imbalanced data by using D-optimal design, DOE and RSM to develop an effective classification model. This article considered the Tong et al. (2011)'s proposed method as a cost-sensitive problem [6].

The proposed procedure contains the following three steps:

4.2.1. Design an experiment

An experiment is designed to obtain an appropriate resampling strategy for the majority and minority class in a two-class imbalanced data, while the number of instances drawn from the majority class and the number of instances duplicated from the minority class are designed using undersampling and oversampling, respectively. The experiment considers two factors. Factor A and factor B represent a/b and d/b, respectively, where a denotes the total number of the re-sampling instances in the majority class; b denotes the total number of instances in the minority class of the training data, and d denotes the number of instances duplicated in the minority class. Both factors are continuous, ranging from 0 to r, where r represents N_L / N_S, $r \geq 1$; N_L represents the total number of instances in the majority class, and N_S is the total number of instances in the minority class.

This work adopts the D-optimal design with a quadratic model. The D-optimal design is generated using the Design-Expert 8.0.7.1 computer software, in which a 20-run design is generated, including five replications at the center. The experimental error is estimated using replications, and the adequacy of a fitted model is confirmed. The misclassification cost of CART is considered as the response variable.

4.2.2. Conducting experiment

This step consists of four processes:

(a) Randomly split the data into the training data (D_1) and testing data (D_2). Do (b) and (c) for each fold.

(b) Sample and duplicate D_1 based on each generated combination in 4-2-1 to obtain a new data composition (D_3).

(c) Utilize cost-sensitive CART algorithm to construct a classification model using D_3; use the classification model to classify D_2.

(d) Calculate the misclassification cost as the response variable.

4.2.3. Fit a model and obtain optimal resampling strategy

The response surface model is obtained to demonstrate the relation between factor A, factor B, and the response variable, i.e. misclassification cost. The fitted model adequacies are confirmed by the lack-of-fit test, coefficient of determination (R^2), and adjusted coefficient of determination R^2 ($Adj - R^2$). Finally, the optimal resampling strategy for the majority class and minority class is obtained.

4.3. Cost-sensitive classification

One of the basic steps in the KDD process is to select method(s) to do searching for patterns in the data. This includes deciding which models may be appropriate, and matching a particular data mining method with the overall criteria of the KDD process. The end-user may be more interested in understanding the model than its predictive capabilities [2]. In the fraud detection concept, both goals (predicting and describing) are important. Thus we will use those algorithms that are easy to understand. In this work, the cost-sensitive C4.5 decision tree is used as the base learner of Adaboost (adaptive boosting). The AdaBoost algorithm has been proposed in 1997 by Yoav Freund and Robert Shapire as a general method for generating a strong classifier out of a set of weak classifiers [45].

5. Case study

In this section, we run our methodology with a real data from a CB bank[1]. This data was obtained from a large Brazilian bank and used in [46] and [47]. This dataset includes registers within four months' time window. One applies the following rule for classifying an authorization: a transaction is considered fraudulent if, in the next 2 months after the date of the transaction, which is called the performance period, either the client queried the transaction, or the bank distrusts it as a legitimate transaction and confirms it does not belong to the client; otherwise, the transaction is tagged as legitimate. When an authorization is tagged as fraudulent5, the bank has almost 100% of certainty about this claim, but when the transaction is tagged legitimate, but it can only be sure that the transaction was still not identified as fraudulent in the performance window. However, according to the bank, at least 80% of the occurred frauds are identified as fraudulent in a 2-month period [47].

[1] Detailed information is discarded due to privacy reasons.

5.1. Dataset

The sampling of transactions is done in two steps: first, one randomly samples card numbers to be analyzed in this period; secondly, there is a weighted random sampling of the classes where 10% of legitimate transactions and all fraudulent transactions are used. At the end, the database that has been received from the bank contains 41647 registers, from which 3.74% are fraudulent.

Next, statistical analysis has been applied to remove the variables that are considered unimportant for the modeling (ex: card number). From 33 variables in the beginning, 17 independent variables and one dependent variable (flag fraud) have been selected after this phase. Finally, all variables but Merchant Category Code (MCC) are categorized in at most 10 groups (see Table 1).

