2018 IEEE International Conference on Systems, Man, and Cybernetics

Credit Card Fraud Detection Using Capsule Network

Shuo Wang, Guanjun Liu, Zhenchuan Li, Shiyang Xuan, Chungang Yan, Changjun Jiang
Department of Computer Science, Tongji University, Shanghai, China
{wangshuo, liuguanjun, 1510482, xsyfor, yanchungang, cjjiang}@tongji.edu.cn

***Abstract*—Credit card is now popular in daily life. Meanwhile, credit card fraud events occur more frequently, which result in massive financial losses. There are a number of fraud detection methods, but they do not deeply mine features of customer's transaction behavior so that their detection effectiveness is not too desirable. This paper focuses on two aspects of feature mining. Firstly, the features of credit card transactions are expanded in time dimension to characterize the distinct payment habits of legal users and criminals. Secondly, Capsule Network (CapsNet) is adopted to further dig some deep features on the base of the expanded features, and then a fraud detection model is trained to identify if a transaction is legal or fraud. Through experiments on a real transaction dataset, we demonstrate that the time dimension extension can improve the performance of fraud detection, and then CapsNet is further illustrated to be more advantageous in fraud detection compared with other models.**

***Index Terms*—credit card, fraud detection, feature, Capsule network**

I. Introduction

From the 2008-2013 research report in [1], we can see that Singapore's non-cash payment accounts for 61%, and the United States is 45%. In 2016, 1 out of 7 people in the UK no longer carry or use cash [2]. When people buy goods and services, credit card transactions are the most common form of cashless presentation. According to the 2016 U.S. Consumer Survey, 75% of respondents prefer to use a credit or debit card as a payment method. Credit card has become one of the most important cashless payment instruments. However, global financial losses caused by credit card fraud in 2015 reached a staggering $21.84 billion [1]. A series of models for detecting fraud transactions have been proposed, including expert systems [12], machine learning [9], [13]–[17], [24]–[27], and deep learning [4], [18]–[20]. Simultaneously, engineering of dimensionality reduction [5]–[7] and feature expansion [3], [4] for fraud transaction detecting has also been paid attention to, since it is an important way to improve the effectiveness of fraud detection. However, the existing methods have not achieved a very desirable effectiveness.

This paper presents a feature expand method inspired by the work in [4]. Every transaction record of a user is converted into a feature matrix associated with the previous transactions of the user. In the matrix, features are expanded in time dimension and can describe the consumption patterns of legitimate and fraudulent transaction comprehensively. Since the feature matrix is so complex and expressive, a more powerful feature mining model should be applied to capture more distinct features. Therefore, this paper introduces the capsule network (CapsNet) [23] for the first time in fraud detection problem. CapsNet is able to represent various attributes of a particular entity (such as position, size, and texture) via different capsules and achieve the state-of-the-art results in many datasets for image recognition. It is expectable that CapsNet can catch more distinctive deep features to identify fraudulent transactions form feature matrix designed in this paper.

The rest parts of this paper are organized as follows. In Section II, we review the studies on credit card fraud detection. Then the capsule network is introduced in Section III. Section IV describes how a transaction record is expanded to a feature matrix in time dimension. The details of experiments are shown in Section V. Finally, a summary of this paper is provided in Section VI.

II. Related Work

For credit card fraud problem, researchers usually deal with it through feature engineering and model selection.

A. Feature Engineering

Feature engineering is the first phase for handling fraud detection problem. The number of fraud transactions is much less than legitimate ones. Too few samples of fraud transactions can lead to a high rate of false detection or make them be ignored as noise [8]. Literature [9] uses two approaches, sampling method and cost-based method, to address class imbalance. Literatures [10] and [11] focus on the concept drift, that is, users' transaction habits will change over time and then affect their statistical characteristics. Literature [10] proposes a new generation of JIT classifiers to deal with recurrent concept drift. Literature [6] uses feature subset selection and feature ranking methods to remove some related and redundant features from a large number of features for reducing the computational cost. Literature [3] evaluates the performance and impact of several feature selection techniques (CFS, Gain Ratio, Relief) in the scenario of web transactions. Literature [4] transforms transaction records into feature matrices and then intrinsic patterns of fraud can be captured by classifiers. In [5], it is found that the same user's daily spending time

[1]The Nilson Report (October 2016) [Online]. Available: https://www.nilsonreport.com/upload/content_promo/The_Nilson_Report_10-172016.pdf

2577-1655/18/$31.00 ©2018 IEEE
DOI 10.1109/SMC.2018.00622

was concentrated in a certain period of the day, and a feature related to that is added. The features in the above literatures are just expanded slightly, and only a small number of new features are generated from the original features, which cannot fully characterize the pattern of user consumption. This paper inspired by the work in [4], uses different time windows to enrich the number of features.

