Credit Card Fraud Detection with a Cascade Artificial Neural Network and Imperialist Competitive Algorithm

Morteza Kolali Khormuji
Department of Computer,
Islamic Azad University,
Science and Research Branch-Bushehr,
Bushehr, Iran

Mehrnoosh Bazrafkan
Department of Computer,
Islamic Azad University,
Science and Research Branch-Bushehr,
Bushehr, Iran

Maryam Sharifian
Department of Computer,
Islamic Azad University,
Science and Research Branch-Bushehr,
Bushehr, Iran

Seyed Javad Mirabedini
Department of Computer,
Islamic Azad University,
Central Tehran Branch,
Tehran, Iran

Ali Harounabadi
Department of Computer Engineering,
Tehran Center Branch,
Islamic Azad University,
Tehran, Iran

ABSTRACT

Credit Card Fraud is one of the biggest threats to business establishments today. This paper presents a cascade artificial neural network for the recognition of credit card fraud detection. This system aims at attaining a very high recognition rate and a very high reliability, In other words, excellent recognition performance of credit card fraud detection was obtained. Then, One solution was proposed: utilizing a cascade artificial neural networks for enhancing recognition rate and reducing rejection rate. The gating networks (GNs) are used to congregate the confidence values of three parallel artificial neural networks (ANNs) classifiers. The Imperialist Competitive Algorithm (ICA) is a new evolutionary algorithm which was recently introduced and has a good performance in some optimization problems. The weights of the GNs are trained by the Imperialist Competitive Algorithm (ICA) to achieve the overall optimal performance. The experiments conducted on the database from a large Brazilian bank produced encouraging results: high accuracy of 98.56% with minimal rejection in the last cascade layer.

General Terms:

Pattern Recognition, Neural Network, Security, Evolutionary Algorithm

Keywords:

Credit Card Fraud Detection, Cascade Neural Networks, Imperialist Competitive Algorithm

1. INTRODUCTION

Credit card fraud can be defined as Unauthorized account activity by a person for which the account was not intended. This is an event for which action can be taken to stop the abuse in progress and incorporate risk management practices to protect against similar actions in the future. In simple terms, Credit Card Fraud is defined as when an individual uses another individual credit card for personal reasons while the owner of the card and the card issuer are not aware of the fact that the card is being used. Contrary to popular belief, merchants are far more at risk from credit card fraud than the cardholders.

While consumers may face trouble trying to get a fraudulent charge reversed, merchants lose the cost of the product sold, pay charge back fees, and fear from the risk of having their merchant account closed. Increasingly, the card not present scenario, such as shopping on the internet poses a greater threat as the merchant (the web site) is no longer protected with advantages of physical verification such as signature check, photo identification, etc. In fact, it is almost impossible to perform any of the ”physical world” checks necessary to detect who is at the other end of the transaction.

This makes the internet extremely attractive to fraud perpetrators. According to a recent survey, the rate at which internet fraud occurs is 12 to 15 times higher than ”physical world” fraud. However, recent technical developments are showing some promise to check fraud in the card not present scenario. Frauds can be broadly classified into three categories, i.e., traditional card related frauds, merchant related frauds and internet frauds [2]. The different types of methods for committing credit card frauds are listed below[2]:

> Application Fraud, Lost/ Stolen Cards, Account Takeover, Fake And Counterfeit Cards, Erasing The magnetic strip, Creating a fake card, Altering card details, Skimming, Site cloning, False merchant sites, Credit card generators

Unfortunately, occurrences of credit card frauds have only shown an upward trend so far. The fraudulent activity on a card affects everybody, i.e., the card holder, the merchant, the acquirer as well as the issuer. Analyzing whether each transaction is legitimate or not is very expensive. Confirming whether a transaction was done by a client or a fraudster by phoning all card holders is cost prohibitive if we check them in all transactions.

There are multiple algorithms for credit card fraud detection. They are artificial neural network models which are based upon artificial intelligence and machine learning approach, distributed data mining systems[5], sequence alignment algorithm which is based upon the spending profile of the cardholder[4], Meta learning Agents and Fuzzy based systems[18],Credit Card Fraud Detection with Artificial Immune System[3],anti-k nearest neighbor algorithm[1]. Most of the credit card fraud detection systems mentioned above are based on artificial intelligence, Meta learning and pattern matching. us suggest a credit card fraud detection system using Bayesian and neural network techniques to learn models of fraudulent credit card transactions.

Fraud prevention by automatic fraud detections is where the well-known classification methods can be applied, where pattern recognition systems play a very important role. In recent research paper, the recognizers were designed for pursuing the highest recognition rate and less attention was paid to the recognizer accuracy or error rate. Recognition with an appropriate rejection option provides a means to reduce the error rate through a rejection transaction scheme, i.e., the mechanism can withhold a decision if the confidence level is not high enough and it can direct a rejected transaction pattern to the exceptional process of manual handling.