Table 1. Number of categories for each variable.

name	# of categ.	Att. type
Mcc	33	Nominal
mcc_previous	33	nominal
zip_code	10	nominal
zip_code_previous	10	nominal
value_trans	10	ordinal
value_trans_previous	10	Ordinal
pos_entry_mode	10	nominal
credit_limit	10	ordinal
brand	6	nominal
variant	6	nominal
Score	10	Ordinal
type_person	2	nominal
type_of_trans	2	nominal
# of statements	4	ordinal
speed	8	ordinal
diff_score	6	Ordinal
credit_line	9	ordinal
flag_fraud (class)	2	nominal

At the next step, 10 splits are generated from the databases. Each split contains a pair of datasets: 70% of transactions for development (training set), and 30% of transactions for testing. Table 2 shows that these splits have about the same number of frauds and legitimate transactions. We use these splits because the results of this paper can be comparable with the previous ones [46-47].

If we denote fp and fn as the number of false positives (false frauds) and false negatives, the misclassification cost of a classifier is defined by (3).

$$\text{Misclassification cost} = \frac{(10 \times fn) + fp}{N} \qquad (3)$$

5.2. Feature selection

Using the first split of table 2, two different methods are applied to do feature selection; the chi statistic and GA. In the χ^2 method, the chi-square test provides a method for testing the association between the row and column variables in a two-way table.

Table 2. Number of fraud and legitimates in each split.

Splits	Testing set		Training set	
	Legitimates	Frauds	Legitimates	Frauds
1	12184	475	27904	1084
2	12184	475	27904	1084
3	12076	467	28012	1092
4	12027	471	28061	1088
5	11943	484	28145	1075
6	12043	478	28045	1081
7	12115	443	27973	1116
8	11975	460	28113	1099
9	12204	453	27884	1106
10	11960	459	28188	1100

The null hypothesis H_0 assumes that there is no association between the variables (in other words, one variable does not vary according to the other variable), while the alternative hypothesis H_a claims that some association does exist. The alternative hypothesis does not specify the type of association, so a close attention to the data is required to interpret the information provided by the test. A high value of χ^2 indicates that the hypothesis of independence, which implies that expected and observed counts are similar, is incorrect. In this work, three unimportant features are selected after using the chi-square test at a 95% confidence level, features 12, 13, and 17.

The next feature selection method that is used in this phase is GA. The misclassification cost of a cost-sensitive CART is considered as the fitness of each chromosome. A generation of 50 chromosomes is repeated 30 times and the 6th feature is selected as the unimportant one.

At the next step, 50 random datasets are generated from the first dataset of table 1. The results of these two methods are applied to these datasets and the average misclassification cost has been obtained 2234 and 2158, respectively, for chi-square and GA. Thus, the result of the GA algorithm will be applied to other datasets (Section 5-3), since this method has a minor misclassification cost.

5.3. Different pre-processing strategies

In [48], the authors have added three new features to the dataset based on the clustering results, and

have shown that adding newly constructed features can improve the performance of DT and SVM significantly.

In this step, four pre-processing scenarios are constructed:

Strategy 1: the original dataset is used for the modeling (S_1).

Strategy 2: the result of k-means clustering method is added to the original dataset (S_2).

Strategy 3: the result of the feature selection phase is applied to the original dataset (the 6th feature will be removed) (S_3).

Strategy 4: the results of both the k-means and the feature selection phase are applied to the original dataset (S_4).

At the next step, different datasets are built from the original datasets of table 2 using upper different strategies, and the average of all strategies is compared pair-wisely.

5.4. Optimal resampling strategy

This work adopts the D-optimal design with a quadratic model to design an experiment that is used to obtain an appropriate resampling strategy. The D-optimal design is generated using the Design-Expert 8.0.7.1 computer software, in which a 20 run design is generated, including five replications at the center. The response variable is the misclassification cost of a cost-sensitive CART.

Using the first dataset of table 2, the factors of interest range from 1 to 26 ($r = [27904/1084]$). A D-optimal design with 20 combinations is generated using Design-Expert 8.0.7.1, as shown in table 3.

Next, the misclassification costs are calculated, as shown in the last column of table 3.

Table 4 summarizes the results of the lack-of-fit test of the fitted models with R^2 and $Adj-R^2$. In table 4, boldface represents the results of the selected quadratic model. The values of R^2 and $Adj-R^2$ for the quadratic model are 81.6% and 75.07%, respectively.