B. Model Selection

Decision tree classification method is simple and intuitive, and it is also the earliest method used for fraud detection of credit card transactions. Kokkinaki et al. [24] use decision trees and Boolean logic functions to describe cardholders' spending habits in normal transactions. Then they use a clustering method to analyze the difference between normal transactions and fraudulent ones. Finally, the trained model distinguishes whether each cardholder's credit card transaction is normal or not. Random Forest, an ensemble learning method, is firstly proposed by Leo Breiman. It does a final decision by integrating a series of decisions made by its base classifier and achieves better results. Chao and Leo Breiman et al. propose a random forests method to detect fraud under considering the unbalanced data [25].

The neural network algorithm is the artificial intelligence algorithm and can also be used in credit card anti-fraud system. Aleskerov et al. use a hidden layer, self-organizing neural network with the same number of input and output units to conduct anti-fraud research [13]. Aleskerov et al. propose a fuzzy neural networks method to mine the abnormal transactions. By only analyzing the fraudulent transaction data and parallel processing at the same time, fuzzy neural networks can rapidly generate fraudulent regularity information [26].

Bhinav Srivastava [16] uses Hidden markov model (HMM) to model a customer's history normal spending behavior. For a new transaction, if the fluctuation of transaction sequence is relatively large, it is considered to be a fraud.

In recent years, with the rapid development of deep learning, their application scenarios have penetrated into all walks of life and also been quickly introduced into credit card fraud detection [4], [17]–[20]. Literature [19] proposes a dynamic machine learning method to simulate the inherent time series transaction sequence of the same card, and then LSTM method is applied. Literature [4] uses a convolutional neural network to classify normal and abnormal transactions. But more specific indicators such as recall and precision should be given in the paper.

All work of feature engineering and machine learning provides very useful experience for fraud detection. With the development of feature engineering, the fraud detecting models should be boosted synchronously. This paper presents a method of feature extension in time dimension and applies a powerful feature digging model, CapsNet, on this expended features. Through some experiments on a real transaction database from a financial company in China, the effectiveness of our feature extension method and the outperformance of CapsNet are illustrated in this paper.

III. Capsule Network

Capsule network is proposed by Hinton et al. [23] based on convolutional neural networks (CNN). A capsule is a set of neurons whose activity vectors represent instantiation parameters of a certain type of entity, such as an object part or an entire object. The activity of neurons in an active capsule represents a variety of attributes of an entity. These attributes can include different types of instantiation parameters such as position, size, orientation, deformation, velocity, albedo and color. The length of the activity vector means the probability that the entity exists and the direction represents parameters of the instantiation. In what follows, we introduce capsule network briefly, and please refer to [24] for more details.

A. Capsule network framework

Fig.1. shows the architecture of the capsule network. The first layer Conv1 is convolutional layers, and the second layer is PrimaryCaps. DigitCaps is the last layer and is fully connected layer. Conv1 has 256 kernels with stride of 1 and ReLU activation, and the kernel size is 9×9. The outputs of Conv1 are delivered into the successive layer PrimaryCaps. The convolution results from Conv1 are divided into 32 groups (channels) and each group has 8 convolutional units with a 9×9 kernel and a stride of 2. The outputs of PrimaryCaps consist of $[32 \times 6 \times 6]$ capsules. Each output is an 8-dimensional vector, and each capsule in $[6 \times 6]$ grid shares the weights with each other. The vectors will apply Eq.(1). to ensure that the length is in the range of [0,1] where the length means the probability of entity presence. DisgitCaps is the last layer that has 10 class with a 16-dimensional capsule. Each class represents one digit.

$$V_j = \frac{\| S_j \|^2}{1 + \| S_j \|^2} \frac{S_j}{\| S_j \|} \tag{1}$$

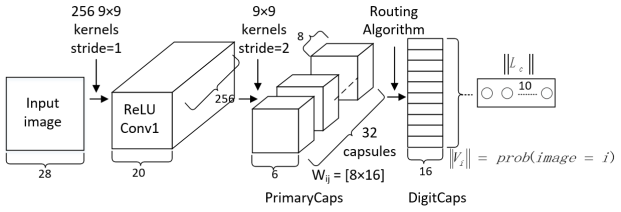

Fig. 1. Architecture of capsule network.