With a rejection transaction option the system reliability is enhanced. It is very difficult for a single classifier to obtain a very high accuracy for a complex pattern recognition problem, especially for the Credit Card Fraud Detection recognition due to the variability of Fraud Detection styles. There are a few possible solutions to reduce the errors [6]. Although the idea of rejection was initially proposed and analyzed three decades ago [9, 10], in the past few years ,researchers focused mainly on reaching the highest recognition rate without taking rejection transaction into consideration.

In this paper, we will mainly address the proposal of using a cascade neural network with a rejection strategy and Imperialist Competitive Algorithm (ICA). In each layer, a new classifier is used to either accept the input pattern, or reject transaction the input pattern. If the input pattern is rejected, then it is sent to a higher cascade recognition engine for further processing. Theory and experiments will prove that our proposed cascade neural network with rejection transactions option can achieve a very low error rate while pursuing a very high recognition rate.

2. STRUCTURE OF THE CREDIT CARD DATA

The credit card data used in this study are taken from a national banks credit card data warehouses with the required permissions. The past data in the credit card data warehouses are used to form a data mart representing the card usage profiles of the customers. Though some of the customers may have more than one credit card, each card is taken as a unique profile because customers with more than one card generally use each card for a different purpose. Every card profile consists of variables each of which discloses a behavioral characteristic of the card usage. These variables may show the spending habits of the customers with respect to geographical locations, days of the month, hours of the day or merchant category codes (MCC) which show the type of the merchant where the transaction takes place. Later on, these variables are used to build a model to be used in the fraud detection systems to distinguish fraudulent activities which show significant deviations from the card usage profile stored in the data-mart.

The number of transactions for each card differs from one to other; however, each transaction record is of the same fixed length and includes the same fields. Hand and Blunt gave a detailed description of the characteristics of credit card data . These fields range from the date and hour of the transaction to the amount, transaction type, MCC code, address of the merchant where the transaction is done and etc. The date and hour of the transaction record shows when the transaction is made. Transaction type shows whether this transaction is a purchase or a cash-advance transaction. MCC code shows the type of the merchant store where the transaction takes place. These are fixed codes given by the members of the VISA International Service Association. However, many of these codes form natural groups. So, instead of working with hundreds of codes, we grouped them into 25 groups according to their nature and the risk of availability to commit a fraud.

The goods or services bought from merchant stores in some MCC codes can be easily converted to cash. As a result, transactions belonging to these MCC codes are more open to fraud and more risky from the transactions belonging to others. The grouping of the MCC codes are done according to both the number of the fraudulent transactions made belonging to each MCC code and the interviews done with the personnel of the data supplier bank with domain expertise about the subject.

3. APPROACH CREDIT CARD FRAUD DETECTION

In this section we discuss credit card fraud and the specific problem that arise with it. Credit card fraud prevention is the first line of defense in reducing costs associated with credit card fraud. Data mining is a process of extracting patterns from data, and a process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cut costs, or both. Machine learning in general falls into two main categories, supervised learning and unsupervised learning. Fraud detection methods can be categorized into either supervised or unsupervised learning [7].

Supervised machine learning in credit card fraud detection is a technique that applies algorithms on both fraudulent and legitimate instances to construct models that assign new observations into one of the two classes ? the classes being either fraudulent or legitimate. The goal of supervised learning is to build a concise model of the distribution of class labels in terms of predictor features (Witten and Frank 2005). The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown.

3.1 Bayesian Network

Bayesian belief networks [11] are powerful modeling tools for condensing what is known about causes and effects into a compact network of probabilities. A Bayesian network is a graphical model for probabilistic relationships among a set of variables. The Bayesian network has become a popular representation for encoding uncertain expert knowledge in expert systems (Heckerman,Geiger and Chickering 1995). Bayesian networks can readily handle incomplete data sets and can learn about causal relationships. Bayesian belief networks are very effective for modeling situations where information about the past and/or the current situation is vague, incomplete, conflicting, and uncertain, whereas rule-based models result in ineffective or inaccurate predictions when the data is uncertain or unavailable.

In a Bayesian Network graphical model each node represents a random variable, and the directed edges of the graph represent conditional dependence assumptions. Hence they provide a compact representation of joint probability distributions.

3.1.0.1 . The probability of joint events can be defined as:

$$P(E_1, E_2) = P(E_1).P(E_2|E_1) \tag{1}$$

Where $P(E_1)$ is the probability of event 1 being true, $P(E_2|E_1)$ is the marginal probability of event 2 being true given the condition that event 1 is also true, finally $P(E_1, E_2)$ is the probability that both events occur. The Bayesian Network diagram is constructed to show the marginal and joint probabilities of events.

3.2 Artificial Neural Networks

Artificial Neural Networks (ANN) [13] are computational models that try to mimic our body's biological neural networks and can easily adapt to change. This mathematical model consists of interconnected artificial neurons (nodes) that can receive one or more inputs and sums them to produce a prediction (output). A neuron has two modes of operation: training mode, and usage mode. In training mode, the neuron can be taught to associate a certain prediction with an input pattern. While in usage mode, if a taught input pattern is detected by the neuron its associated prediction is outputted.

