Improving Credit Card Fraud Detection using a Meta-Classification Strategy

Joseph Pun, Yuri Lawryshyn
Department of Applied Chemistry and Engineering,
University of Toronto – Toronto

ABSTRACT

One of the issues facing credit card fraud detection systems is that a significant percentage of transactions labeled as fraudulent are in fact legitimate. These "false alarms" delay the detection of fraudulent transactions and can cause unnecessary concerns for customers. In this study, over 1 million unique credit card transactions from 11 months of data from a large Canadian bank were analyzed. A meta-classifier model was applied to the transactions after being analyzed by the Bank's existing neural network based fraud detection algorithm. This meta-classifier model consists of 3 base classifiers constructed using the decision tree, naïve Bayesian, and k-nearest neighbour algorithms. The naïve Bayesian algorithm was also used as the meta-level algorithm to combine the base classifier predictions to produce the final classifier. Results from the research show that when a meta-classifier was deployed in series with the Bank's existing fraud detection algorithm improvements of up to 28% to their existing system can be achieved.

General Terms

Data Mining, Neural Networks, k-Nearest Neighbor, Naïve Bayesian, Decision Tree

Keywords

Meta-classification

1. INTRODUCTION

Credit card fraud continues to be a significant cost for financial institutions (FIs) and the enhancement of fraud detection can provide significant savings for the FIs. Many different data mining techniques have been applied to the field of fraud detection in the past, however neural network (NN) based algorithms are currently most prevalent in the industry, as well as in academic literature. Neural networks are made up of interconnected nodes that try to imitate the functioning of the human brain. Each node has a weighted connection to several other nodes in adjacent layers. Individual nodes take the input received from connected nodes and use the weights, together with a simple function, to compute output values. Currently, large Canadian banks rely heavily on NN scores determined by a neural network based algorithm to detect fraudulent transactions. This NN score ranges from 1 to 999, where 1 represents the lowest and 999 represents the highest chance of a fraudulent transaction occurrence. Analysis of credit card transaction data from a large Canadian bank showed that transactions with NN scores from 990 to 999 had four times more fraudulent instances than transactions with NN scores from 900 to 910. This suggests that the NN scoring metric is able to identify transactions that are more likely to be fraudulent. However, the data also showed that the majority of transactions with NN scores greater than or equal to 990 are actually legitimate and, on average, only 20% of transactions with NN scores greater than or equal to 990 are fraudulent. Since banks rely heavily on these NN scores to determine fraudulent activity, fraud analysts spend a large portion of their time investigating legitimate accounts, leading to an inefficient use of their time, with the undesirable consequence of unnecessary customer concerns and an increased potential of delay in investigating fraudulent accounts.

This research applies a meta-classifier to 11 months of credit card transactions data that have NN scores greater than or equal to 900 from a large Canadian bank. The goal of this work is to test whether applying a meta-classifier (a multiple algorithm learning technique) to a post-neural network can improve upon the fraud detection system currently in place. Furthermore, the meta-learning aims to filter the legitimate transactions from the fraudulent ones, and by quickly and accurately identifying the fraudulent transactions, fraud losses can be reduced. The goal of our model is to separate the legitimate transactions from the fraudulent ones in this post-neural network dataset. One of the key contributions of this paper is the application of a meta-learning strategy, in a post neural network implementation, on a real historical credit card database consisting of 11 months of all the transactions recorded by a large Canadian bank.

An open-sourced data mining program called Weka was used for this thesis. This software is well established and is widely used in literature, and can be easily adopted by the bank as a separate module complimenting their existing fraud detection system (a neural network based system). Given the significant concerns the Bank had with security issues regarding the data set, a very limited amount of time was available to work with the data, even after all visa client information was masked and the data set was only made available on a secure computer within the Bank's secure IT area. Our strategy was to have all of the data mining technology in place before access to the data was given to us. Unfortunately, perhaps in hind sight not unexpectedly, a significant amount of time was spent cleaning the data. However, because we had our strategy well prepared in advance, we were able to perform a successful analysis within the allotted time frame. We chose to apply a meta-learning strategy for two reasons: 1) the data was already processed through a NN system and the NN score was utilized in our methodology to further enhance detection therefore further NN analysis would be redundant, and 2) previous literature (see [1], [2]) on credit card fraud detection[1] indicated positive results with meta-learning, and in particular, the application of the Naïve Bayesian as the meta-algorithm.

