilarity.

3.2.1.3 The customer profile

Each customer profile is a package consisting of his/her time-trend-vector (ttv) and amount-trend-vector (atv), each of which has at most 144 trends. A profile package of a user (i) for a specific time period (tp) is as shown in equation (7):

$$Prof_i^{tp} = \{(atv(i))^{tp}, ttv(i)^{tp}\} \quad (7)$$

$$tp = \{1\ month, 2\ months, 3\ months, 6\ months, 12\ months\}$$

When a new transaction arrives at the system according to its customer ID, its merchant group and its channel type, 32 of his profiles and 32 of general profiles will be retrieved (Table 2) and the new transaction will be compared to them in order to estimate the extent of dissimilarity. Relation (8) is the process of retrieving related profiles for an arrived transaction.

$$Retrieved\_Prof_i^{tp}(ch, mer) = \{(atv(i, ch, mer))^{tp}, ttv(i, ch, mer)^{tp}\} \quad (8)$$
$$tp = \{1\ month, 2\ months, 3\ months, 6\ months, 12\ months\}$$

Table 2. Extracted normal trends of cards

trends	Description
1-4	Normal trend of transactions amounts of a card in last first, three, six and twelve months
2-8	Normal trend of transaction amounts of all cards in last first, three, six and twelve months
9-12	Normal trend of transactions amounts of a card through a specific channel type in last first, three, six and twelve months
13-16	Normal trend of transactions amounts of all cards through a specific channel type in last first, three, six and twelve months
17-20	Normal trend of transactions amounts of a card through a specific merchant group in last first, three, six and twelve months
21-24	Normal trend of transactions amounts of all cards through a specific merchant group in last first, three, six and twelve months
25-28	Normal trend of transactions amounts of a card through a specific merchant group and a specific channel type in last first, three, six and twelve months
29-32	Normal trend of transactions amounts of all cards through a specific merchant group and a specific channel type in last first, three, six and twelve months
33-36	Normal trend of transactions times of a card in last first, three, six and twelve months
37-40	Normal trend of transactions times of all cards in last first, three, six and twelve months
41-44	Normal trend of transactions times of a card through a specific channel type in last first, three, six and twelve months
45-48	Normal trend of transactions times of all cards through a specific channel type in last first, three, six and twelve months
49-52	Normal trend of transactions times of a card through a specific merchant group in last first, three, six and twelve months
53-56	Normal trend of transactions times of all cards through a specific merchant group in last first, three, six and twelve months
57-60	Normal trend of transactions times of a card through a specific merchant group and a specific channel type in last first, three, six and twelve months
61-64	Normal trend of transactions times of all cards through a specific merchant group and a specific channel type in last first, three, six and twelve months

3.2.1.4 Estimating dissimilarities

After retrieving profiles, by placing the threshold parameters

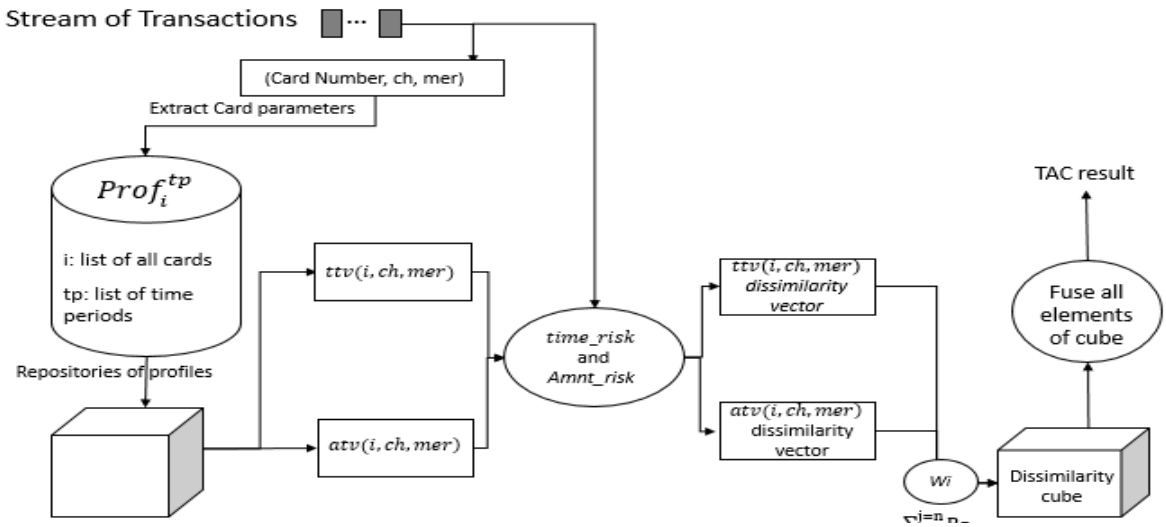

Figure 8. The process of TAC

of the retrieved profiles in equations (3) and (5), the vectors of dissimilarities for amount and time is acquired. The values of the elements of the dissimilarity matrix are numbers which are between 0 and 1. Number 0 means no dissimilarity and 1 means the greatest dissimilarity.