The effect of each input's contribution to the final prediction is dependent on the weight of the particular input. To determine a neural network that is an accurate predictor, appropriate weights for the connections must be determined. The most widely used method to determine the optimal connection weights is called *backpropagation*. This method was introduced by Rumelhart, Hinton, and Williams (1986) and through their work artificial neural network research gained recognition in machine learning. Backpropagation utilizes a mathematical algorithm called gradient descent which iteratively adjusts a function's parameters to minimize the squared error function of the network's output. If the function has several minima the gradient descent method might not find the best one.

3.2.0.2 . The sigmoid function is used to calculate the output of each network layer and is defined as follows:

$$f(x) = \frac{1}{1 - e^{-x}} \tag{2}$$

3.2.0.3 . The squared error function is defined as follows:

$$E = \frac{1}{2}(y - f(x))^2 \tag{3}$$

Where $f(x)$ is the network's prediction obtained from the output unit and y is the instance's class label.
To find the weights of a neural network , the derivative of the squared error function must be determined. The derivative of the error function with respect to a particular weight is defined as :

$$\frac{dE}{dw_i} = (y - f(x)) f'(x) a_i \tag{4}$$

Where w_i are the weights for the ith input variable, x is the weighted sum of the inputs, and a_i are the inputs to the neural network. This computation is repeated for each training instance,and the changes associated with a particular weight w_i are added up, multiplied by the learning rate (small constant), and subtracted from the $w_i's$ current value. This is repeated until the changes in the weights become very small.

3.3 Three-Layer ANN Classifier

An ANN classifier consists of input layers, hidden layers, and output layers. In terms of classifying tow transaction, we will have tow output layer, one for each of the transaction(transaction is legitimate or transaction is fraud), and the signal from each output layer is the discriminant function $g_k(x)$.

3.3.0.4 . The discriminant function can be expressed as:

$$g_k(x) \equiv z_k = f\left(\sum_{j=1}^{r} w_{kj} f\left(\sum_{i=1}^{d} w_{ji} x_i + w_{j_0}\right) + w_{ko}\right) \tag{5}$$

where x_i is a feature component; w_{ji} is a weight between the input layer and the hidden layer; w_{kj} is a weight between the hidden layer and the output layer; $i = 1, ..., d$, and d is the number of nodes in the input node; $j = 1, ..., r$, and r is the number of nodes in the hidden layer; $k = 0, 1$ which represents the number of nodes in the outputs layer. For example, 2 nodes of outputs represent 2 transaction.

Thus, the discriminant function can be implemented by a three-layer neural network. A more intuitive proof of the universal expressive power of three-layer nets is inspired by Fourier?s theorem. The theorem states that any continuous function $g_k(x)$ can be approximated arbitrarily by a possible infinite sum of the harmonic function, given a sufficient number of hidden units n_H, proper nonlinearities, and weights [15].
We now turn to the crucial problem of setting the weights based on training patterns and the desired output.

3.4 Backpropagation Algorithm

Backpropagation [14] is one of the simplest and most general methods for the supervised training of multilayer neural networks. The training error on a pattern is considered to be the sum of the output units from the squared differences between the desired output t_k given by a teacher and the ANN' output z_k:

$$J(w) \equiv \frac{1}{2}\sum_{k=1}^{c}(tp_k - z_k)^2 = \dot{\frac{1}{2}}\|tp - z\|^2 \tag{6}$$

where tp and z are the target and the network output vectors of length c, and w represents the weights in the network.The backpropagation learning rule is based on gradient descent. The weights are initialized with random values, and then they are changed in a direction that leads to a reduction in the squared error in equation (6):

$$\Delta w = \eta \frac{\partial y}{\partial w} \tag{7}$$

where η is a learning rate. The iterative algorithm updates the weights as follows:

$$w(m+1) = w(m) + \Delta w(m) \tag{8}$$

where m indexes the particular pattern presentation. For a three-layer neural network, consider first the hidden to output weights: w_{ij} ,if we do differentiation:

$$\frac{\partial J}{\partial w_{kj}} = \frac{\partial J}{\partial net_k}\frac{\partial net_k}{\partial w_{kj}} = -\delta_k \frac{\partial net_k}{\partial w_{kj}} \tag{9}$$

Apply equation (6) to equation (9), then δ_k can be simply represented as:

$$\delta_k = -\frac{\partial J}{\partial net_k} = -\frac{\partial J}{\partial z_k}\frac{\partial z_k}{\partial net_k} = (tp - z_k) f'(\mathrm{net}_k) \tag{10}$$

Each output similarly computes its net activation based on the hidden unit signal y_j as

$$\text{net}_k = \sum_{j=1}^{r} y_j w_{kj} + w_{k0} = \sum_{j=0}^{r} y_j w_{kj} \tag{11}$$

where $y_0 = 1$, and the following derivative exists:

$$\frac{\partial net_k}{\partial w_{kj}} = y_j \tag{12}$$

So the weight update or learning rule for the hidden-to-output weights is:

$$\Delta w_{kj} = \eta \delta_k y_j = \eta \left(tp_k - z_k \right) f' \left(net_k \right) y_j \tag{13}$$

In analogy with equation (10), the sensitivity of the hidden unit is defined as [15]:

$$\delta_j \equiv f' \left(net_j \right) \sum_{k=1}^{C} w_{kj} \delta_k \tag{14}$$

The learning rule for the input-to-hidden weight is:

$$\Delta w_{ji} = \eta x_i \delta_j = \eta \sum_{k=1}^{c} w_{kj} \delta_k f' \left(net_j \right) x_i \tag{15}$$

We use equations (8, 13, 15) to update the weights in the three-layer ANNs in order to minimize the squared errors in equation (6).