By utilizing the response surface model (as shown in Figure 2), the optimal factor-level combination of factor *A* and factor *B* is determined as (*A*, *B*) = (23.15, 19.72). Notably, the number of sampling instances is (the level of factor) * (the total number of the minority class in dataset).

5.5. Cost sensitive modeling

In this research, we have used of cost sensitive C4.5 decision tree as the base learner of Adaboost for the modeling. The robust parameters of [47] have used as the input parameters of all models. In this paper, we have used two different cost matrices, one for the training phase and the other for the testing phase. The cost matrix that is used for the testing phase is equal to C, where $C = \begin{bmatrix} 0 & 1 \\ 10 & 0 \end{bmatrix}$ (as we explained in Section 5.1). The cost matrix that is used for the training phase considers more costs for false-negative predictions, and is considered as $C' = \begin{bmatrix} 0 & 1 \\ 17.5 & 0 \end{bmatrix}$ after examining different cost-values. At the following step, the cost sensitive C.45 tree is used as the base learner of Adaboost algorithm with 10 replications using the *weka 3.7.10* computer software.

Table 3. Experiments and results for the dataset.

Run	Factor (1)	Factor (2)	Misclassification cost
1	13/5	13/5	1827
2	13/5	1	3961
3	1	26	2223
4	1	26	2536
5	26	26	1509
6	26	26	1800
7	26	1	3347
8	26	1	3054
9	13/5	26	2008
10	13/5	26	1686
11	7/25	19/75	1733
12	19/75	19/75	2143
13	1	9/33	2644
14	26	9/33	2246
15	1	1	3269
16	1	1	3984
17	1	17/67	2242
18	26	17/67	1869
19	7/25	7/25	1821
20	19/75	7/25	2011

Table 4. lack of fit tests and model summary statistics.

Source	df	p-value Prob > F	R^2	Adj-R^2	Predicted R^2
Linear	12	0/085	0/6035	0/5568	0/4698
2FI	11	0/074	0/6066	0/5328	0/4103
Quadratic	**9**	**0/249**	**0/8163**	**0/7507**	**0/6368**
Cubic	5	0/3144	0/8926	0/7960	0/3904

The results of applying the four strategies of Section 5.3 to the 9 data splits of table 1, with and without the resampling method (Section 5.4) are shown in table 5.

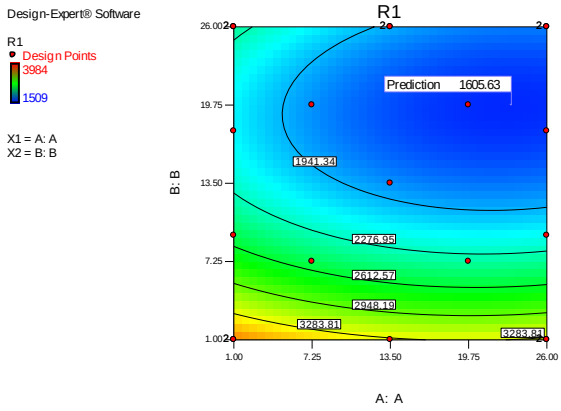

Figure 2. Response surface of dataset.

Table 5. Results of proposed methodology (cost-sensitive learning).

	Without resampling		With resampling	
Strategy	**Average cost**	**Standard deviation**	**Average cost**	**Standard deviation**
S_1	2029	0.89	1825	1.60
S_2	2029	1.60	1839	1.44
S_3	2005	0.96	**1743**	0.86
S_4	1949	1.12	1767	1.28
average	2003	1.15	1794	1.29

Table 5 shows that using the resampling method clearly reduces the average cost of each strategy. If we use simple c4.5 tree as the base learner of Adaboost algorithm, the result will be as table 6.

Table 6. Results of proposed methodology (without cost sensitive learning).

	Without resampling		With resampling	
Strategy	**Average cost**	**Standard deviation**	**Average cost**	**Standard deviation**
S_1	2179	0.91	1992	1.36
S_2	2179	1.25	1955	1.09
S_3	2150	1.37	1974	0.92
S_4	2137	1.11	1966	1.02
average	2161	1.16	1972	1.10

The average sensitivity and accuracy rate of all different strategies are shown in tables 7 and 8.