The remainder of the paper is organized as follows. Section 2 presents credit card fraud techniques reported in literature.

[1] We note that none of the previous studies applied meta-learning after NN scoring.

Section 3 describes the meta-learning strategy, and the evaluation and ranking methodologies used in this work. Section 4 presents the discussion and results. Section 5 provides the conclusions of this research.

2. RELATED WORK

Although data mining techniques are used frequently in literature for prediction purposes, few studies have focused on using data mining for credit card fraud detection, likely due to the difficulty in obtaining a real valid dataset. Among the reported studies for credit card fraud detection, the most prominent technique is the neural network algorithm[3]. Studies have shown that this algorithm is able to achieve a reduction of 20% to 40% in total credit card fraud losses, is able to detect credit card fraud in real time, is easy to implement with commercial databases, and is able to quickly and accurately classify transactions [4]. Other methods that have been used in the literature to detect credit card fraud include: Bayesian Belief Networks [5], rule-based systems [6], decision trees[7], support vector machines [8], logistic regression[3], random forests[3], Hidden Markov Models[9], and other single algorithm data mining methods.

Rather than using single algorithm techniques, a second group of research studies focused on applying multiple algorithm techniques in credit card fraud detection. The most quoted research is the meta-learning technique proposed by Chan and Stolfo[10]. In their research they utilized naïve Bayesian, C4.5, CART, and RIPPER as base classifiers and combined them by implementing a stacking method. It was found that their multi-classifier meta-learning approach can significantly reduce the loss amount due to fraudulent transactions by using a 50:50 fraudulent to legitimate distribution in the datasets for training. Brause et al [11] combined a rule-based technique with a neural network to identify fraudulent credit card transactions. It was found that this combined technique increases the probability for the diagnosis of fraud to be correct and therefore is able to decrease the number of false alarms while increasing the confidence level. Phua et al [12] proposed the use of backpropogation neural networks, naïve Bayesian, and C4.5 algorithms as base classifiers, and to combine the base classifiers' predictions using a meta-classifier technique to detect fraudulent automobile insurance claims. Duman and Ozcelik[13] used a novel combination of the genetic algorithm and the scatter search algorithm to detect credit card fraud in a large Turkish bank. By combining these two algorithms Duman and Ozcelik were able to improve the bank's existing fraud detection strategy by 200%.

The abovementioned studies show that the neural network technique is still the most widely used method in fraud detection and that multiple algorithm techniques often improve upon single algorithm techniques. However, none of these studies have looked into applying a multiple algorithm technique to a post-neural network dataset. Furthermore, previous credit card fraud detection methods lack integration with existing commercial fraud detection systems (integration with a bank's neural network system). Our research proposes to apply a meta-classifier to an updated dataset that consists of real-world neural network classified credit card transactions. The meta-classifier is constructed from readily available and well proven algorithms.

3. METHODOLOGY

The methodology applied in this paper closely follows the "meta-learning" techniques introduced by Chan and Stolfo[1]. The meta-learning technique aims to coalesce the results of multiple learners to improve prediction accuracy and to utilize the strengths of one method to complement the weaknesses of another. There are two methods of combing algorithms that were introduced by Chan and Stolfo, the arbiter and the combiner strategies. Through experimentation conducted in previous papers, Chan and Stolfo found that the combiner strategy performs more effectively than the arbiter strategy. Therefore, the combiner strategy is used in this research. In the combiner strategy the attributes and correct classifications of credit card transaction instances are used to train multiple base classifiers. The predictions of the base classifiers are used as new attributes for the meta-level classifier. By combining the original attributes, the base classifier predictions, and the correct classification for each instance, a new "combined" dataset is created which is used as the training data to generate the meta-level classifier. The predictions from the meta-level classifier are then used as the final predictions in the combiner strategy.