4. ANALYSIS OF ACCURACY RATE , REJECTION RATE AND CLASSIFICATION METHODS

In this section, we will analyze the tradeoffs in accuracy and rejection rates [8] in a cascade neural network system which consists of several levels of classifiers. The tradeoff analysis is conducted on an ANN classifier, a classifier and a cascade neural network, respectively.

In our proposed classification system, ANNs are the dominant classifiers. We will analyze the relationships among the accuracy and rejection rates of an ANN classifier using Bayesian probability theory [8]. According to the rule of thumb for a multi-class problem, a multilayer perceptron neural network trained with backpropagation has good estimates of Bayesian probabilities interpretation of network outputs as Bayesian probabilities makes it possible to compensate for differences in pattern class probabilities between test and training data, the error analysis of an ANN classifier is based on Bayesian estimation.

In order to pursue the highest reliability and the lowest error rate with rejection strategy, a recognition rule is optimum if for a given recognition rate, it minimizes the error rate (error probability) and puts the uncertain testing candidates into the rejection transaction category. According to references [12, 16, 17], suppose there is the n-class problem and X is a feature vector, if the decision rule has a rejection strategy, we need to build up an additional class (for example, the 0^{th} class) to represent the rejection transaction category, so that
If $(Score\left(d_k|X\right) = 1)$ and $(1 \leq k \leq n)$ then X is classified;
If $(Score\left(d_k|X\right) = 1)$ and $(k == 0)$ then X is rejected. The optimum rule is to reject the pattern if the maximum of the a posteriori probabilities is less than the defined threshold. According to Bayesian probability theory, the optimum rule has the following two conditions:

(1) To accept the pattern X for recognition and to identify it as belonging to the $k - th$ pattern:
$Score\left(d_k|X\right) = 1$ if and only if $p\left(\omega_k\right) F\left(X|\omega_k\right) \geq p\left(\omega_i\right) F\left(X|\omega_i\right)$ and

$$p\left(\omega_k\right) F\left(X|\omega_k\right) \geq (1-t) \sum_{i=1}^{n} p\left(w_i\right) F\left(X|\omega_i\right) \tag{16}$$

(2) To reject the pattern X:
$Score\left(d_0|X\right) = 1$ whenever

$$max_k \left(p\left(\omega_k\right) F\left(x|\omega_k\right) \leq (1-t) \sum_{i=1}^{n} p\left(\omega_i\right) F\left(x|\omega_i\right) \right) \tag{17}$$

where n is the number of classes, $p\left(\omega_i\right)$ $(i = 1, 2, 3, ..., n)$ is a priori probability of observing class ω_i , $P(X|\omega_i)$ is the conditional probability density for X given the ith class, and t is a constant parameter between 0 and 1. The relationships among the accuracy, error, and rejection rates are listed below:
The probability of error, or error rate, is:

$$E\left(t\right) = \int_{v} \sum_{i=1}^{n} \sum_{j=1}^{n} Score\left(d_j|X\right) p\left(\omega_i\right) F\left(X|\omega_i\right) dX \tag{18}$$

4.0.0.5 . The probability of rejection or reject rate is:

$$R\left(t\right) = \int_{v} Score\left(d_0|X\right) \sum_{i=1}^{n} p\left(\omega_i\right) F\left(X|\omega_i\right) dX \tag{19}$$

ReR= (Number of rejected Transaction)/(Total Number of testing Transaction)

4.0.0.6 . The probability of a correct recognition rate is :

$$C\left(t\right) = \int_{v} \sum_{i=1}^{n} Score\left(d_i|X\right) p\left(\omega_i\right) F\left(X|\omega_i\right) dX = 1 - E(t) - R(t) \tag{20}$$

From the above analysis, we know that the error, rejection, and accuracy rates are implicit functions of the threshold parameter t.
The probability of the acceptance or acceptance rate is defined as:

$$A(t) = C(t) + E(t) \tag{21}$$

4.0.0.7 . The accuracy of the transaction recognition system is denoted as:

$$ACC(t) = C(t) + R(t) \tag{22}$$

ACC = (total number of testing transaction - number of misrecognized transaction - number of rejected transaction) / (total number of testing transaction - number of rejected transaction)
In a neural network classifier, the confidence threshold $(Conf)$ can be related to the parameter t .

