CREDIT CARD FRAUD DETECTION USING SELF-ORGANIZING MAPS

Vladimir ZASLAVSKY and Anna STRIZHAK

Abstract: Nowadays, credit card fraud detection is of great importance to financial institutions. This article presents an automated credit card fraud detection system based on the neural network technology. The authors apply the Self-Organizing Map algorithm to create a model of typical cardholder's behavior and to analyze the deviation of transactions, thus finding suspicious transactions.

Keywords: Payment System, Transaction, Fraud Detection, Self Organizing Map.

Introduction

Any payment system (PS) is characterized by a high level of risk in its different domains caused by great volume and number of operations, a lot of complex relations between clients and increasing speed of data transmission. In order to manage risks, PS should develop and use mathematical models to determine suspicious/ risky situations, establish scenarios for its development and evaluate consequences of their realization.

Nowadays, one of the most important and challenging problems for PS and its members becomes credit card fraud – the illegal use of credit cards by third parties. Fraudulent electronic transactions have already been a significant problem that grows in importance as the number of access points grows, especially when transactions are fully enabled on the Internet for electronic commerce.[1] Fraud detection and prevention methods are being continuously improved; however, banks all over the world lose millions of US dollars each year. Experts from Visa International predict annual growth of some fraud types up to 65%.[2] According to the Association for Payment Clearing Services, fraud losses per one credit card are expected to increase up to $11 by year 2008.

Credit card fraud is perpetrated in various ways and, generally, it is based on unauthorized write-off of funds from accounts of banks' clients – cardholders.[3] Credit card

fraud can be broadly categorized as application, 'missing in post,' stolen/ lost card, counterfeit card and 'cardholder not present' fraud.[4,5] The number of different variants of fraud is great enough, they change continuously, and new ways of fraud appear as far as protection of credit cards is improved. In the past, banks—members of PS—had solved fraud prevention problems by means of organizational measures: limits on number and amounts of cardholder's operations, monitoring of transactions in high risk countries, use of various methods for card verification, etc.[6,7] According to the theory and practice of risk management, each bank has to implement special measures in order to detect and prevent fraud in time. International payment systems, such as Visa International and MasterCard International, demand from their banks-members implementation of various measures in order to reduce the number of fraudulent operations in PS and they recommend turning from reaction methods to proaction methods for dealing with fraudulent operations with cards.[8]

In order detection and prevention of fraud to be effective, banks should develop and use in their practice special fraud detection systems targeted to reveal among stream of transactions the fraudulent ones and thus to prevent banks as well as their clients from the illegal activities of fraudsters.[9] One should develop special rules for analysis, models and methods that can describe fraudulent behavior, rules and methods of fraud prevention and generation of different decision alternatives in risky situations. Mathematical models and algorithms for classification and pattern recognition problems could be considered as a basis for such systems.

In this article, models and algorithms for detection of fraudulent operations in PS are proposed.

Problem Definition

Banks-members of PS keep databases (DB) of all their cards issued in PS. For each card, the database holds card number, account number, operational limits, current state of account (account balance) and some other data about the cardholder. Let $C_n = \{c_1, \ldots, c_k, \ldots, c_{k_n}\}$ be a set of records in DB that contains information about all cards used in PS; $c_k = (c_1^k, c_2^k, \ldots, c_s^k)$ is a record in DB, which contains information about the card c_k and its component c_1^k is a unique card number.

The processing centre of PS constantly receives information about operations carried out by cardholders (such as cash withdrawal, balance statement, purchase, etc.). The information about an operation is represented in the form of transaction message (in accordance with ISO 8583) that includes various operation parameters: card number, amount of transaction, date and time of transaction, type of operation, number of terminal, retailer identifier, etc.

Let $X_n = \{x^1,\ldots,x^i,\ldots,x^n\}$ be the set of transactions carried out in PS up to some moment t_n, where $x^i = (x_1^i,\ldots,x_j^i,\ldots,x_m^i)$ is the message about i-th transaction. Each component x_j^i holds numerical (for example transaction amount) or symbolic information (operation type, retailer code, terminal, city, etc.). An analogue (numerical) component $x_j^i \in R$. A symbolic component x_j^i (which are a majority) takes its values from some discrete set $x_j^i \in T_j = \{\tau_j^1,\ldots,\tau_j^s,\ldots,\tau_j^{s_j}\}$, where τ_j^s – s-th unique value of x_j^i. For example, component x_j^i – "terminal type" may take its values from the set T_j ={'ATM', 'POS'}, where 'ATM' means that transaction was initiated on ATM and 'POS' means that transaction was carried out in Point-Of-Sale. A symbolic field may contain from at least two values (e.g. the type of credit card) up to several hundred thousand values (as merchant code, for instance).

As time goes by the size of the set X_n grows as new transactions are executed in PS. Let us suppose that the transactions executed after moment t_n up to t_{n+k} are new ones and denote them as $x^{n+1}, x^{n+2},\ldots, x^{n+k}$.