3.1 The Data Set

The dataset that was initially received from the Bank contained 11 months of data from December 2008 to October 2009 with one data file per month. For each month, the datasets contained, on average, 100,000 transactions that had NN scores greater than or equal to 900, and of these 100,000 transactions approximately 10% of them were fraudulent. The datasets initially contained 41 attributes, however, after pre-processing and data cleansing only 29 attributes were chosen. A few modifications to the attributes were made, the "Time and "Date" attributes were converted to a more useful attribute by computing the *difference* in time and days between subsequent credit card transactions. The province / state attribute was converted to regions to reduce the number of unique instances. Specifically, the 50 US states were converted and reduced to 4 regions, namely, NEUS (North Eastern United States), MWUS (Mid-Western United States), WUS (Western United States), and SUS (Southern United States), while the 10 Canadian provinces and 3 territories were left unchanged. This was done because the majority of the transactions were within Canada. All transactions outside of Canada and the US, which represented a very small fraction of total transactions, were labeled as "Others". As discussed previously, we emphasize that given our time constraint, there was limited time available for algorithm optimization, and seven commonly used and well documented algorithms were selected to be studied in our work. Furthermore, due to the time restrictions, our evaluations only looked at comparing the caught (TPs) and missed (FNs) transactions between the FI method and the meta-classifier method (these methods are discussed in Section 3.4).

3.2 Selecting Base Classifiers Using a Diversity Metric

The number of base classifiers used for the training stage and the type of algorithms used for each classifier were chosen based on a *diversity* metric as described by Chan [14]. Studies have shown that the accuracy of a prediction model is increased in meta-learning when the diversity of the base classifiers is increased [14]. This entropy-based metric measures the "randomness" of the predictions and how "different" the base classifiers are, based on their predictions. It measures the average amount of information required to represent each event. The larger the diversity value, the more evenly distributed the predictions are for the base classifiers, while a smaller diversity value represents base classifiers that have predictions that have more bias (some predictions are more likely to occur) [14]. As will be shown in the results section, the optimal number of base classifiers to use based on diversity calculations was two. However, since the calculated diversity values for two classifiers versus three classifiers were very similar, the combination of three algorithms was chosen. The best algorithms to use were found to be the naïve Bayesian, decision tree C4.5, and k-nearest neighbor algorithms.

3.3 Meta-Learning Stages

There are four main stages in the meta-learning process. Stage 1 establishes the base classifiers using a training dataset that consists of 50% fraudulent transactions and 50% legitimate transactions (Figure 2). This was done on a month by month basis for the first 8 months (i.e. December, 2008 to July, 2009) where all of the fraudulent transactions for the given month were matched with an equal number of randomly chosen legitimate transactions (Figure 1). In Stage 2, the base classifiers are applied to a validation dataset to generate base predictions. The validation set consisted of all of the transactions of August, 2009 and September, 2009. The predictions from the second stage are then combined with the validation dataset in Stage 3 and a meta-algorithm is applied to this combined dataset (for the months August, 2009 and September, 2009) to produce a meta-classifier (Figure 3) (the naïve Bayesian algorithm has been shown in literature to give the best results as a meta-algorithm). Finally, in Stage 4, the forward predicting test stage, the meta- classifier is applied to the testing dataset (October, 2009) to produce forward looking predictions (Figure 4). These predictions are compared to the NN system predictions, alone, to see if the meta-classifier can improve on fraud detection.