B. Iterative routing-by-agreement mechanism

Between PrimaryCaps and DigitCaps, the routing-by-agreement algorithm is applied. In deep learning, we use backward propagation to train model parameters. In capsule network, the transformation matrix W_{ij} are still trained during back propagation. However, the correlation coefficient c_{ij} are calculated using a new iterative routing method. The prediction vector $\hat{u}_{j|i}$ is computed as follows (with the transformation matrix):

$$\hat{u}_{j|i} = W_{ij} u_i \tag{2}$$

in which u_i is the activity vector for the capsule i in the layer below. The activity vector v_j for the capsule j in the layer above is computed as follows:

$$S_j = \sum_i c_{ij}\hat{u}_{j|i}, \quad V_j = \frac{\| S_j \|^2}{1+\| S_j \|^2}\frac{S_j}{\| S_j \|} \tag{3}$$

Intuitively, prediction vector $\hat{u}_{j|i}$ is the prediction (vote) from the capsule i on the output of the capsule j above. If the activity vector has a strong similarity to the prediction vector, we conclude that capsule i is highly correlated with the capsule j. For example, eye capsules are closely related to facial capsules. This similarity is measured by the scalar product of the prediction and activity vector. Therefore, the similarity considers both likeness and feature attributes. We calculate a relevancy score b_{ij} based on the similarity:

$$b_{ij} \leftarrow \hat{u}_{j|i} \cdot v_j \tag{4}$$

The coupling coefficient c_{ij} is calculated as the softmax of b_{ij}:

$$c_{ij} = \frac{e^{b_{ij}}}{\sum_k e^{b_{ik}}} \tag{5}$$

In order to make b_{ij} more accurate, it is iteratively updated over multiple iterations (usually in 3 iterations):

$$b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j \tag{6}$$

Algorithm 1 describes the routing pseudo code.

Algorithm 1 routing-by-agreement($\hat{u}_{j|i}, k, l$)

1: for every capsule j in layer $l+1$ and capsule i in layer l: $b_{ij} \leftarrow 0$.
2: **for** k iterations **do**
3: for every capsule i in layer l: $c_i \leftarrow$ softmax(b_i) $\circ$ corresponds to Eq.(5)
4: for every capsule j in layer $l+1$: $s_j \leftarrow \sum_i c_{ij}\hat{u}_{j|i}$
5: for every capsule j in layer $l+1$: $v_j \leftarrow$ squash(s_j) $\circ$ corresponds to Eq.(1)
6: for every capsule i in layer l and cpasule j in layer $l+1$: $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j$
7: **end for**
8: **return** v_j

C. Margin loss for digit existence

We use the length of the instantiation vector to represent the probability that the capsule entity exists and use a separate margin loss L_c for each digit capsule k:

$$\begin{aligned} L_c =& T_c max(0, m^+ - \| v_c \|)^2 + \\ & \lambda(1-T_c)max(0, \| v_c \| - m^-)^2 \end{aligned} \tag{7}$$

in which $T_c = 1$ if an object of class c is present. M^+ is set to 0.9, m^- is set to 0.1. The down-weighting λ (default 0.5) stops the initial learning from shrinking the activity vectors of all classes. The total loss is just the sum of the losses of all classes.

D. Reconstruction as a regularization method

The extra reconstruction loss is used to encourage the digital capsule to encode the instantiated parameters of the input number. In the training phase, the digit capsule will be masked out but the activity vector of correct capsule. This activity vector is used to reconstruct the input image. Fig.2 shows that the output of the digit capsule is fed into a decoder that models the pixel intensities. The decoder is consisting of 3 fully connected layers. A reconstruction loss ‖image - reconstructed image‖ is added to the marginal loss function. The reconstruction loss is multiple by a regularization factor (0.0005) so that it does not dominate over the marginal loss.

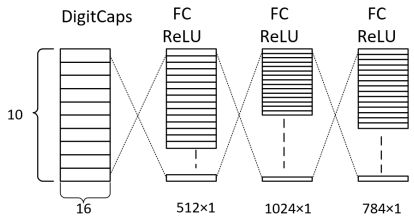

Fig. 2. Decoder structure.