Table 7. Results of proposed methodology (cost-sensitive learning).

	Without resampling		With resampling	
Strategy	**Accuracy**	**Sensitivity**	**Accuracy**	**Sensitivity**
S_1	0.974	0.577	0.963	0.639
S_2	0.976	0.596	0.963	0.641
S_3	0.975	0.589	0.965	0.675
S_4	0.975	0.594	0.961	0.678
average	0.975	0.589	0.962	0.656

Table 8. Results of proposed methodology (without cost sensitive learning).

	Without resampling		With resampling	
Strategy	**Accuracy**	**Sensitivity**	**Accuracy**	**Sensitivity**
S_1	0.976	0.536	0.970	0.599
S_2	0.976	0.530	0.971	0.601
S_3	0.977	0.552	0.971	0.607
S_4	0.976	0.553	0.970	0.608
average	0.976	0.542	0.97	0.604

Using pairwise t-test, all strategies of tables 5 and 6 are examined, whether these differences of averages are statistically significant or not. The following results are obtained after using a 95% confidence level:

1) Using the resampling method significantly reduces the misclassification cost of all strategies (in cost-sensitive and without cost-sensitive learning) ($\text{p-value} < 0.002$).

2) Using cost-sensitive learning reduces the misclassification cost of all strategies, and this reduction is statistically significant ($\text{p-value} < 0.007$).

3) The result of cost-sensitive learning and resampling did not differ significantly.

4) In cost-sensitive learning, applying the result of feature selection phase significantly improves the misclassification cost ($\text{p-value} < 0.032$).

5) Using hybrid method averagely reduces the misclassification cost by 12.69% and 14.27%, comparing with resampling and cost-sensitive learning methods, respectively ($\text{p-value} = 0.00$).

6) Adding the result of k-means clustering to the dataset does not have a significant effect on the misclassification cost.

7) Using of matrix C' instead of C in the training phase significantly reduces the misclassification cost by averagely 4% (at 94% confidence level, $\text{p-value} < 0.060$).

Considering the third strategy of table 5 as the best result, steps of the proposed method of this work are suggested in figure 3.

In order to compare the results of this work with the previous ones, different algorithms of Gadi et al. (2008a) and Gadi et al. (2008b) are run 10 times by the *weka 3.7.10* software. The results obtained show that applying the proposed method significantly improves the misclassification cost of the compared classifiers ($\text{p-value}=0.00$). Detailed results for C4.5 Decision Tree (DT), Artificial Immune System (AIS), Bayesian Networks (BN), Neural Networks (NN), and Naïve Bayse (NB) algorithms are shown in table 9. All of these

methods used the misclassification cost metric of equation 3. The accuracy of the proposed method is equal to 96.59%. accuracies of DT, BN, and NB are respectively, equal to 93%, 94.3%, and 87.6% with sensitivities equal to 67%, 61%, and 67%.

Procedure proposed method (Dataset)
begin
do repeated Hold out k times
/* build $train_1$, …, $train_k$ and $test_1$, …, $test_k$ */
/* keep the ration of classes in each train and test data set constant */
begin preprocessing phase ($train_1$ and $test_1$)
(1) do cost sensitive Genetic based feature selection
(2) determine the optimal resampling strategy using cost sensitive S-RSM methodology
end
apply (1) to $train_2$,…, $train_k$ and $test_2$,…, $test_k$
apply (2) to $train_2$,…, $train_k$
learn model using $train_2$,…, $train_k$ /* use cost sensitive C4.5 tree as the base learner of Adaboost algorithm */.
test the learner model using $test_2$,…, $test_k$.
return average sensitivity and accuracy
end

Figure 3. Pseudocode of proposed method.

Table 9. Comparison of the results.

Strategy	DT	AIS	BN	NN	NB	Proposed method
Default	2042	3080	3057	2576	2820	**1743**
Optimized	2017	2156	2225	2207	2820	------
Roubust	2042	2192	2213	2127	2820	------

6. Conclusion

In this work, we made use of data mining and statistical tools in order to solve the problem of credit card fraud detection. The problem with a fraud data set is the skewed distribution of the classes that makes the learning algorithms ineffective, especially in predicting the minority class. Such datasets are called imbalanced datasets. Different algorithms have been proposed to solve the imbalanced learning problem, which falls largely into two major categories. The first one is data sampling and the second one is the algorithmic modification.