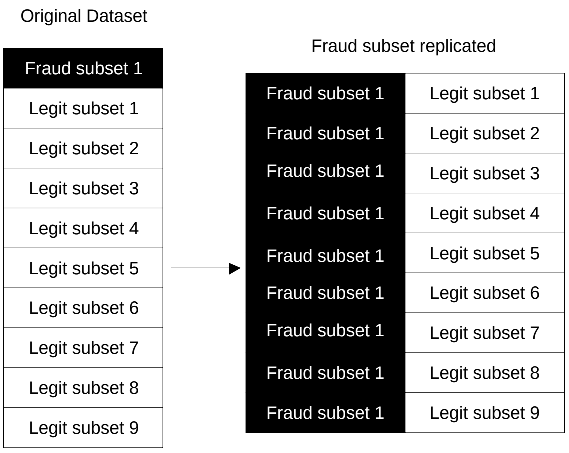
Original Dataset
Fraud subset 1
Legit subset 1
Legit subset 2
Legit subset 3
Legit subset 4
Legit subset 5
Legit subset 6
Legit subset 7
Legit subset 8
Legit subset 9
Fraud subset replicated
Fraud subset 1 | Legit subset 1
Fraud subset 1 | Legit subset 2
Fraud subset 1 | Legit subset 3
Fraud subset 1 | Legit subset 4
Fraud subset 1 | Legit subset 5
Fraud subset 1 | Legit subset 6
Fraud subset 1 | Legit subset 7
Fraud subset 1 | Legit subset 8
Fraud subset 1 | Legit subset 9

Figure 1: Constructing a 50:50 distribution for the training datasets

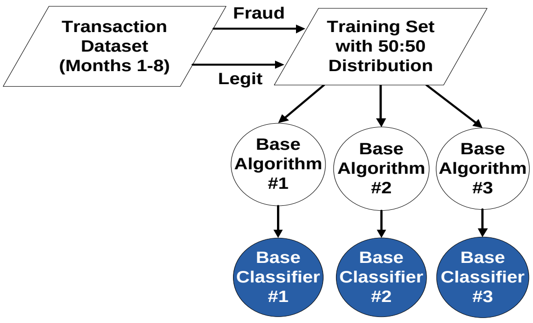

Figure 2: Stage 1 – Training stage in the Meta-learning process

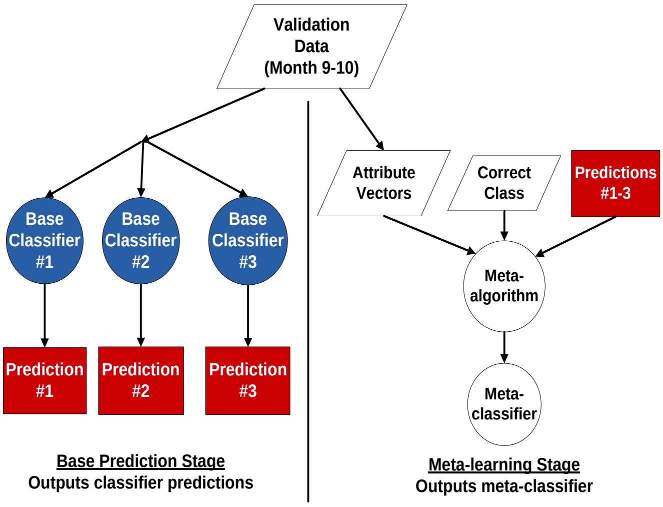

Figure 3: Stage 2 & 3 – Generating the base classifier predictions and constructing the meta-classifier

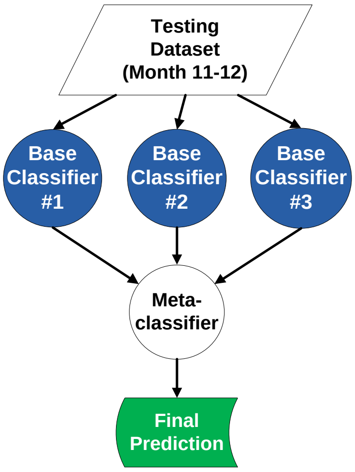

Figure 4: Stage 4 – Generating the final predictions for the dataset

3.4 Ranking and Evaluation

The purpose of ranking is to give priority to transactions that have the highest risk of being fraudulent. Different ranking methods were used for the FI (financial institution) and MC (meta-classifier) evaluations. The five ranking methods considered in this study were:

1. FI: NN Score
2. FI: Transaction Amount
3. MC: NN Score – P > 0.5
4. MC: Transaction Amount – P > 0.5
5. MC: Probability

It was assumed that the FI investigates transactions in one of two ways: either by investigating transactions with the highest NN scores first (FI: NN score), or by investigating transactions with high NN scores (greater than or equal to 900) that have the highest transaction amounts first (FI: Transaction Amount). In the meta-classifier method, the meta-classifier assigns a fraudulent or legitimate classification to each transaction based on a probability score. If the calculated probability was greater than or equal to 0.5, the transaction was considered fraudulent and was flagged accordingly, while if the probability was less than 0.5 the transaction was considered legitimate. Three ranking approaches were considered for the meta-classifier method. The first approach was to rank transactions by highest NN scores with meta-classifier probabilities of 0.5 or greater (MC: NN score – P > 0.5). The second approach was to rank transactions by transaction amount with high NN scores (greater than or equal to 900) by the highest transaction amounts and investigate transactions that had meta-classifier probabilities of 0.5 or greater (MC: Transaction Amount – P > 0.5). The third approach was to rank transactions by highest meta-classifier probabilities and then by NN scores and investigate transactions that were highest on this list (MC: Probability).

Clearly, there is potential in the meta-classifier method to catch fraudulent accounts earlier than the FI method, or vice versa. For example, say the FI method successfully identifies a fraudulent account after 5 transactions while the meta-classifier method is able to identify the same fraudulent account after only 2 transactions. To quantify this difference in performance an evaluation method was applied to determine whether the meta-classifier could catch fraudulent accounts earlier than the FI method. This evaluation analyzed the number of “caught” fraudulent accounts (TPs) and the number of “missed” fraudulent accounts (FN transactions and non-investigated fraud transactions) on a per day basis. The FI method was evaluated using the ‘FI: NN score’ and the ‘FI: Transaction Amount’ rankings, while the meta-classifier method was evaluated using the ‘MC: NN score – P > 0.5’, the ‘MC: Transaction Amount – P > 0.5’ and the ‘MC: Probability’ rankings. Three comparisons were conducted to determine the savings improvements the meta-classifier can provide, the comparisons were as follows:

1. ‘FI: NN Score’ versus ‘MC: NN Score – P > 0.5’
2. ‘FI: NN Score’ versus ‘MC: Probability’
3. ‘FI: Transaction Amount’ versus ‘MC: Transactions Amount’

4. RESULTS

The best base algorithms to train the meta-classifier were selected based on diversity values. The optimal dataset sizes for training, validating, and testing were selected based on the largest Receiver Operative Characteristic (ROC) areas [15]. The Savings Improvement Evaluation was applied to both the meta-classifier and the FI rankings to determine the performance improvement a meta-classifier system can provide to the existing FI system in fraud detection.

4.1 Base Algorithm Selection

Diversity values were calculated for 10 different combinations of algorithms consisting of up to 7 different algorithms. Table 1 shows the number of classifiers tested and the diversity values for different combinations of classifiers.

Table 1: Diversity values for different classifiercombinations

# of Classifiers	**Classifiers**	**Diversity Value**
2	k-nearest neighbor (kNN) &Naïve Bayesian (NB)	0.368051
2	Decision Tree (DT) & NB	**0.400208**
2	DT &kNN	0.091721
3	DT, kNN, NB	**0.394858**
3	DT, kNN, Bayesian Belief Network (BBN)	0.281256
4	DT, NB, kNN&Support Vector Machines (SVM)	0.389205
4	DT, NB, kNN&Neural network (NN)	0.370881
5	DT, NB, kNN, SVM & NN	0.33016
6	DT, NB, kNN, SVM, NN &Logistic Regression	0.308171
7	DT, NB, kNN, SVM, NN,Logistic Regression, & BBN	0.348375

The combinations with the highest diversity values were Decision Tree with Naïve Bayesian, and Decision Tree with Naïve Bayesian and k-Nearest Neighbour. The combination with more classifiers was chosen because each learning algorithm covers a region of tasks favoured by its bias [16], therefore by choosing 3 classifiers, more of the region under study can be covered. Therefore, the Decision Tree algorithm [17], Naïve Bayesian algorithm [18], and the kNN algorithm [19] were chosen as the three base classifiers.