3.2.1.5 Giving weight to profiles

As depicted in Figure 1, after calculating the dissimilarities, a weight is given to each of the profiles. Each person has some dominant behavioral characteristics. The weights of dominant characteristics are more than the others. In order to calculate the weights, the historical calculated dissimilarities for each behavioral characteristic are regarded in a certain window of time (for example the customer's last 10 transactions). Since the thresholds are updated after each transaction, the previous values are expected to be low for genuine previous transactions, and if it is not so, it is concluded that the customer's behavior of this profile is not certain or its profile has not been modeled correctly. Hence, in order to moderate the effect of this profile, the weight is regarded as low and vice versa. The formula for calculating the weight is as equation 9:

$$w_{P_i} = 1 - \frac{\sum_{j=1}^{j=n} R_{P_j}}{n} \quad (9)$$

In this formula w_(P_i)is the weight for the ith characteristic of a customer profile; R_(P_j)is the risk of j previous transactions for the selected characteristic, and n is the window size. The result is always between 0 and 1. For example, if all last n transaction risks are 0, then the weight will be 1 and if all were 1 then the weight will be 0.

3.2.1.6 Fusion of dissimilarities

The matrix of dissimilarities must be fused in order to reach a final value for the extent of dissimilarity. Panigrahi et al. [27] have used Dempster-Shafer Adder (DSA) for combining the evidence of different rules. DSA method can also be used here but in order to hold simplicity, we used a simple method based on the weighted average for combining the results. The process is demonstrated in Figure 8. At first, the Matrix of dissimilarities is multiplied in their weights (weights are assigned by experts). Next, those dissimilarities which are greater than a specified threshold are selected for making weighted average on them. The final result is a number between 0 and 1. The whole process of TAC is depicted in Figure 8.

3.3. SBC

When a new transaction arrives, it will be compared to known fraudulent scenarios and the degree of similarity of the transaction to known fraudulent scenarios is estimated. The aim of any fraudster is to get the most benefit in a short time and with minimum risk [28], [29]. Based on this hypothesis, several scenarios are designed in order to be applied on arrived transactions. The scenarios are designed for when a fraudulent transaction is intelligently and the fraudster tries to behave in a way too similar to the genuine behavior of a card owner. In such situations, the transaction can pass the TAC and RBC steps and we hope it can be stuck in SBC. The SBC targets a sequence of transactions and watches out the changes across them.

In one hand to get the most benefit, a fraudster tries to withdraw large amounts of money, in another hand to minimize the risk of being stuck he/she will regard the thresholds and tries to not withdraw amounts larger than card owner hard or soft thresholds. In order to fix this contradiction a fraudster tries to perform his plan in a sequence of transactions but since the fraudster is going to do his plan in a short time, by regarding the sequence of transactions in a fixed time interval, the anomaly in the sequence can be detected. As shown in equation (10), in a sequence of transactions and in a short interval of time, the sum of amounts must be less than the card owner's hard threshold. Here R is the number of transactions in a sequence. tR is the time of the Rth transaction and t1 is the time of the first transaction in a sequence. T is a limitation for the fraudster and is the minimum time interval of R successive transactions in card holder's historical data.

$$\begin{gathered} \sum_i^R Amount_i \le Customer_{hard-threshold} \\ Amount_i < Customer_{hard-threshold} \qquad i = 1..R \\ t_R - t_1 < T \end{gathered} \quad (10)$$

A sample of testing a transaction against a fraudulent scenario is shown in Figure 9. While each transaction is normal when it is regarded by itself, the sequence is beyond the soft and

hard thresholds of the cardholder.