4.0.0.8 . The relation is denoted as:

$$ANN_{conf} = w^{*}(1-t) \tag{23}$$

where w is an empirical factor, which is selected based on how high the reliability is set. Normally, in our experiments, w is set between 1.25-2.25. Equation (23) demonstrates that an ANN classifier can introduce a rejection strategy by setting a high threshold of confidence value. Based on the above analyses, there are several ways to reduce both the rejection rate and the error rate:

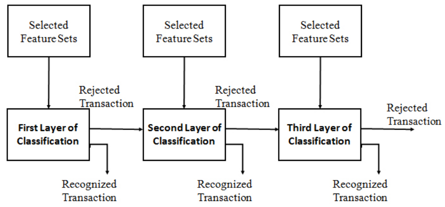

Fig. 1. Cascade neural network structure with rejection strategy

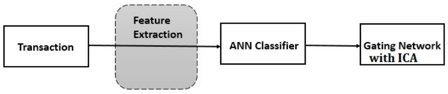

Fig. 2. Schematic diagram of an ANNGNICA classifier.

(1) In order to reduce the error rate, we need to expand the ?rejection region? by setting a smaller parameter t in equations (16) and (17). As a result, more patterns are rejected and fewer patterns are either falsely or correctly accepted. For example, in a neural network classifier, the confidence threshold ANN_{conf} can be set at a high value.

(2) Based on equation (17), in order to reduce the rejection rate, we can increase the value of $max_k(p(\omega_k)F(x|\omega_k)$ and at the same time, we can reduce the value of $\sum_{i=1}^{n} p(\omega_i)F(x|\omega_i))$ In practical applications, when a feature vector (X), which is extracted from a labeled class i, is input into an ANN classifier, the ANN classifier should have the highest conditional probability density $F(x|\omega_i)$ for the labeled class i and the lowest probability density in all other classes. This means that the more discriminative features play an important role in reducing the rejection transaction rate. The recognition scheme is shown in Fig. 1

The cascade neural network consisted of three layers of classification, which were serially linked. As discussed, the combination of classifiers is said to constitute an classifier. In the proposed system, each layer was composed of three levels of the ANNs and gating networks classifiers using Imperialist Competitive Algorithm(ANNGNICA).
Each ANNGNICA classifier consisted of one ANNs and one GNs with Imperialist Competitive Algorithm. Fig.2 shows schematic diagram of an ANNGNICA classifier.

In the training procedure, those classifiers from second to third levels were trained by the rejected transaction of the training set at previous level of the classifier. If the classifier was at the first level of each layer classification, the classifier would be trained by the whole training samples with different feature sets, as previously shown in Fig. 1.

In the testing procedure, most of the transactions in the testing set has to be correctly recognized at the first level of the classifiers in the first layer. More difficult transaction are rejected and sent to the higher level classifiers for recognition by going through the different layers. In other transactions, those classifiers were designed at higher levels and trained to recognize those difficult transactions which were rejected by lower level classifiers.

In order for enough training samples to remain for the second and higher levels, confidence values for the classifiers in the training procedure could be set much higher than those used in the testing procedure. This procedure allowed a much higher percentage of the training samples in the training procedure to be rejected and used as training samples for the next level of classification. Overall, the proposed cascade neural network system had the following advantages:

a) As the cascade recognition scheme with classifier is applied, the transaction recognition system can use a rejection strategy to reject those transactions with relatively low confidence values rather than taking the risk of their misrecognition. The rejected transactions are sent to the higher level of classifiers for recognition.

b) A framework with GNs is proposed for congregating multi-ANN classifiers' outputs. At the same time, GNs can remedy setback of the ANN classifiers. This framework helps to significantly improve recognition rate and accuracy of the cascade recognition system significantly.

c) Three ANNs and three GNs are used to construct a classifier. The output is voted on the three ANNs and three GNs, rather than on one classifier. This mechanism is based on the democratic voting system so as to achieve more reliable performance. The detailed scheme of voting will be given in the experimental results section of this paper.

5. IMPERIALIST COMPETITIVE ALGORITHM

The Imperialist Competitive Algorithm (ICA) is a new evolutionary optimization algorithm which is derived by imperialistic competition. ICA, like other evolutionary algorithms commences with an initial population which is known as the country; a country contains types of: $colonies$ and $imperialists$ which together form empires [19]. Indeed, imperialist countries try to overcome other countries and turning them to their colonies. Also, imperialist countries compete strongly with each other for taking occupancy of other countries; Imperialistic competition among these empires forms the proposed evolutionary algorithm. During this competition the weakest empires collapse and stronger ones will get more potency [20]. Imperialistic competition converges to a state in which there exists only one empire and colonies have the same cost function value as the imperialist.

The pseudo code of the Imperialist Competitive Algorithm is introduced as:

(1) Select some random points on the function and initialize the empires.
(2) Move the colonies toward their relevant imperialist (Assimilation).
(3) Randomly change the position of some colonies (Revolution).
(4) If there is a colony in an empire which has lower cost than the imperialist, exchange the positions of that colony and the imperialist.
(5) Unite the similar empires.
(6) Compute the total cost of all empires.
(7) Pick the weakest colony (colonies) from the weakest empires and give it (them) to one of the empires (imperialistic competition).
(8) Eliminate the powerless empires. If stop conditions are satisfied, stop, if not go to 2.

After dividing all colonies among imperialists and creating the initial empires, these colonies commence moving into their relevant imperialist territory which is based on assimilation policy [20]. Fig.