4.2 Selecting Dataset Sizes

The construction of the meta-classifier requires the use of a training dataset, validation dataset, and a testing dataset. Figure 5 shows the ROC areas for meta-classifier models in which the validation dataset size was held at 2 months, the testing dataset size was held at 1 month, and the training dataset size was varied from 5 months to 7 months. Figure 6 shows the ROC areas in which the validation dataset size was varied and the other dataset sizes were held constant. Lastly, Figure 7 shows the ROC areas in which the testing dataset size was varied and the other dataset sizes were held constant.

Results from this study show that the model with the highest prediction accuracy, the largest ROC area, is the model where 8 months of data were used for training, 2 months for validating, and 1 month for testing. However, the ROC areas for the three scenarios resulted in very similar values which suggest that the meta-classifier is a robust model that is able to utilize varying dataset sizes for training, validating, and testing.

Training Dataset Size	Validation Size	Testing Size	ROC Area
Training - 5 months	Validation – 2 months	Testing – 1 month	0.836
Training - 6 months	Validation – 2 months	Testing – 1 month	0.836
Training - 7 months	Validation – 2 months	Testing – 1 month	0.838
Training - 8 months	Validation – 2 months	Testing – 1 month	0.844

Figure 5: ROC areas for meta-classifier models with varying training dataset sizes

Training Dataset Size	Validation Size	Testing Size	ROC Area
Training – 7 months	Validation – 1 month	Testing – 1 month	0.838
Training – 7 months	Validation-2 months	Testing – 1 month	0.841
Training – 7 months	Validation - 3 months	Testing – 1 month	0.841

Figure 6: ROC areas for meta-classifier models with varying validation dataset sizes

Training Dataset Size	Validation Size	Testing Size	ROC Area
Training – 5 months	Validation – 2 months	Testing – 1 month	0.836
Training – 5 months	Validation – 2 months	Testing – 2 months	0.828
Training – 5 months	Validation – 2 months	Testing – 3 months	0.819

Figure 7: ROC areas for meta-classifier models with varying testing dataset sizes

4.3 Savings Improvement Evaluation Results

The number of caught and missed fraudulent accounts for the testing month was counted for both the FI ranking methods and the meta-classifier ranking methods. Table 2 presents the percentage of the total number of fraudulent accounts that are caught by the ranking methods.

The results from Table 2 show that the meta-classifier helps improve the number of caught fraudulent accounts. The largest percentages are seen when transactions are ranked by utilizing both the NN scores and the meta-classifier probabilities (MC Probability). Although Table 2 indicates that the ‘MC: NN Score – P > 0.5’ and the ‘MC: Probability’ ranking methods are comparable in percentage of caught fraudulent accounts, our data show that the ‘MC: Probability’ ranking method was able to flag fraudulent transactions earlier for investigation. Furthermore, results indicate that the ‘MC: NN Score’ and the ‘MC: Probability’ ranking methods are able to improve upon the ‘FI: NN Score’ ranking method by 19% and 28% respectively.

5. RESULTS

The evaluation results show that the meta-classifier can provide quantifiable savings improvements to the assumed FI ranking methods. The Savings Improvement Evaluation successfully showed that when the meta-classifier probability score is combined with a NN scoring metric, a larger percentage of fraudulent accounts are caught. Furthermore, the meta-classifier is able to catch more fraudulent accounts at an earlier time. By ranking transactions using both NN scores and meta-classifier probabilities (MC Probability) to determine the transactions with greatest fraud risk, a 28% improvement to the FI’s NN score ranking method can be achieved. Based on a simple analysis of the average cost of an undetected fraudulent transaction, this improvement was roughly estimated to lead to a savings of approximately $3.5 million per year for the Bank.