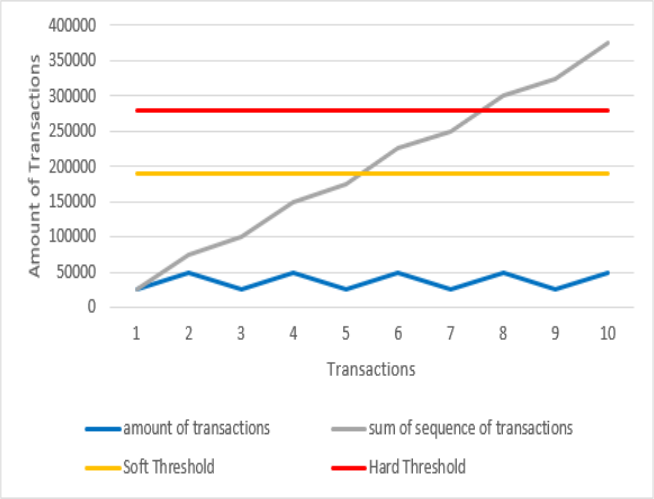

Figure 9. A sequence of transactions, which their sum of amount is beyond the soft and hard thresholds

4. Experiments and evaluation

We built the online banking risk management framework. It incorporates the rule-based component, trend-analysis component and scenario-based component for online banking fraud detection. The aims of the experimental evaluation are to find an optimized combination for the results of this three components and also to compare the performance and the degree of accuracy of each component alone and then all together.

4.1 Data

The dataset deployed in this research is provided by an Iranian private bank and consists of cards data from February 2015 to January 2016. It consists of 900,000,000 transactions belonging to 1100,000 cards. Each transaction has 12 raw attributes as shown in Table 1, plus 160 aggregated attributes as described in section 3.1. There are 1926 fraudulent transactions belonging to 580 cards in the dataset which are called victim cards here. The number of all the transactions (fraudulent plus genuine transactions) made by the victim cards is 683100. A visual overview of the available data is shown in Figure 10.

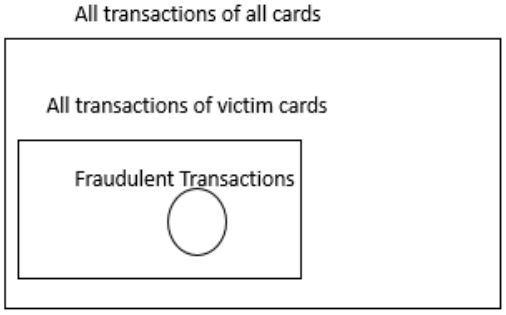

Figure 10. An overview of the available data

The existing dataset has label only for the transactions of the victims. These fraudulent transactions are reported by the card owners and receive their label as fraudulent after being assessed by auditors. The other transactions in our dataset have no label, so we do not know if they are fraudulent or not. For evaluating the rule-based component, sample datasets (train and test dataset) are created in two ways:

Random_transactions: From the transactions of victim cards, 512000 (about 75 percent) transactions were selected randomly for the train (rand_tr_train) and the remaining was set for the test (rand_tr_test).

Random_cards: From the 580 victim cards, 420 cards (75 percent of the victim cards) were selected randomly and set their transactions for train set (random_card_train), and set the next remaining 140 cards transactions for test set (random_card_test).

Selecting the train and test set transaction in such a way help us to know how the performance of RBC and SBC can be affected when new transactions from new cards are coming to the system. Furthermore, it helps us to know how having historical data can improve the results of a fraud detection system.

4.2 Experimental settings

Alert volume and detection rate are two main metrics for evaluating the performance of an online banking fraud detection system [8]. Since every triggered alert has to be investigated manually for further investigation and it is a labor-intensive work, a fraud detection system attempts to reduce the number of alerts. The fraud detection system also tries to increase the detection rate, which is the percentage of detected fraud by the system.

5. Results

In this section, the experiments are conducted on both Random_transactions and Random_cards datasets. At first, the results of using each component solely is shown and then the output of the framework, which is gained from combining the results of all three components, is analyzed.

5.1 Results of RBC, TAC and SBC

As mentioned in section 4.1, two datasets namely Random_cards and random_transactions are used for evaluating the results. The results for these two datasets is shown in Table 3. As shown, for RBC the detection rate in Random_transactions dataset is better than detection rate in Random_cards dataset (85.41 vs 60.74). This despite the fact that the alarming rate in Random_cards dataset is more than the alarming rate for Random_transactions dataset (3.99 vs 2.92). It can be concluded that the rules provided by the RBC cannot be generalized especially for newly arrived transactions from new cards.