Let $X_{c_k} = \{x^i \mid x_1^i = c_k, x^i \in X_n\}$ be the set of transactions $X_{c_k} \subseteq X_n$, executed in PS using card $c_k \in C_n$ up to moment t_n.

The problem of detection of fraudulent transactions in PS lies in classifying a new transaction $x^{n+1} = (x_1^{n+1};\ldots;x_j^{n+1};\ldots;x_m^{n+1})$ using the information about transactions X_n performed earlier and the appropriate record C_n in DB. To classify means to determine the class (fraudulent or legal) to which a transaction belongs.

Problem Analysis

A variety of methods can be applied to solve the presented problem. The simplest method used in the earliest transaction monitoring systems was control of transaction parameters $x^{n+1} = (x_1^{n+1};\ldots;x_j^{n+1};\ldots;x_m^{n+1})$; these transaction variables were compared to established levels/ thresholds.[10] The thresholds $L_n = \{l_1,\ldots,l_s,\ldots,l_{s^*}\}$ are the critical levels for the most significant parameters $x_{j_1}^{n+1},\ldots,x_{j_s}^{n+1},\ldots,x_{j_{s^*}}^{n+1}$ set by domain experts based on their experience and knowledge of the data domain. For example, if $x_{j_s}^{n+1} \geq (\leq) l_s$ for some s, then transaction x^{n+1} is classified as fraudulent.

Another approach is to apply some set of rules $R = \{R_1,\ldots,R_i,\ldots,R_{i^*}\}$ for verification of transaction x^{n+1}. Such rules describe fraudulent behavior and should be de-

fined by experts on the basis of analysis of wide range of transactions.[11] Each rule $R_i,\ i = 1,\ldots,i^*$ is a structure "$R_i : IF\ < Condition_i >\ THEN\ Transaction\ x^{n+1}\ is\ fraudulent$," which means that transaction x^{n+1} is considered to be fraudulent if it satisfies the condition of some rule $R_i \in R$.

The described methods are rather simple; however, they suffer from the following shortcomings: they detect only fixed suspicious situations established beforehand and do not take into account the variable nature of fraud; they do not consider the individual characteristics of cardholders' behavior; the control of such rule-based system is rather complex task for the expert.[12]

The authors argue that a more efficient way is to use methods such as neural networks, fuzzy logic, theory of probability, statistics and other data mining methods for automatic creation of fraudulent transaction patterns on the basis of transactions' history, its constant update and checking of all new transactions for deviation.[13,14]

In this article, the authors propose to use a type of neural network algorithm—the Self Organizing Map (SOM)—for transactional data analysis and detection of fraudulent behavior.

Principles of Transaction Classification

The described fraud detection task can be considered as pattern recognition or classification problem.[15] The set X_n of all transactions in PS is divided into two disjoint subsets: legal transactions $X_n^l \subseteq X_n$ and fraudulent ones $X_n^f \subseteq X_n$, $X_n^l \cap X_n^f = \varnothing$. If we assume that the numerical images (i.e., points in some multidimensional space) of fraudulent and legal transactions belong to different areas in this space, then it is possible to make a decision about the image of a new transaction x^{n+1}.[16]

The following two hypotheses are considered as a basis for such classification.

- *Hypothesis* H_l: Transaction $x^{n+1} = (x_1^{n+1},\ldots,x_m^{n+1})$ on card c_k is similar to all previous transactions from the set X_{c_k}, which were carried out earlier by the cardholder. If hypothesis H_l is confirmed for transaction x^{n+1}, then the transaction x^{n+1} is classified as legal and included into the set X_n^l.
- *Hypothesis* H_f: Transaction $x^{n+1} = (x_1^{n+1},\ldots,x_m^{n+1})$ is similar to earlier executed fraudulent transactions $X_n^f = \{x^i - considered\ fraudulent \mid x^i \in X_n\}$.

If hypothesis H_f is confirmed for transaction x^{n+1}, then transaction x^{n+1} is classified as fraudulent and included into the set X_n^f.

It seems reasonable to use neural network techniques for clustering and classification in order to check the proposed hypotheses H_l and H_f. The main idea is to create (and later recognize) pattern of "legal cardholder" and pattern of "fraudster" on the basis of neural network "learning" from the transactions X_n executed earlier and to develop "rules" of cardholder's behavior and fraudster's behavior. Learning algorithms allow the system to follow the cardholder's behavior and self-adapt to changes in it. If a transaction does not correspond to the pattern of "legal cardholder" or is similar to the "fraudulent" pattern it is classified as suspicious for fraud.

One of the most suitable methods of data analysis for the described problem is the Self-Organizing Map (SOM), an unsupervised neural network.[17] Neural networks of this type are often used to solve a great variety of problems from recovery of missing data to data analysis and retrieval of patterns.