Table 2:Percentage of fraudulent accounts caught using each ranking method

	Percent of Fraudulent Accounts Caught				
# of Accounts Investigated in a day	**FI: NN Score**	**FI: Transaction Amount**	**MC: NN Score – P > 0.5**	**MC: Transaction Amount – P > 0.5**	**MC: Probability**
200	36%	5%	43%	22%	41%
500	57%	19%	67%	60%	70%
800	75%	44%	90%	86%	91%

6. REFERENCES

[1] P. K. Chan and S. J. Stolfo, "Experiments in Multistrategy Learning by Meta-Learning," *Proceedings of the second international conference on Information and knowledge management*, pp. 314-323, 1993.

[2] A. L. Prodromidis and S. J. Stolfo, "A Comparative Evaluation of Meta-Learning Strategies over Large and Distributed Data Sets," in *Proceedings of the ICML-99 Workshop on Recent Advances in Meta-learning and Future Work*, Ljubljana, 1999.

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

[4] S. Ghosh and D. L. Reilly, "Credit card fraud detection with a neural network," in *Proceedings of the 27th Hawaii International Conference on System Sciences*, Los Alamitos, CA, 1994.

[5] S. Maes, K. Tuyls, B. Vanschoenwinkel and B. Manderick, "Credit Card Fraud Detection Using Bayesian and Neural Networks," in *Proceedings of the 1st International NAISO Congress on Neuro Fuzzy Technologies*, Havana, Cuba, 2002.

[6] C. Chiu and C. Tsai, "A Web Services-Based Collaborative Scheme for Credit Card Fraud Detection," in *Proceedings of 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service*, 2004.

[7] D. Patil, S. Karad, V. Wadhai, J. Gokhale and P. Halgaonkar, "Efficient Scalable Multi-Level Classification Scheme for Credit Card Fraud Detection," *Internation Journal of Computer Science and Network Security*, vol. 10, no. 8, pp. 123-130, 2010.

[8] Q. Lu and C. Ju, "Research on Credit Card Fraud Detection Model Based on Class Weighted Support Vector Machine," *Journal of Convergence Information Technology*, p. 62, 2011.

[9] A. Srivastava, A. Kundu, S. Sural and A. K. Majumdar, "Credit Card Fraud Detection Using Hidden Markov Model," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 1, pp. 37-48, 2008.

[10] P. L. Chan and S. J. Stolfo, "Toward Scalable Learning with Non-uniform Class and Cost Distributions: A Case Study in Creidt Card Fraud Detection," *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 164-168, 1998.

[11] R. Brause, T. Langsdorf and M. Hepp, "Neural Data Mining for Credit Card Fraud Detection," in *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, Silver Spring, 1999.

[12] C. Phua, D. Alahakoon and V. Lee, "Minority Report in Fraud Detection: Classification of Skewed Data," *SIGKDD Explorations*, pp. 50-59, 2004.

[13] E. Duman and H. Ozcelik, "Detecting Credit Card Fraud by Genetic Algorithm and Scatter Search," *Expert Systems with Applications*, pp. 13057-13063, 2011.

[14] P. K. Chan, *An Extensible Meta-Learning Approach for Scalable and Accurate Inductive Learning*, 1996.

[15] A. Bradley, "The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145-1159, 1997.

[16] R. Vilalta and Y. Drissi, "A Perspective View and Survey of Meta-Learning," *Artificial Intelligence Review*, vol. 18, no. 2, pp. 77-95, 2002.

[17] J. R. Quinlan, C4.5: Programs for machine learning, San Francisco: Morgan Kaufmann, 1993.

[18] G. H. John and P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, SanMateo, 1995.

[19] D. W. Aha, D. Kibler and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, pp. 37-66, 1991.