Table 1. Only ANN classifiers were used as the classifier.

	Information and Overall on single-layer neural network		
Information	Result	Overall	Result
Testing transactions	12000	Recognition rate	94.65%
Misrecognized transactions	642	Reliability rate	94.65%
Rejected transactions	null	Misrecognized transactions	642
Recognized transactions	11358	Accuracy rate	94.65%

Table 2. Three $ANNs$ and three GNs with $imperialist\ competitive\ algorithm$ were used as a cascade neural network system.

	each ANN was connected to a GN so as to verify and correct results of ANN				
Information all layer	Testing	Misrecognized	Rejected	Recognized	Accuracy rate
First layer	11220	108	1826	9281	98.74%
Second layer	1826	32	1029	765	95.98%
Third layer	1029	23	779	227	90.8%
	Information and Overall on single-layer neural network				
Overall Information	Recognition rate	Reliability rate	Misrecognized	Accuracy rate	
Result	91.60%	98.56%	163	98.56%	

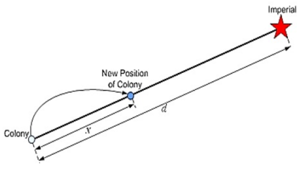

Fig. 3. Movement of colonies toward their relevant imperialist.

3 shows the motion of a colony into its relevant imperialist [19]. In this motion, ? and x are random numbers with uniform distribution as demonstrated in formula (24, 25) and d is the distance between the colony and the imperialist.

$$u \sim \cup(0, \beta \times d) \quad (24)$$

$$\theta \sim \cup(-\gamma, \gamma) \quad (25)$$

where β is a positive number less than 1, d is the space between the imperialist and its colony and orders the derivation from the original direction; In this paper β and γ are considered as 1.00 and 0.25 respectively.
The total power of each empire depends on both the power of the imperialist country and the power of its colonies. This fact is modelled by defining the total cost by:

$$T.C._n = Cost(imperialist_n) + \xi \left\{cost(colonies \cdot of \cdot impire_n)\right\} \quad (26)$$

where $T.C._n$ is the total cost of the nth empire, and xi is a positive number which is considered to be less than 1. A little value for xi causes the total power of the empire to be determined by just the imperialist and increasing it will increase the role of the colonies in determining the total power of an empire. In imperialistic competition, all imperialists try to take the assets of other imperialists and develop their own power.

The problem of the weight selection in the gating network is well suited to the evolution by ICA. Our proposed scheme congregates the ANN classifiers outputs with their confidence values through the weighted linear combination and nonlinear generalization. ICA-based GNs have the following advantages: Given a vector X, with n-dimensional random weights, the task of the ICA is to find a vector of the weights that minimizes the coxt function in Eq. (27).
$G_i = g_{i0}, g_{i1}$ $i = 1, 2, 3$ for three
G_i is the output vector of the GN_i

$$\sum_{i=1}^{3} |Y - G_i|^2 \Longrightarrow 0 \quad (27)$$

In the transaction recognition area, the most difficult problem is to find a reasonable cost function for a large set of training samples. Ideally, the accuracy rate can be used as a cost criterion to train a classifier. However, if an ANN is used as a classifier, it is unfeasible for the accuracy rate to be used as the cost function in ICA evolution because it needs huge computations for each generation of learning.

When Eq. (27) is used as the cost function, the ICA pursue the smallest confidence differences between the GNS outputs G_i and the target confidence vector Y. The inputs of each GN are the corresponding ANN?s outputs, which can be obtained from training samples beforehand.

6. DISCUSSION

The experiments were conducted using cascade neural network and imperialist competitive algorithm (ICA) schemes. The database from a large Brazilian bank, which included 29,104 training samples and 12,000 testing samples, was applied to train and test the proposed system.

In experiment one, only ANNs was used to construct a classifier. In experiment two, three ANNs and three GNs with the ICA were used as a cascade neural network system. In experiment three, was done to test the present classifier system in order to show how a high accuracy rate with different confidence thresholds could be achieved.

Table 3. Accuracy performances of the two cascade neural network system.

Scheme	Recognition Rate (%)	No.of rejected transaction	No.of error	Accuracy rate (%)
Only ANN	94.65%	Not	642	94.65%
Cascade ANN with GN and ICA	91.65%	779	163	98.56%

Table 4. Accuracy thresholds in step 4 and step 9.

Information Step 4	Three GNs were used by setting different confidence thresholds of the three ANNs in step 4			
Parameter thresholds	Recognition rate	Reliability rate	Misrecognized	Accuracy rate
0.92	93.01%	98.13%	212	98.14%
Information Step 4	Three GNs were used by setting different confidence thresholds of the three ANNs in step 9			
Parameter thresholds	Recognition rate	Reliability rate	Misrecognized	Accuracy rate
0.99	63.78%	99.26%	65	99.26%

6.1 Experiment one: Only ANN classifiers were used as the classifier

The first experiment used one layers of the neural network structure. This classifier consisted of only ANNs with GNs.