Transaction Analysis with SOM

SOM Main Principles

SOM is a neural network with feed-forward topology and an unsupervised training algorithm that uses a self-organizing process to configure its output neurons according to the topological structure of the input data.[18] The self-organizing process is based on competitive training and consists in tuning the weights $w^i = (w_1^i, w_2^i, \ldots, w_q^i)$, $i = \overline{1;d}$ (q is the dimension of the input vector $a^j = (a_1^j, a_2^j, \ldots, a_q^j)$) by a method of progressive approximation using weights' values from the previous iteration [19]: $w^i(t+1) = w^i(t) + h(t) \cdot (a^j(t) - w^i(t))$; here t is the iteration number and $h(t)$ is function of the radius of the considered neighborhood. As a result from the learning process, a matrix of weights of the input connections of neurons is obtained, which allows to group subsets of input data and form prototypes (profiles):

$$W = \begin{pmatrix} w_1^1 & w_1^2 & \cdots & w_1^d \\ w_2^1 & w_2^1 & \cdots & w_2^1 \\ \cdots & \cdots & \cdots & \cdots \\ w_q^1 & w_q^1 & \cdots & w_q^1 \end{pmatrix}.$$

Testing the Hypotheses H_l and H_f

The authors propose to use the SOM for testing the hypotheses H_l and H_f in the following way.

Testing the hypothesis H_l for transaction x^{n+1} on card c_k includes the following steps:

1. Create a typical cardholder's behavior model (pattern) W_{c_k} on the basis of past transactions $X_{c_k} \in X_n^l$ executed earlier with the card c_k. This model W_{c_k} is represented as a SOM, which is the cardholder's profile.
2. Determine the similarity rate $\delta(x^{n+1}, W_{c_k})$ of transaction x^{n+1} to profile W_{c_k}.
3. Hypothesis H_l is accepted if the similarity rate $\delta(x^{n+1}, W_{c_k})$ satisfies the condition $\delta(x^{n+1}, W_{c_k}) \le \varepsilon_l$, where ε_l is some parameter.

Testing the hypothesis H_f for transaction x^{n+1} is performed in a similar to the previous scheme way:

1. Create a typical fraudster's behavior model (pattern) W_f on the basis of fraudulent transactions X_n^f executed earlier in PS and determined as fraudulent. This model W_f is also represented as a SOM, which is the fraudster's profile.
2. Determine the similarity rate $\delta(x^{n+1}, W_f)$ of transaction x^{n+1} to profile W_f.
3. Hypothesis H_f is accepted if the similarity rate $\delta(x^{n+1}, W_f)$ satisfies the condition $\delta(x^{n+1}, W_f) \le \varepsilon_f$, where ε_f is some parameter.

The authors describe the algorithm of profile creation and calculation of similarity rate $\delta(x^{n+1}, W_{c_k})$ for hypothesis H_l below. (The scheme for testing of the hypothesis H_f is similar.)

Creation of Cardholder's Profile

Cardholder's profile W_{c_k} is a typical cardholder behavior model, which represents a generalized pattern of the transactions executed earlier by the holder of card c_k. This model is a special structure neural network trained by the SOM algorithm on the basis of the set of transactions $X_{c_k} \in X_n^l$ and is able to recognize typical transactions of a legal cardholder.

In the process of building the self-organizing map, the authors suggest not to use $x^i \in X_{c_k}$ directly, but rather vectors $p^i = (p_1^i, \ldots, p_m^i, \ldots, p_M^i) \in P_{c_k}$, $i = \overline{1;v}$, obtained from the vectors $x^i \in X_{c_k}$ and the parameters of the current state of the card account $c_k = (c_1^k, \ldots, c_s^k)$. To build the set P_{c_k}, the authors apply a function $\varphi : X_{c_k} \to P_{c_k}$ that is a composition of functions $\varphi_0, \varphi_1, \ldots, \varphi_{M-m}$ described later.

The components of the vector $p^i \in P_{c_k}$ can be divided into two groups:

1) The characteristics $p_1^i, \ldots, p_m^i$ of the current transaction $x^i \in X_{c_k}$, which are in fact the values of the appropriate components $x^i \in X_{c_k}$ to which a function φ_0 is applied:

$$p_j^i = \varphi_0(x_j^i) = \begin{cases} x_j^i, \text{if } x_j^i \text{ is a numeric characteristic} \\ I(x_j^i), \text{if } x_j^i \text{ is a symbolic characteristic} \end{cases}, \quad j = \overline{1;m}.$$

The function $I(x_j^i)$ is built using a statistics-based indexing method. Each symbolic value is associated with a numeric index according to its frequency in the training set, which is later used in the training of the neural network as described below.

- The frequency $F(\tau_j^s)$ in the training set X_{c_k} of each unique value $\tau_j^s \in T_j$ of a symbolic parameter x_j^k is calculated as follows:

$$F(\tau_j^s) = \sum_{k=1}^{v} \chi_k(\tau_j^s), \text{ where } \chi_k(\tau_j^s) = \begin{cases} 1, \text{if } x_j^k = \tau_j^s \\ 0, \text{if } x_j^k \neq \tau_j^s \end{cases}, s = \overline{1;s_j}.$$

- The set of unique values $T_j = \{\tau_j^1, \ldots, \tau_j^s, \ldots, \tau_j^{s_j}\}$ is ordered according their decreasing frequency $F(\tau_j^s)$, $s = \overline{1;s_j}$, $F(\tau_j^1) \geq F(\tau_j^2) \geq \ldots \geq F(\tau_j^{s_j})$.