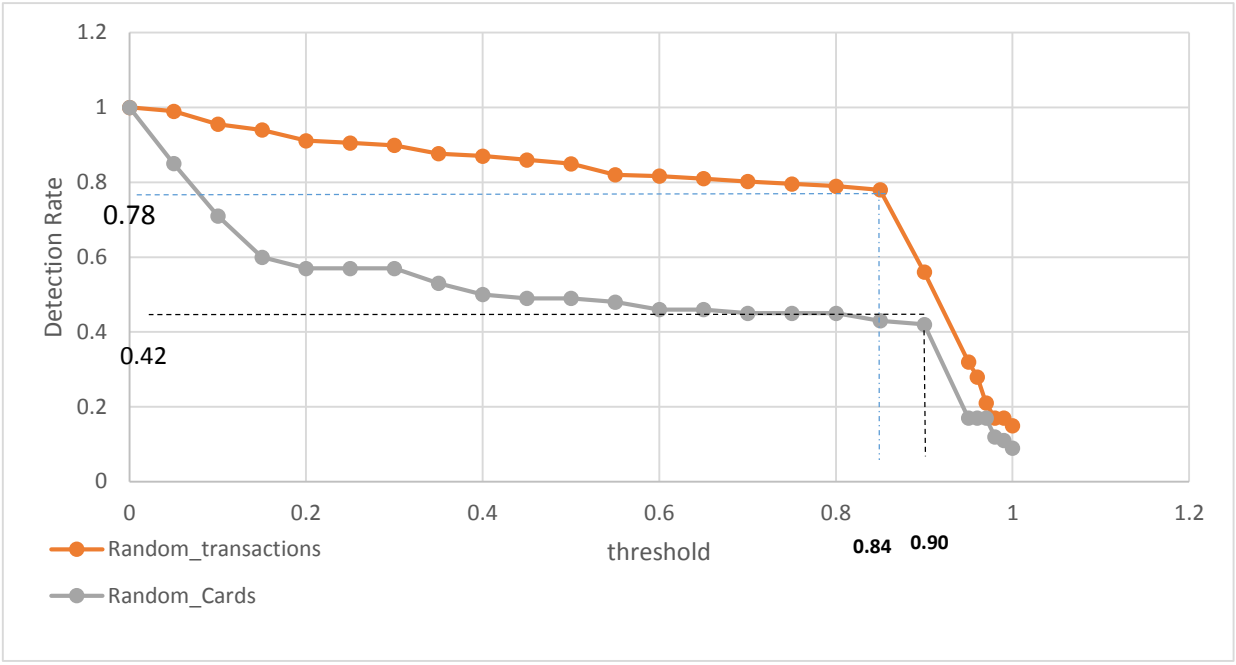

Figure 11. Optimized threshold for TAC i

Table 3. detection rate vs alarm rate for two sample datasets (Random_transactions and Random_cards) in RBC, TAC and SBC

		Random_transactions	Random_ Cards
RBC	Detection Rate	85.41%	60.74%
	Alarm Rate	2.92%	3.99%
TAC	Detection Rate	78.14 %	42%
	Alarm Rate	1.52%	2%
SBC	Detection Rate	4.93%	4.23%
	Alarm Rate	0.11%	0.13%

As shown in equation (1) the output of the TAC component is depended to the value of θ, which is a threshold for announcing a transaction as Risky or Not-Risky. By depicting the threshold versus the detection rate (Figure 11), the optimized threshold for TAC in Random_transactions dataset is 0.84 and for θ=0.84 the detection rate is 78%. The optimized value of θ in Random_cards dataset is 0.9 and the detection rate for this threshold is 42%. Figure 12 shows that for detection rate equals to 78.14 in Random_transactions dataset the alarming rate is 0.0152. The alarm rate is 0.02 for detection rate equal to 42% in Random_cards data set (Table 3). Although for thresholds less than 0.84 in Random_transactions dataset and less than 0.9 in Random_cards dataset the detection rate is higher, the alarming rate also will be higher which is not desirable. In TAC As RBC, having historical transactions of cards has great impact on the correctness of results.

Both detection rate and alarm rate are low in SBC component. In SBC like RBC and TAC, the Random_transactions results are better than the results in Random_cards dataset. A remarkable point in SBC is that most of the frauds detected by SBC are those, which cannot be detected by RBC and TAC.

5.2 Combined result of the framework

The combination of results of the three main components of the framework always takes one of the eight states shown in Table 4 and Table 5. The percent of all transactions occurring in each state is shown in the fourth column of the tables and the number of frauds in each state is shown in column 5 of the table. Row one of the table is a state where the outputs of RBC, TAC, and SBC are not-fraud (NF), not-risky (NR) and not-risky respectively. About 95.87 percent of arrived transactions are identified as not fraud and not risky by RBC, TAC and SBC. However, there are 143 fraud cases in this group. In other word, 143 fraudulent transactions are not detected by any of these components.