In this scheme, 642 transactions were $misrecognized$ in total and no transaction was $rejected$. The overall recognition and accuracy rates were 94.65% and 94.65%, respectively. The classifier combination strategy was required to be further explored by conducting more experiments in order to reduce $error\ rate$ and $rejection\ rate$ while increasing $accuracy\ rate$. Detailed information for the scheme is given in Table 1.

6.2 Experiment two: Three ANNs and three GNs with imperialist competitive algorithm were used as a cascade neural network system

In experiment two, used three layers of the cascade neural network structure shown previously in Fig. 1. This scheme congregated three $ANN?s$ recognition results into a $gating\ network$. Each GN was trained by $imperialist\ competitive\ algorithm$ in order to achieve an optimal solution.

Because a gating network was added to the outputs of three $ANNs$, the $gating\ network$ was able to correct some errors that occurred in the $ANNs$ outputs. For example, if some confidence values of one ANN were relatively low, after the $gating\ networks$ correction, the confidence values of the $gating\ network$ were increased. Some rejected transaction of $ANNs$ were recognized by the $gating\ network$. Compared to experiment one, the accuracy rate in experiment two increased from 94.65% to 98.56%.

The experimental results demonstrated that the cascade neural network system with the classifier consisting of three ANNs and three GNs could achieve very high recognition performance. Detailed information for the two schemes is shown in Table 2.
Compared with the classifier without three ANNs and GNs, the classifier with the three ANNs and GNs had a better recognition performance in terms of the overall recognition accuracy and overall error rate. The detailed information for the two schemes is shown in Table 3. The main reason for this improvement is that three $gating\ networks$ are linked to the outputs of three $ANNs$ (each ANN has one gating network). Each GN was trained by $imperialist\ competitive\ algorithm$ in order to achieve an optimal solution. Therefore, each gating network can correct or remedy the errors of the corresponding ANN effectively. As a result, the overall recognition performance is increased.

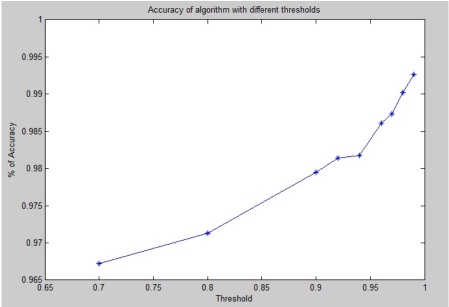

Fig. 4. Accuracy of algorithm with different thresholds

6.3 Experiment three: threshold in cascade neural network system

Three GNs, (each ANN classifier was connected to one gating network) were used by setting different confidence thresholds of the three $ANNs$. In order to investigate the threshold between correct recognition, error and rejection rates and accuracy rate of the three $ANNs$ and three GNs with $imperialist\ competitive\ algorithm$, different confidence thresholds *(Conf)*, which were related to rejecting parameter threshold in the proposed accuracy analysis, were set in this experiment.

The proposed experiment was conducted as follows:
With increasing the threshold, accuracy rate was increased; for example, the following results were obtained by setting the threshold=0.92 in step 4 and also threshold=0.99 in step 9. The results are given in Table 4.

Fig. 4 shows threshold of the percentage change in accuracy, recognition, error and rejection categories by setting different confidence thresholds. In Fig. 4, if confidence threshold was increased from

Table 5. Accuracy rate in confidence threshold was increased from 0.70 to 1.00.

	Accuracy rate of algorithm with different thread		
threshold	Misrecognition Transaction	Recognition rate(%)	Accuracy rate (%)
0.70	393	96.62	96.72
0.80	340	96.00	97.13
0.90	235	93.42	97.95
0.92	212	93.01	98.14
0.94	206	91.86	98.17
0.96	150	87.46	98.61
0.97	133	84.50	98.73
0.98	97	78.38	99.02
0.99	65	63.78	99.26

0.70 to 1.00, accuracy rate would gradually increased to 1; however, the misrecognized transaction number would decreased from 393 to 65. At the same time, some of the rejected transaction at the previous level were correctly recognized. All the threshold parameters associated with the overall system performance are described in Table 5.

Therefore, it could be concluded that the proposed classifier system could achieve much higher recognition performance on credit card transaction when both accuracy and recognition rates were taken into consideration at the same time.

7. CONCLUSION

Credit card fraud has become more hazard in recent years. Handling credit card, risk monitoring system is the key task for the merchant banks to improve merchants risk management level in a scientific, automatic and valuable way of building an accurate, available and easy system. This paper intended to introduce method for credit card fraud detection and the way credit card fraud impacts on the financial institution, merchant and customer. The main contribution of this paper was that a $cascade\ neural\ network$ system with $imperialist\ competitive\ algorithm$ was proposed for increasing transaction recognition system$'s$ and accuracy rate at the same time. The accuracy rate was theoretically analyzed on the system, one fundamental solution for enhance to the increasing its recognition performance: utilizing a cascade artificial neural networks recognition system.