- Each unique symbolic value $\tau_j^s \in T_j$ is associated with a numeric index $I_{\tau_j^s}$:

$$I_{\tau_j^1} = 1; I_{\tau_j^s} = I_{\tau_j^{s-1}} + 1, s = \overline{2; s_j} .$$

- Then, the function $I(x_j^i)$ is defined as:

$$I(x_j^i) = I_{\tau_j^s} \text{ when } x_j^i = \tau_j^s .$$

Such an indexation allows maintaining the initial relative importance of the unique values and the correlation between them.

Examples of characteristics $p_1^i, \ldots, p_m^i$ are transaction amount, transaction time, transaction type, terminal number, terminal city, etc.

2) The characteristics $p_{m+1}^i, p_{m+2}^i, \ldots, p_M^i$ of the transaction history on card c_k, calculated using the functions $\varphi_0, \varphi_1, \ldots, \varphi_{M-m}$ on the basis of the set of transactions X_{c_k}, executed earlier with card c_k up to moment t_n:

$$p_{m+1}^i = \varphi_1(x^1, x^2, \ldots, x^i), \quad p_{m+2}^i = \varphi_2(x^1, x^2, \ldots, x^i), \quad p_M^i = \varphi_{M-m}(x^1, x^2, \ldots, x^i) .$$

Examples of characteristics $p_{m+1}^i, p_{m+2}^i, \ldots, p_M^i$ are: number of transactions carried out during a period of D hours, cumulative amount of transactions during D hours, number of terminals used by the cardholder during D hours, etc.

The resultant set $P_{c_k} = \{p^1 = (p_1^1, \ldots, p_M^1), \ldots, p^v = (p_1^v, \ldots, p_M^v)\}$ is the training set used for creating cardholder's profile W_{c_k}.

As a result of SOM learning [20] with the training set P_{c_k} a matrix of neuron weights of the trained map is obtained, which is actually the cardholder's profile for card c_k: $W_{c_k} = \left\| w_k^s \right\|_{\substack{s=\overline{1;d} \\ k=\overline{1;M}}}$. The weight vectors $w^i = (w_1^i, \ldots, w_M^i), i = \overline{1;d}$ specify the most typical values of the components of vector $p^i = (p_1^i, \ldots, p_M^i)$, which are present in the training set P_{c_k}.

In result, for each transaction $x^i \in X_{c_k}$ there is a certain j-th cell on the SOM such that $\left\| x^i - w^j \right\| = \min_{k=1;2;\ldots;d} \left\| x^i - w^k \right\|$.

Calculation of Transaction Similarity Rate to Profile

Once the neural network learning process is over, every new transaction x^{n+1} on card c_k is checked for similarity to profile W_{c_k}.

The similarity rate $\delta(x^{n+1}, W_{c_k})$ of transaction $x^{n+1} = (x_1^{n+1}, \ldots, x_m^{n+1})$ to profile W_{c_k} can be determined as the deviation of the vector $p^{n+1} = \varphi(x^{n+1})$ from the nearest cell of the map W_{c_k}, or in other words as the minimum of the distances between the vector $p^{n+1} = (p_1^{n+1}, \ldots, p_M^{n+1})$ and the vectors of neurons' weights $w^1, \ldots, w^d$:

$$\delta(x^{n+1}, W_{c_k}) = \min_{i=1;2;\ldots;d} \left\| p^{n+1} - w^i \right\|.$$

The most commonly used type of distance measure is the Euclidean distance:

$$\left\| p^{n+1} - w^i \right\| = \sqrt{\sum_{k=1}^{M} (p_k^{n+1} - w_k^i)^2}.$$

However, in some applications more complex distance measures are required. It depends mainly on specific characteristics of the data space and the expected results:

- *Squared Euclidean distance*: $\left\| p^{n+1} - w^i \right\| = \sum_{k=1}^{M} (p_k^{n+1} - w_k^i)^2$. This distance measure place progressively greater weight on objects that are further apart;
- *Manhattan distance*: $\left\| p^{n+1} - w^i \right\| = \sum_{k=1}^{M} \left| p_k^{n+1} - w_k^i \right|$. In most cases, this distance measure yields results similar to the simple Euclidean distance. However, the effect of single large differences (outliers) is dampened;
- *Chebychev distance*: $\left\| p^{n+1} - w^i \right\| = \max_{k=1,\ldots,M} \left| p_k^{n+1} - w_k^i \right|$. This distance measure may be appropriate in cases when one wants to define two objects as "different" if they are different on any one of the dimensions/ coordinates;
- *Power distance*: $\left\| p^{n+1} - w^i \right\| = \left(\sum_{k=1}^{M} \left| p_k^{n+1} - w_k^i \right|^p \right)^{1/r}$. Sometimes one may want to increase or decrease the progressive weight that is placed on dimensions on which the respective objects are very different;

- *Percent disagreement*: $\left\| p^{n+1} - w^i \right\| = \frac{1}{l} \sum_{k=1}^{M} \psi(p_k^{n+1}; w_k^i)$, where

$$\psi(p_k^{n+1}; w_k^i) = \begin{cases} 1, \text{if } p_k^{n+1} \neq w_k^i \\ 0, \text{if } p_k^{n+1} = w_k^i \end{cases}$$, which is useful for categorical features.