In this work, we used a hybrid approach that makes use of both categories. Our proposed process consists of three major phases: feature selection, resampling, and cost sensitive classification. Appropriate tools were employed commensurate to each phase. In the feature selection phase, two different methods were evaluated, chi-square and genetic algorithm. In the second phase, we used design of experiments and response surface analysis to determine the optimal resampling strategy. Finally, cost sensitive C4.5 tree was used as the base learner of Adaboost algorithm.

A large Brazilian banks' data was used as our case study to evaluate the proposed methodology. The performance of all classifiers was evaluated based on the misclassification cost metric. In order to examine the effectiveness of each proposed phase, different strategies were defined. The research findings showed that the proposed process had a high performance, and the resulting outcomes significantly reduced the misclassification cost compared with NN, DT, AIS, NB, and BN, by at least 14.62%. The accuracy and sensitivity of our proposed method were 96.59% and 67.52%, respectively. This shows that our hybrid proposed method has a good performance to detect fraud transactions compared with other data mining algorithms. Our proposed method works well because this method has incorporated both the data and algorithmic level approaches to deal with a high imbalanced dataset. If someone has access to other credit card datasets, it is recommended to compare this method with other proposed methods and report the results.

References

[1] Bolton, R. J., & Hand, D. J. (2002). Statistical Fraud Detection: A Review. Statistical Science, vol. 17, no. 3, pp. 235-255.

[2] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). Knowledge discovery and data mining: towards a unifying framework. 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland,OR, 1996.

[3] Kim, Y., & Sohn, S. Y. (2012). Stock fraud detection using peer group analysis. Expert Systems with Applications, vol. 39, no. 10, pp. 8986-8992.

[4] Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decis. Support Syst., vol. 50, no. 3, pp. 559-569.

[5] López, V., Fernández, A., Moreno-Torres, J. G., & Herrera, F. (2012). Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. Expert Systems with Applications, vol. 39, no. 7, pp. 6585-6608.

[6] Tong, L.-I., Chang, Y.-C., & Lin, S.-H. (2011). Determining the optimal re-sampling strategy for a classification model with imbalanced data using design of experiments and response surface methodologies.

Expert Systems with Applications, vol. 38, no. 4, pp. 4222-4227.

[7] Krawczyk, B., Woźniak, M., & Schaefer, G. (2014). Cost-sensitive decision tree ensembles for effective imbalanced classification. Applied Soft Computing, vol. 14, Part C, pp. 554-562.

[8] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. J. Artif. Int. Res., vol. 16, no. 1, pp. 321-357.

[9] Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem. 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Bangkok, Thailand, 2009.

[10] Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. International conference on Advances in Intelligent Computing - Volume Part I, Hefei, China, 2005.

[11] Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2012). DBSMOTE: Density-Based Synthetic Minority Over-sampling TEchnique. Applied Intelligence, vol. 36, no. 3, pp. 664-684.

[12] Zhang, D., Liu, W., Gong, X., & Jin, H. (2011). A Novel Improved SMOTE Resampling Algorithm Based on Fractal. Journal of Computational Information Systems, vol. 7, no. 6, pp. 2204-2212.

[13] Tomek, I. (1976). Two Modifications of CNN. IEEE Transactions on Systems, Man, and Cybernetics, vol. 6, no. 11, pp. 769-772.

[14] Wilson, D. L. (1972). Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. Systems, Man and Cybernetics, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-2, no. 3, pp. 408-421.

[15] Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. Newsl., vol. 6, no. 1, pp. 20-29.

[16] Thanathamathee, P., & Lursinsap, C. (2013). Handling imbalanced data sets with synthetic boundary data generation using bootstrap re-sampling and AdaBoost techniques. Pattern Recognition Letters, vol. 34, no. 12, pp. 1339-1347.

[17] Laurikkala, J. (2001). Improving Identification of Difficult Small Classes by Balancing Class Distribution. 8th Conference on AI in Medicine. AIME 2001. Lecture Notes in Computer Science, vol 2101. Springer, Berlin, Heidelberg, 2001.

[18] Liu, Y., Yu, X., Huang, J. X., & An, A. (2011). Combining integrated sampling with SVM ensembles for learning from imbalanced datasets. Information Processing & Management, vol. 47, no. 4, pp. 617-631.