To reach a final belief about a transaction (fraud or not fraud) the outputs of RBC, TAC and SBC are combined in nine different ways as shown in Table 6. Row C1 of the table is a combination model in which all components have the same weight and if only one component output is fraud or risky, then the result will be fraud. In row C2, all component have the same weight, but if at least 2 component's output are risky or fraud then the final result will be fraud. In row C3, all components have the same weight but for announcing the result as fraud, all three components output must be fraud or risky. In rows C4, C5 and C6 RBC, TAC and SBC have higher weight respectively. In rows C7, C8 and C9 SBC, TAC and RBC have lower weights respectively. In Table 7 and Table 8, the results of these nine combination models, in terms of detection rate and alarm rate on Random_transactions and Random_cards are shown. As said previously more detection rate and lower alarm rate is desirable for fraud detection systems. For Random_transaction dataset, C1 has the most detection rate but its alarm rate is high (3.86%). Although C2 and C5 have both the least alarm rate, the detection rate of C5 is far better than C2 (79.12% vs 58.71%). Compared to RBC, TAC and SBC alone, C1 and C7 have better detection rates

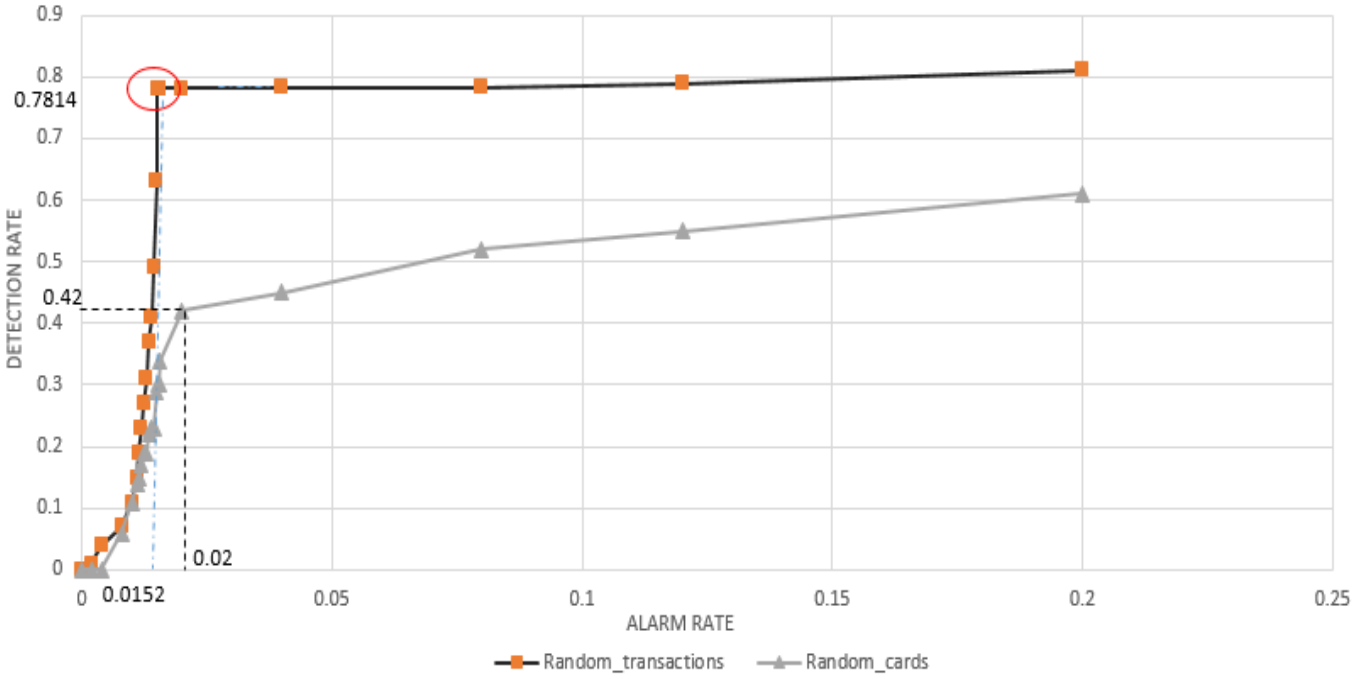

Figure 12. Alarm rate vs detection rate

Table 4. The 8 states of outputs of RBC, TAC and SBC in Random_transactions dataset

	RBC	TAC	SBC	Percent of results	Percent of fraud cases
1	NF	NR	NR	95.87%	7.42%
2	NF	NR	R	0.06%	3.94%
3	NF	R	NR	0.08%	27.89%
4	NF	R	R	0.00%	0%
5	F	NR	NR	2.50%	9.50%
6	F	NR	R	0.05%	0.98%
7	F	R	NR	1.44%	50.26%
8	F	R	R	0.00%	0%