Algorithm

The proposed method for transaction analysis is represented as a block diagram in Figure 1. The process of transaction monitoring consists of three stages: data accumulation, training (building of cardholder's profile) and control of transactions.

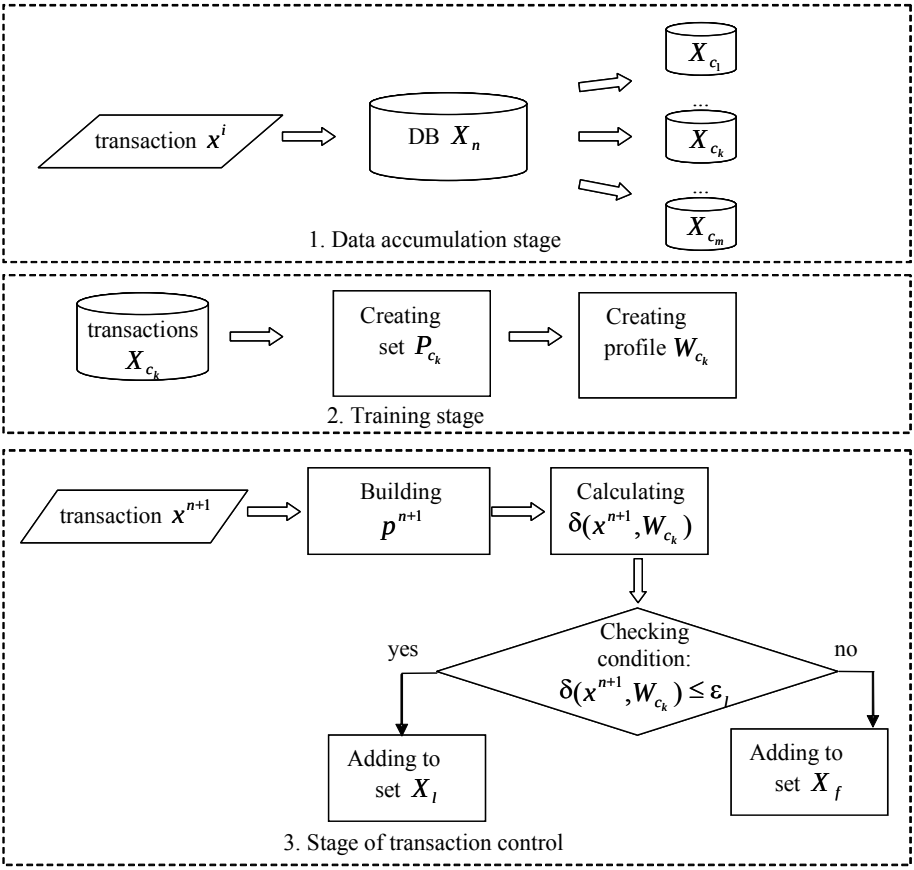


Figure 1: Block Diagram of Transaction Monitoring Algorithm.

At the stage of data accumulation, the data about the transactions on card c_k are collected in the database DB. If the size of X_{c_k} exceeds some predefined level, sufficient to build an adequate profile, then the monitoring process goes to stage two.

At stage two, the training stage, the cardholder's profile W_{c_k} is created as follows:

- The set P_{c_k} is built using the function φ;
- The neural network is trained on the basis of set P_{c_k};
- The profile $W_{c_k} = \left\| w_k^s \right\|_{\substack{s=\overline{1;d} \\ k=\overline{1;M}}}$ is built as a result of training.

After the training stage, the process goes to the stage of transaction control, which consists of the following:

- The vector p^{n+1} is built applying the function φ to every new transaction x^{n+1}: $p^{n+1} = \varphi(x^{n+1})$;
- The deviation of the current transaction x^{n+1} from the profile W_{c_k} (created at the training stage) is calculated: $\delta_0 = \delta(x^{n+1}, W_{c_k})$;
- The value δ_0 is compared with the threshold ε_l fixed for the profile W_{c_k} (ε_l is a boundary value for the degree of similarity of the transactions on card c_k to the profile W_{c_k}. It makes it possible to cut off transactions that deviate from the early established norm and to control the accuracy of fraud detection.);
- If $\delta_0 \le \varepsilon_l$ then transaction x^{n+1} is considered typical/ legal and the vector x^{n+1} is added to the set $X_l = X_{c_k}$;
- If $\delta_0 > \varepsilon_l$ then transaction x^{n+1} is considered suspicious for fraud and is added to the set X_f for further expert analysis.