[19] Qian, Y., Liang, Y., Li, M., Feng, G., & Shi, X. (2014). A resampling ensemble algorithm for classification of imbalance problems. Neurocomputing, vol. 143, pp. 57-67.

[20] Paasch, C. (2008). Credit Card Fraud Detection Using Artificial Neural Networks Tuned by Genetic Algorithms. (Doctoral Dissertation), Hong Kong University of Science and Technology (HKUST), Hong Kong.

[21] Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, vol. 50, no. 3, pp. 602-613.

[22] Lin-Tao, L., Na, J., & Jiu-Long, Z. (2008). A RBF neural network model for anti-money laundering. 6th International Conference on wavelet Analysis and Pattern Recognition. ICWAPR '08, Hong Kong, China, 2008.

[23] Xuan, L., & Pengzhu, Z. (2010). A Scan Statistics Based Suspicious Transactions Detection Model for Anti-money Laundering (AML) in Financial Institutions. International Conference on Multimedia Communications. Mediacom, Kraków, Poland, 2010

[24] Zhang, Z., Salerno, J. J., & Yu, P. S. (2003). Applying data mining in investigating money laundering crimes. 9th ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, D.C, 2003.

[25] Larik, A. S., & Haider, S. (2011). Clustering based anomalous transaction reporting. Procedia Computer Science, vol. 3, pp. 606-610.

[26] Jun, T., & Jian, Y. (2005). Developing an intelligent data discriminating system of anti-money laundering based on SVM. 4th International Conference on Machine Learning and Cybernetics, Guangzhou, China, 2005.

[27] Sánchez, D., Vila, M. A., Cerda, L., & Serrano, J. M. (2009). Association rules applied to credit card fraud detection. Expert Systems with Applications, vol. 36, no. 2, Part 2, pp. 3630-3640.

[28] Quah, J. T. S., & Sriganesh, M. (2008). Real-time credit card fraud detection using computational intelligence. Expert Systems with Applications, vol. 35, no. 4, pp. 1721-1732.

[29] Kirkos, E., Spathis, C., & Manolopoulos, Y. (2007). Data Mining techniques for the detection of fraudulent financial statements. Expert Syst. Appl., vol. 32, no. 4, pp. 995-1003.

[30] Duman, E., & Ozcelik, M. H. (2011). Detecting credit card fraud by genetic algorithm and scatter search. Expert Systems with Applications, vol. 38, no. 10, pp. 13057-13063.

[31] Somasundaram, A., & Reddy, U. S. (2018). Cost Sensitive Risk Induced Bayesian Inference Bagging

(RIBIB) for Credit Card Fraud Detection. Journal of Computational Science, vol. 27, pp. 247-254.

[32] Zhang, X., Han, Y., Xu, W., & Wang, Q. (2019). HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture. Information Sciences, in press.

[33] Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2019). Combining unsupervised and supervised learning in credit card fraud detection. Information Sciences, in press.

[34] Kim, E., Lee, J., Shin, H., Yang, H., Cho, S., Nam, S.-k., et al. (2019). Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning. Expert Systems with Applications, vol. 128, pp. 214-224.

[35] Correa Bahnsen, A., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. Expert Systems with Applications, vol. 51, pp. 134-142.

[36] Carneiro, N., Figueira, G., & Costa, M. (2017). A data mining based system for credit-card fraud detection in e-tail. Decision Support Systems, vol. 95, pp. 91-101.

[37] West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. Computers & Security, vol. 57, pp. 47-66.

[38] Karimi Zandian, Z., & Keyvanpour, M. R. (2019). MEFUASN: A Helpful Method to Extract Features using Analyzing Social Network for Fraud Detection. Journal of AI and Data Mining, vol. 7, no. 2, pp. 213-224.

[39] Kalyani, S., & Swarup, K. S. (2011). Particle swarm optimization based K-means clustering approach for security assessment in power systems. Expert Systems with Applications, vol. 38, no. 9, pp. 10839-10846.

[40] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, vol. 3, pp. 1157-1182.

[41] Ladha, L., & Deepa, T. (2011). Feature Selection Methods and Algorithm. International Journal on Computer Science and Engineering (IJCSE), vol. 3, no. 5, pp. 1787-1797.