Table 5. The eight states of outputs of RBC, TAC and SBC in Random_cards dataset

	RBC	TAC	SBC	Percent of results	Percent of fraud cases
1	NF	NR	NR	93.82%	21.61%
2	NF	NR	R	0.17%	2.80%
3	NF	R	NR	0.19%	19.67%
4	NF	R	R	0.06%	0.62%
5	F	NR	NR	3.63%	7.83%
6	F	NR	R	0.07%	1.46%
7	F	R	NR	1.98%	45.74%
8	F	R	R	0.08%	0.26%

and C2, C5, C6 and C9 have better alarm rates (lower alarm rates). For a business like bank, having low alarm rate is a critical criteria so C5 and C9 both are good choices since their detection rates are high (79.12 %, 83.07%) and their alarm rates are low (1.34%, 1.58%). In Random_cards dataset, the results are deteriorate. Here the best detection rate is for C1 (78.38%) but in a tradeoff between detection rate and alarm rate, C5 and C9 give better results. The results are also depicted for both datasets in Figure 13 and Figure 14.

Table 6. Combination models of RBC, TAC and SBC

C1	**RBC OR TAC OR SBC**
C2	(RBC AND TAC) OR (RBC AND SBC) OR
C3	RBC AND TAC AND SBC
C4	RBC OR (TAC AND SBC)
C5	TAC OR (RBC AND SBC)
C6	SBC OR (RBC AND TAC)
C7	RBC OR TAC
C8	RBC OR SBC
C9	TAC OR SBC

What is detected by the TAC is not completely overlapped with the cases that are detected by the RBC, and also the cases that are detected by the SBC is completely different from those that are detected by both RBC and TAC. Rather than finding more fraud cases by functions like C1 and C7 of the bagging model compared to TAC, RBC and SBC independently, by using the C5 function, lower alerts are emitted. Making these components to work in parallel mode is modeled by C1 function. However, since the alert rate is too important in banking system, the C5 function is more preferable.

Table 7. different combinations of the outputs of components for deciding about the final result in Random_transactions dataset

Output of components				Result of combination of all components in Random_transactions dataset								
	RBC	TAC	SBC	C1	C2	C3	C4	C5	C6	C7	C8	C9
1	NF	NR	NR	NF	NF	NF	NF	NF	NF	NF	NF	NF
2	NF	NR	R	F	NF	NF	NF	NF	F	NF	F	F
3	NF	R	NR	F	NF	NF	NF	F	NF	F	NF	F
4	NF	R	R	F	F	NF	F	F	F	F	F	F
5	F	NR	NR	F	NF	NF	F	NF	NF	F	F	NF
6	F	NR	R	F	F	NF	F	F	F	F	F	F
7	F	R	NR	F	F	NF	F	F	F	F	F	F
8	F	R	R	F	F	F	F	F	F	F	F	F
Detection Rate				92.57%	51.24%	0%	60.74%	79.12%	55.19%	88.62%	64.69%	83.07%
Alarm Rate				3.86%	1.34%	0%	3.82%	1.34%	1.39%	3.75%	3.86%	1.58%

Table 8. different combinations of the outputs of components for deciding about the final result in Random_cards dataset

	Output of			Result of combination of all components in Random_cards dataset								
	RBC	TAC	SBC	C1	C2	C3	C4	C5	C6	C7	C8	C9
1	NF	NR	NR	NF	NF	NF	NF	NF	NF	NF	NF	NF
2	NF	NR	R	F	NF	NF	NF	NF	F	NF	F	F
3	NF	R	NR	F	NF	NF	NF	F	NF	F	NF	F
4	NF	R	R	F	F	NF	F	F	F	F	F	F
5	F	NR	NR	F	NF	NF	F	NF	NF	F	F	NF
6	F	NR	R	F	F	NF	F	F	F	F	F	F
7	F	R	NR	F	F	NF	F	F	F	F	F	F
8	F	R	R	F	F	F	F	F	F	F	F	F
Detection Rate				78.38%	58.71%	0.26%	55.91%	67.75%	50.88%	75.58%	58.71%	70.55%
Alarm Rate				6.18%	1.34%	0.08%	5.82%	2.38%	2.36%	6.01%	5.99%	2.55%

Figure 13. Comparison the detection rate of combination models and RBC, TAC and SBC in Random_transactions dataset and Random_cards dataset