Example

This section will illustrate the proposed approach to fraud detection. Transactional data is confidential information; therefore, the initial data set was simulated (a list of real transaction parameters and a range of their values were used). The following characteristics (features) were chosen to analyze the transactions: p_1 – transaction amount, p_2 – transaction type, p_3 – terminal identifier, p_4 – city, p_5 – country,

p_6 – number of transactions over the last 48 hours, p_7 – accumulated amount of transactions over the last 48 hours, and p_8 – number of terminals used in the last 48 hours.

A number of credit cards with different characteristics of cardholder's behavior were examined in order to explore the dependence of the constructed model of cardholder's behavior on the transaction similarity degree and to define the required minimum number of transactions in the training set (see Table 1). A small number of transactions (100) in the training set were used intentionally considering the specificity of the Ukrainian credit card market. Most of the cards are characterized by a low number of transactions per month and thus poor transaction history.

Table 1: Characteristics of Credit Cards.

Card #	*Total Amount of Transactions* [21]	*Type of Cardholder Behavior*
Card #1	100+10	All transactions are similar
Card #2	100+10	Most of the transactions are similar, but rare atypical transactions appear
Card #3	90+10	Various transactions

Several models of typical cardholder behavior were built using different number of transactions in the training set. Computational results are given in Table 2.

In the table, ε denotes the average error in the set and $\varepsilon_{\max}$ – the maximum error in the set.

As could be seen from Table 2, the accuracy of detection of fraudulent and legal transactions (test and validation sets) increases with the increase of the number of transactions in the training set. For Card #1, acceptable recognition accuracy (ε=0.0068) has already been reached with 30 transactions in the initial set; for Card #2 and Card #3, with more heterogeneous cardholders behavior, similar recognition accuracy is reached with 60 and 90 transactions, respectively.

Two-dimensional Kohonen maps were built for cardholder behavior model. Figure 2 depicts the distance matrix and the clusters formed for the model of cardholder behavior for Card #3.

The clusters on the map show that cardholder's behavior is characterized by three pronounced types, which were named "Typical ATM transactions," "Typical POS transactions," and "Rare/ anomalous transactions." After processing the anomalous transactions for Card #3, it was observed that their deviation from the model of typi-

cal behavior greatly exceeded the error of recognition of legal transactions (as illustrated in Figure 3). Moreover the more anomalous a transaction is, the greater its deviation from the model. So, this characteristic can be used as degree of suspiciousness of a transaction.

Conclusion

This article has proposed a new approach to transaction monitoring and credit card fraud detection using the Self-Organizing Map algorithm. It enables automated creation of transaction monitoring rules in a learning process and makes possible their continuous improvement in an environment of dynamically changing information in an automated system.

Table 2: Results from Experiments.

Card #	*Set*	*Number of Transactions in the Initial Set*					
		30+10		*60+10*		*90+10*	
		ε	$\varepsilon_{\max}$	ε	$\varepsilon_{\max}$	ε	$\varepsilon_{\max}$
1	Training	7.25E-10	8.71E-9	5.18E-10	9.16E-9	0.0013	0.0263
	Test	0.0068	0.0340	0.0039	0.0395	0.0034	0.0339
	Validation (legal)	0.0068	0.0909	0.0015	0.0395	0.0030	0.0250
	Validation (fraud)	0.9190	1.8136	0.8916	1.7192	0.8917	1.7192
2	Training	0.0007	0.0099	0.0042	0.0351	0.0027	0.0404
	Test	0.0299	0.1495	0.0049	0.0404	0.0381	0.1575
	Validation (legal)	0.1069	0.2257	0.0084	0.1575	0.0456	0.0923
	Validation (fraud)	0.8334	1.7198	0.7840	1.6757	0.7235	1.6757
3	Training	0.0136	0.0754	0.0202	0.0879	0.0224	0.1502
	Test	0.0659	0.3432	0.0508	0.3066	0.0576	0.1371
	Validation (legal)	0.0536	0.3951	0.0321	0.1796	0.0345	0.1252
	Validation (fraud)	0.6930	1.4015	0.6458	1.4852	0.6575	1.5122