[42] Pei, M., Goodman, E. D., Punch Iii, W. F. P., & Ding, Y. (1995). Genetic algorithms for classification and feature extraction. Annual Meeting, Classification Society of North America, North America, 1995.

[43] Montgomery, D. C. (2005). Design and analysis of experiment (6 ed.). New York: John Wiley and Sons.

[44] Manning, C. D., & Raghavan, P. (2008). Introduction to Information Retrieval: Cambridge University Press.

[45] Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119-139.

[46] Gadi, M. F. A., Xidi, W., & do Lago, A. P. (2008). Comparison with Parametric Optimization in Credit Card Fraud Detection.. Seventh International Conference on Machine Learning and Applications. ICMLA '08, San Diego, California, USA, 2008.

[47] Gadi, M., Wang, X., & Lago, A. (2008). Credit Card Fraud Detection with Artificial Immune System. International Conference on Artificial Immune Systems. ICARIS 2008. Lecture Notes in Computer Science, vol 5132. Springer, Berlin, Heidelberg.

[48] Farvaresh, H., & Sepehri, M. M. (2011). A data mining framework for detecting subscription fraud in telecommunication. Engineering Applications of Artificial Intelligence, vol. 24, no. 1, pp. 182-194.

ارائه یک مدل هوشمند برای شناسایی کلاهبرداری در کارت های اعتباری با استفاده از داده‌کاوی و روش‌های آماری

سکینه بیگی[۱،۲] و محمدرضا امین ناصری[۱،*]

[۱] دانشکده مهندسی صنایع و سیستم‌ها، دانشگاه تربیت مدرس، تهران، ایران.

[۲] گروه مهندسی صنایع، دانشکده علوم پایه و فنی مهندسی، دانشگاه کوثر بجنورد، بجنورد، ایران.

ارسال ۲۰۱۸/۱۰/۱۰؛ بازنگری ۲۰۱۹/۰۹/۰۳؛ پذیرش ۲۰۱۹/۱۱/۲۳

چکیده:

امروزه، با پیشرفت تکنولوژی و پیچیده‌تر شدن فرآیندهای کسب‌وکار، روش‌های انجام و در نتیجه شناسایی کلاهبرداری‌ها نیز به مراتب پیچیده‌تر شده‌اند. با توجه به حجم عظیم داده‌های موجود در بانک‌ها، ردیابی عملیات‌های مجرمانه حتی با در نظر گرفتن منابع انسانی بسیار به صورت دستی امکان‌پذیر نیست و نیاز به ابزارهای جدید در این زمینه دارد. در سال‌های گذشته ثابت شده است که ابزارهای داده‌کاوی نسبت به روش‌های آماری از کارایی قابل‌توجهی به خصوص در حوزه مالی برخوردارند. در این تحقیق، با ارائه روشی شامل ابزارهای مختلف داده‌کاوی به بررسی شناسایی کلاهبرداری در کارت‌های اعتباری پرداخته شده است. روش پیشنهادی این تحقیق، شامل سه بخش عمده انتخاب مشخصه‌های مهم، تعیین استراتژی بهینه نمونه‌برداری و مدل‌سازی حساس به هزینه است. در بخش نخست از روش پیشنهادی، از الگوریتم ژنتیک برای تعیین مشخصه‌های مهم استفاده شده است. در ادامه، با استفاده از روشی مبتنی بر طراحی آزمایش‌ها نسبت بهینه هر یک از دسته‌ها برای انجام بازنمونه‌برداری تعیین شده است. در بخش مدل‌سازی نیز از روش درخت تصمیم C4.5 حساس به هزینه به عنوان دسته‌بند پایه در الگوریتم آدابوست استفاده شده است. در پایان، با استفاده از یک مجموعه داده واقعی نشان داده شده است که روش پیشنهادی تحقیق با حداقل ۱۴ درصد کاهش هزینه دسته‌بندی اشتباه به طور معناداری نتیجه بهتری نسبت به روش‌های درخت تصمیم، بیزی ساده، شبکه بیزی، شبکه عصبی و سیستم ایمنی مصنوعی داشته است.

کلمات کلیدی: شناسایی تقلب، کارت‌های اعتباری، انتخاب مشخصه، بازنمونه‌برداری، یادگیری حساس به هزینه.