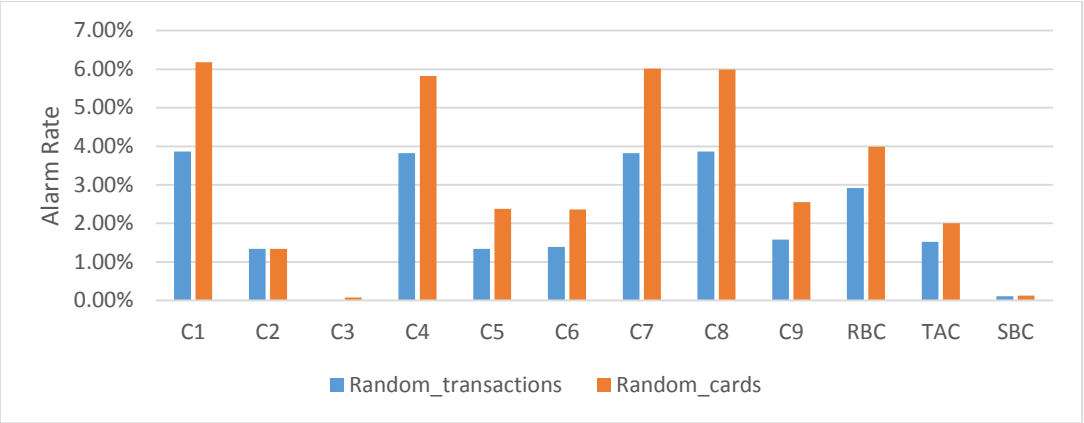

Figure 14. Comparison the Alarm rate of combination models and RBC, TAC and SBC in Random_transactions dataset and Random_cards dataset

6. Conclusion

Several entities and events such as human wisdom, analytic tools and business systems are involved in occurrence of a fraud. Effective and instant detection of sophisticated frauds requires more accurate analytic tools and algorithms. In this paper, we presented the study and practice in the real world and a framework consisting of three main components namely RBC, TAC and SBC is introduced. As mentioned in previous studies in the literature, transaction aggregation and derived attributes are also found to be useful in fraud detection. Here, rather than classifier algorithms used for extracting rules, trend analysis was carried out as a semi-supervised method and it was discovered that although semi-supervised methods have lower detection rates in comparison to classifiers or rule based methods, combining them leads to improved results. Furthermore, applying unsupervised and semi-supervised methods are necessary for fraud detection systems, especially in cases where there is no labeled data for applying supervised methods which is a common problem for most of the studies in this field. Results also revealed that using supervised methods along semi-supervised methods decreased false alarms. False alarms including FP and FN are the main obstacles for applying fraud detection methods in real world. This is still one of the research areas which researchers are advised to focus on. Another problem both for supervised and unsupervised methods is the dearth of historical data for some cards or accounts. As demonstrated in this paper, all the evaluation measures were worse for Random_cards dataset in comparison with Random_transactions dataset. Thus, self-initializing methods which underscore assigning a correct segment or group to new opened accounts or accounts with no historical transaction are recommended as yet another research area in this field.

References

[1] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," Decis. Support Syst., vol. 50, no. 3, pp. 602–613, 2011.
[2] European Central Bank, "Technical Report," 2014.
[3] "Nilson Report," 2016.
[4] R. Nisbet, G. Miner, and K. Yale, "Chapter 15 - Fraud Detection," in Handbook of Statistical Analysis and Data Mining Applications (Second Edition), R. Nisbet, G. Miner, and K. Yale, Eds. Boston: Academic Press, pp. 289–302, 2018.
[5] M. E. Edge and P. R. Falcone Sampaio, "A survey of signature based methods for financial fraud detection," Comput. Secur., vol. 28, no. 6, pp. 381–394, 2009.
[6] S. Wang, "A Comprehensive Survey of Data Mining-Based Accounting-Fraud Detection Research," in 2010 International Conference on Intelligent Computation Technology and Automation (ICICTA, vol. 1, pp. 50–53), 2010.
[7] N. Carneiro, G. Figueira, and M. Costa, "A data mining based system for credit-card fraud detection in e-tail," Decis. Support Syst., vol. 95, pp. 91–101, Mar. 2017.
[8] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, "Effective detection of sophisticated online banking fraud on extremely imbalanced data," World Wide Web, vol. 16, no. 4, pp. 449–475, 2012.
[9] C. Phua, V. Lee, K. Smith, and R. Gayler, "A Comprehensive Survey of Data Mining-based Fraud Detection Research," Comput. Hum. Behav., vol. 28, no. 3, pp. 1002–1013, 2012.
[10] A. Abdallah, A. Maarof, and M. Aizaini Maarof, "Fraud detection system: A survey," J. Netw. Comput. Appl., vol. 68, pp. 90–113, 2016.
[11] M. Carminati, R. Caron, F. Maggi, I. Epifani, and S. Zanero, "BankSealer: A decision support system for online banking fraud analysis and investigation," Comput. Secur., vol. 53, pp. 175–186, 2015.
[12] Hawkins, D, Identification of Outliers. 1980.
[13] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," ACM Comput Surv, vol. 41, no. 3, pp. 15:1–15:58, 2009.
[14] N. Laleh and M. A. Azgomi, "An Overview of a Hybrid Fraud Scoring and Spike Detection Technique for Fraud Detection in Streaming Data," in Information Systems, Technology and Management, pp. 356–357, 2009.
[15] Y. Kou, C.-T. Lu, S. Sirwongwattana, and Y.-P. Huang, "Survey of fraud detection techniques," in 2004