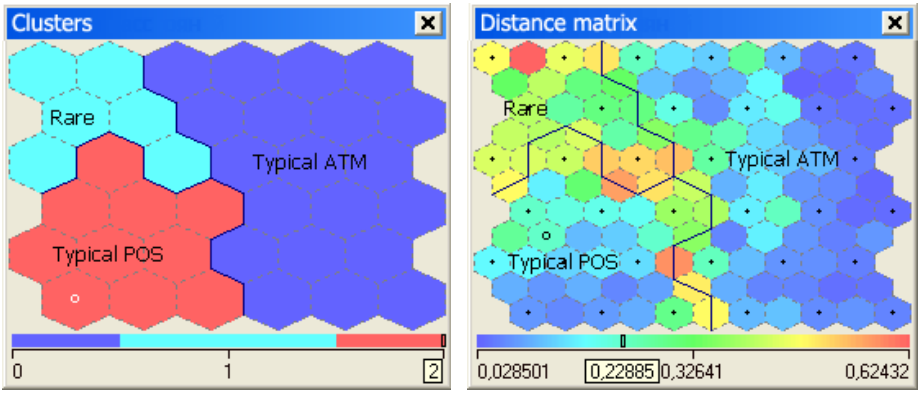
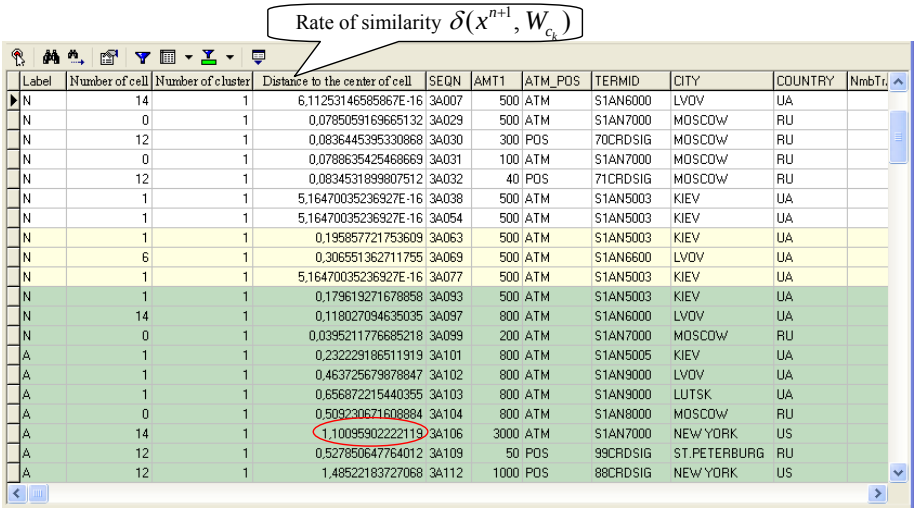


Figure 2: Cardholder's Behavior Model (Card #3).

The advantages of the proposed approach are: the success of the algorithm does not depend on statistical assumptions about data distribution; it deals successfully with noisy data; the method allows modification of the model as new transactions are added and it does not require *a priori* information besides some set of transactions performed by the cardholder; the achieved accuracy of the produced rules is stable (in contrast to the changing concentration and attention of the experts, for example as a



Label	Number of cell	Number of cluster	Distance to the center of cell	SEQN	AMT1	ATM_POS	TERMID	CITY	COUNTRY	NmbTr
N	14	1	6,112531465858676E-16	3A007	500	ATM	S1AN6000	LVOV	UA	
N	0	1	0,0785059169665132	3A029	500	ATM	S1AN7000	MOSCOW	RU	
N	12	1	0,0836445395330868	3A030	300	POS	70CRDSIG	MOSCOW	RU	
N	0	1	0,0788635425468669	3A031	100	ATM	S1AN7000	MOSCOW	RU	
N	12	1	0,0834531899807512	3A032	40	POS	71CRDSIG	MOSCOW	RU	
N	1	1	5,16470035236927E-16	3A038	500	ATM	S1AN5003	KIEV	UA	
N	1	1	5,16470035236927E-16	3A054	500	ATM	S1AN5003	KIEV	UA	
N	1	1	0,195857721753609	3A063	500	ATM	S1AN5003	KIEV	UA	
N	6	1	0,306551362711755	3A069	500	ATM	S1AN6600	LVOV	UA	
N	1	1	5,16470035236927E-16	3A077	500	ATM	S1AN5003	KIEV	UA	
N	1	1	0,179619271678858	3A093	500	ATM	S1AN5003	KIEV	UA	
N	14	1	0,118027094635035	3A097	800	ATM	S1AN6000	LVOV	UA	
N	0	1	0,0395211776685218	3A099	200	ATM	S1AN7000	MOSCOW	RU	
A	1	1	0,232229186511919	3A101	800	ATM	S1AN5005	KIEV	UA	
A	1	1	0,463725679878847	3A102	800	ATM	S1AN9000	LVOV	UA	
A	1	1	0,656872215440355	3A103	800	ATM	S1AN9000	LUTSK	UA	
A	0	1	0,509230671608884	3A104	800	ATM	S1AN8000	MOSCOW	RU	
A	14	1	1,10095902222119	3A106	3000	ATM	S1AN7000	NEW YORK	US	
A	12	1	0,527850647764012	3A109	50	POS	99CRDSIG	ST.PETERBURG	RU	
A	12	1	1,48522183727068	3A112	1000	POS	88CRDSIG	NEW YORK	US	

Figure 3: Anomalous Transactions on Card #3.

result of tiredness); the simple visualization of data (even in the case of a large number of transactions); and the possibility to detect isolated data structures.