In pursuit of the highest recognition accuracy and the lowest misrecognition rate, we introduce a Imperialist Competitive Algorithm and a good performance in some optimization problems. The design of a cascade neural network system with rejection transaction strategies is also introduced. Based on these strategies, gating networks were used to congregate the confidence values of three parallel ANN classifiers. The weights of the gating networks were trained by $imperialist\ competitive\ algorithm\ (ICA)$ to achieve the overall optimal performance. The framework with gating networks and $imperialist\ competitive\ algorithm\ (ICA)$ could remedy the drawback of the ANN classifiers. It led to the significant improvement of both the accuracy rate and the reliability of the transaction recognition system.

The $cascade\ neural\ network$ by GNs system was able to achieve accuracy rate of 98.56% and 91.60% recognition rate with $imperialist\ competitive\ algorithm$ of the cascade system.

8. REFERENCES

[1] Venkata Ganji, Siva Naga Prasad Mannem, Vol. 4 No. 06 June (2012), Credit card fraud detection using anti-k nearest neighbor algorithm, International Journal on Computer Science and Engineering (IJCSE).

[2] RaghavendraPatidar, Lokesh Sharma, Volume-1 Issue-NCAI2011, June (2011), Credit Card Fraud Detection Using Neural Network, International Journal of Soft Computing and Engineering (IJSCE).

[3] Manoel Fernando Alonso Gadi, Xidi Wang and Alair Pereira do Lago, (2008), Credit Card Fraud Detection with Artificial Immune System, Springer-Verlag Berlin Heidelberg.

[4] Kundu, A. ; Sch. of Inf. Technol., Indian Inst. of Technol., Kharagpur, India ; Panigrahi, S. ; Sural, S. ; Majumdar, A.K., (27 February 2009), BLAST-SSAHA Hybridization for Credit Card Fraud Detection Dependable and Secure Computing, IEEE Transactions on (Volume:6 , Issue: 4)

[5] Bhattacharyya, S, S Jha, K Tharakunnel, and Westland J.C., (2011), Data mining for credit card fraud: A comparative study., Decision Support Systems pp. 602-613.

[6] C.Y. Suen, J. Tan, (2005), Analysis of errors of handwritten digits made by a multitude of classifiers, Pattern Recognition Lett. 26 (1) pp. 369-379.

[7] Alahakoon, L.D. and Halgamuge, S.K., (1998), Knowledge Discovery with Supervised and Unsupervised Self Evolving Neural Networks, Proc. Inter. Conf. Information-Intelligent Systems, pp. 907-910.

[8] C. Frelicot, L. Mascarilla, (2002), Reject strategies driven combination of pattern classifiers, Pattern Anal. Appl. 5 (2) pp 234-243.

[9] C.K. Chow,(1970), On optimum recognition error and reject tradeoff, IEEE Trans. Inf. Theory 16 (1) pp 40-46.

[10] Robert N. Ascher, George M. Koppelman, Martha J. Miller, G. Nagy, Glenmore L. Shelton Jr.,(1971), An interactive system for reading unformatted printed text, IEEE Trans. Computer. C-20 (12) pp 1527-1543.

[11] Suvasini Panigrahi, Amlan Kundu, Shamik Sural, A.K. Majumdar,October (2009) Credit card fraud detection: A fusion approach using Dempster-Shafer theory and Bayesian learning Special Issue on Information Fusion in Computer Security science direct Pages 354-363.

[12] O.D. Trier, A.K. Jain, T. Taxt,(1996), Feature extraction methods for character recognition a survey, Pattern Recognition 29 (4) pp 641-662.

[13] White H. (1989), Learning in Artificial Neural Networks: A Statistical Perspective, Neural Computat., Vol. 1, No. 4, pp. 425-469.

[14] Asari, V.K., (2001), Training of a Feedforward Multiple-Valued Neural Network by Error Backpropagation with a Multilevel Threshold Function, IEEE Trans. on Neural Networks, Vol. 12, No. 6, pp.1519-1520, Nov.

[15] R. O. Duda, P. E. Hart, and D. G. Stork, (2000), Pattern Classification, John Wiley and Sons, Inc., Wiley-Interscience, Second Edition.

[16] II-Seok Oh, J. S. Lee, and C. Y. Suen, (1999), Analysis of Class Separation and Combination of Class-Dependent Features for Handwriting Recognition, IEEE Transaction on PAMI, Vol. 21, No. 10, Oct. , pp. 1089-1094.

[17] M. Bressan and J. Vitria, October (2003), On the Selection and Classification of Independent Features, IEEE Transactions on PAMI, Vol. 25, No. 10, pp. 1312-1317.

[18] Peter J. Bentley, Jungwon Kim, Gil-Ho Jung and Jong-Uk Choi (2000). Fuzzy Darwinian Detection of Credit Card Fraud. In the 14th Annual Fall Symposium of the Korean Information Processing Society. 13th-14th October.

[19] E. Atashpaz Gargari, C. Lucas, (2007), Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition , in: IEEE Congress on Evolutionary Computation, Singapore, pp 4661 - 4667.

[20] A.M. Jasour, E. Atashpaz, C. Lucas, (2008), Vehicle fuzzy controller design using imperialist competitive algorithm, in: Second First Iranian Joint Congress on Fuzzy and Intelligent Systems, Tehran, Iran.