IEEE International Conference on Networking, Sensing and Control, vol. 2, pp. 749-754, 2004.
[16] J. Bae, H. Bae, S.-H. Kang, and Y. Kim, "Automatic control of workflow processes using ECA rules," IEEE Trans. Knowl. Data Eng., vol. 16, no. 8, pp. 1010–1023, 2004.
[17] M. H. Cahill, D. Lambert, J. C. Pinheiro, and D. X. Sun, "Detecting Fraud in the Real World," in Handbook of Massive Data Sets, J. Abello, P. M. Pardalos, and M. G. C. Resende, Eds. Springer US, pp. 911–929, 2002.
[18] P. Ferreira, R. Alves, O. Belo, and L. Cortesão, "Establishing Fraud Detection Patterns Based on Signatures," in Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining, pp. 526–538, 2006.
[19] R. J. Bolton, D. J. Hand, and D. J. H, "Unsupervised Profiling Methods for Fraud Detection," in Proc. Credit Scoring and Credit Control VII, pp. 5–7, 2001.
[20] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," Expert Syst. Appl., vol. 41, no. 10, pp. 4915–4928, 2014.
[21] S. S. Mhamane and L. M. R. J. Lobo, "Use of Hidden Markov Model as Internet Banking Fraud Detection," Int. J. Comput. Appl., vol. 45, no. 21, pp. 5–10, 2012.
[22] V. Bhusari and S. Patil, "Study of Hidden Markov Model in credit card fraudulent detection," in 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave), pp. 1–4, 2016.
[23] A. Kundu, S. Sural, and A. K. Majumdar, "Two-Stage Credit Card Fraud Detection Using Sequence Alignment," in Information Systems Security, pp. 260–275, 2006.
[24] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams, "Transaction Aggregation As a Strategy for Credit Card Fraud Detection," Data Min Knowl Discov, vol. 18, no. 1, pp. 30–55, 2009.
[25] A. Correa Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," Expert Syst. Appl., vol. 51, pp. 134–142, 2016.
[26] N. Fisher, Statistical Analysis of Circular Data. Cambridge University Press,Cambridge, UK., 1995.
[27] S. Panigrahi, A. Kundu, S. Sural, and A. K. Majumdar, "Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning," Inf. Fusion, vol. 10, no. 4, pp. 354–363, 2009.
[28] V. Vatsa, S. Sural, and A. K. Majumdar, "A Game-Theoretic Approach to Credit Card Fraud Detection," in Information Systems Security, S. Jajodia and C. Mazumdar, Eds. Springer Berlin Heidelberg, pp. 263–276, 2005.
[29] A. A. I. Alnajem and N. Zhang, "A Copula-Based Fraud Detection (CFD) Method for Detecting Evasive Fraud Patterns in a Corporate Mobile Banking Context," in 2013 International Conference on IT Convergence and Security (ICITCS), pp. 1–4, 2013.

Abdollah Eshghi received his PhD degree in Information Technology (Information Systems Management) from Tarbiat Modares University. His research interests are in the fields of machine learning, network analysis and data analyzing.
Email: a.eshghi@modares.ac.ir

Mehrdad Kargari received his PhD degree in industrial engineering from Tarbiat Modares University of Iran. He is currently an assistant professor at the department of Information Engineering, in Tarbiat Modares University. His research interests are in the fields of machine learning, artificial intelligence, IoT and their applications in health or Banking.
Email: m_kargari@modares.ac.ir

Paper Handling Data:

Submitted: 27.06.2018
Received in revised form: 11.12.2018
Accepted: 09.02.20.19
Corresponding author: Mehrdad Kargari
Affiliation of the corresponding author: Tarbiat Modares University, Industrial and systems engineering, Tehran, Iran