The methodology described in this article is an early stage of research aimed to produce a framework for unsupervised fraud detection. The objective is to improve and implement in detail the proposed method for accurate and fast fraud detection. Furthermore, it would be interesting to compare the results obtained with the proposed in this article method with results obtained with other methods for fraud detection.

The application of the proposed method for transaction analysis is not restricted to the problem described in this article. It could also be used to create a profile of typical activity of Point-Of-Sale, profile of "good" potential clients, general profile of "good" and "bad" transactions, etc.

Notes:

[1] *Managing Risk in the 21st Century. Strategies for Issuing & Acceptance* (Visa Int., October, 2000), 156.

[2] *Managing Risk in the 21st Century. Strategies for Issuing & Acceptance*.

[3] A.I. Ginsburg, *Plastic Cards* (Saint Petersburg: Peter, 2004), 128 (in Russian).

[4] M.C. Vertusaev, Ya.Yu. Kondrat'ev, S.E. Pugachev, A.M. Yurchenko, "Crime Methods Utilizing Bank Cards," *Information Technologies for Information Protection* 3, no. 1 (1999): 50-67.

[5] Tej Paul Bhatla, Vikram Prabhu, and Amit Dua, "Understanding Credit Card Frauds," *Cards Business Review* 1 (Tata Consultancy Services, June 2003).

[6] Bhatla, Prabhu, and Dua, "Understanding Credit Card Frauds."

[7] K. Chikin and I. Shlyik, "Countering Illegal Transactions in Internet Purchasing Systems," *World of Cards* 7 (2002): 15-21.

[8] *Managing Risk in the 21st Century. Strategies for Issuing & Acceptance*.

[9] Ginsburg, *Plastic Cards.*

[10] Bhatla, Prabhu, and Dua, "Understanding Credit Card Frauds."

[11] Bhatla, Prabhu, and Dua, "Understanding Credit Card Frauds."

[12] Bhatla, Prabhu, and Dua, "Understanding Credit Card Frauds;" Chikin and Shlyik, "Countering Illegal Transactions."

[13] Rüdiger W. Brause, T. Langsdorf, and M. Hepp, "Credit Card Fraud Detection by Adaptive Neural Data Mining," Internal Report 7/99 (J.W. Goethe-University, Computer Science Department, Frankfurt, Germany, 1999), <http://www.cs.uni.frankfurt.de/fbreports/07.99.ps.gz> (12 Dec. 2005); Bhatla, Prabhu, and Dua, "Understanding Credit Card Frauds."

[14] Richard J. Bolton and David J. Hand, "Unsupervised Profiling Methods for Fraud Detection," Technical Report (Department of Mathematics, Imperial College, London, 2002).

[15] Philip D. Wasserman, *Neural Computing: Theory and Practice* (New York: Van Nostrand Reinhold, 1990), (Translation into Russian, Moscow, Mir, 1992): 192; Brause, Langsdorf, and Hepp, "Credit Card Fraud Detection by Adaptive Neural Data Mining."

[16] Chikin and Shlyik, "Countering Illegal Transactions."

[17] Teuvo Kohonen, "The Self-Organizing Map," *Proceedings of the IEEE* 78, no. 9 (September 1990): 1464–1480.

[18] Wasserman, *Neural Computing: Theory and Practice*.

[19] A. Gorbunov, "Application of the Self-Organizing Map in Business and Finance," Bank Technologies 4 (1999): 34-40 (in Russian); Wasserman, *Neural Computing: Theory and Practice*.

[20] Kohonen, "The Self-Organizing Map."

[21] For every card 100 typical transactions and 10 anomalous ones were generated.

VLADIMIR A. ZASLAVSKY is Head of Department Mathematical Methods for Ecological and Economic Research, Faculty of Cybernetics, National Taras Schevchenko University of Kiev. He was Vice Dean of the Faculty of Cybernetics in the period 2000-2004. Dr. Zaslavsky received his PhD in Mathematics from the Faculty of Cybernetics, in 1984. He has been Associate Professor since 1992 and has more than 100 publications in the areas of system analysis of complex systems, risk analysis, reliability optimization and redundancy, and decision support systems. He is a member of IIASA Society and President of the AFCEA – Ukraine Chapter. *E-mail*: zas@unicyb.kiev.ua.

ANNA A. STRIZHAK obtained a M.S. degree in Systems Analysis and Theory of Decision Making from National Taras Shevchenko University of Kiev in 2004. She is currently pursuing her Ph.D. degree in Systems Analysis from the Faculty of Cybernetics, National Taras Shevchenko University of Kiev, Ukraine. Simultaneously, she works as system analyst at "UkrCard" company (International Payment System "UkrCard") and is actively involved in a research project called "Development of software for transactional risk analysis and evaluation in PS UkrCard." Her current research interests include systems analysis, analysis and evaluation of risks in payment systems based on card technology, application of neural technologies for card fraud detection, development of automated fraud detection and prevention systems. *E-mail*: st-anna@ukr.net