IV. Feature engineering

In this part, we present a feature extension method that transforms every transaction into a more expressive feature matrix from the aspects of different time dimensions. We can define Avg_Amount_T as the average transaction amount for the same customer over a period of time T where T represents the length of the time window. We can set T to be different values: one hour, one day, one week and one month to describe feature with different time spans. In the experiment, the value of T ranges from 5 seconds to 2 months. Then we can get four features corresponding to four time windows. There are four types features need to extend: 1) TypeA, the most commonly used type, 2) TypeB, the gini impurity change of certain attribute, 3) TypeC, the transaction frequency related to features, and 4) TypeD, the transaction amount related to features.

TypeA only contains one feature: Checking Method. One-hot encoding is applied to deal with the several different values of Checking Method.

Four attributes belong to TypeB: MAC address, merchant ID, phone number, and transactions in different hours. In online credit card transactions, if one user's credit card information is stolen, the criminal possibly use different electronic equipment in order to increase the gini impurity of the MAC address during T. Just as stated in [5], a user's behavior tends to concentrate in a specific time in a day, and thus a fraudulent transaction may fall outside of this specific period which makes the gini impurity of transactions in different hours changed. For example, assuming that the value of T is one month. The total number of transactions for a user in the month prior to the current transaction record is $Count\_In\_T$. The number of transactions belonging to the i-th hour is H_i

(i=1,2,3,...,24), and then the proportion of transactions in the i-th hour is P_i:

$$P_i = \frac{H_i}{Count\_In\_T} \tag{8}$$

Then, the transaction gini impurity value distributed in each hour can be defined as follows:

$$GiniT = 1 - \sum_{i=1}^{24} p_i^2 \tag{9}$$

The above gini impurity does not include the new transaction. Then we add the current transaction to the above calculation in order to obtain the current gini impurity: New_Gini_T. Therefore, the change of gini impurity is defined as follows:

$$Change\_Of\_Gini = New\_Gini\_T - GiniT \tag{10}$$

If the gini impurity increases too much, the related transaction is thought of as a fraud.

TypeC includes "whether there is another transaction during past T" and "the number of transactions in past T". Because most of fraudsters always want to transfer victims money to other accounts as soon as possible, so these two attributes are used to detect the case of replay attacks.

TypeD includes 7 extended attributes: 1) the total transactions amount in a day divided by the one-day transactions limit, 2) the current transaction amount divided by the single transaction limit, 3) the average transactions amount in the past T time, 4) maximum transaction amount in the past T time, 5) the transaction divided by the maximum transaction amount in the past T time, 6) the transaction divided by the average transaction amount in the past T time, 7) the total transaction amount in the past T time. Tab.1 shows the example of feature matrix. We randomly selected four normal and four unnormal samples, and then generate their heat maps after feature conversion as shown in Fig.3. The upper four are normal, and the bottom for are fraudulent. We can see that the normal consumption pattern is more varied.

TABLE I
EXAMPLE OF FEATURE MATRIX

	One Hour	One Day	One Week	...
Avg_Amount_T				
Total_Amount_T				
Max_Amount_T				
Most_Checking_Method				
Gini_Change_MAC				
...				

V. EXPERIMENT

In this section, we describe the dataset, introduce the evaluation criteria of the model, design the experiment, and discuss the experimental results.

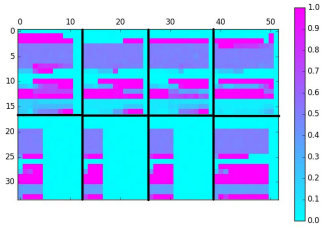

Fig. 3. Feature matrix.

A. Dataset

Our dataset is about online credit card transactions provided by a financial company in China. The data collection spans from April 2017 to June 2017, with approximately 3.5 million records, including approximately 65 thousand fraud records. Feature matrix construction requires users have a sufficient quantity of historical transaction records, and we filter the entire data. The filtered transaction records are divided into two parts: the transaction data in April and May is PartI, and the transaction data in June as the testing set. We sample 120 thousand records from PartI including 24,063 fraudulent records. There are about 780 thousand transactions in the test set including about 16 thousand fraudulent transactions.

B. Performance measures

We choose recall, precision, accuracy, and F1 score as evaluation criteria. Recall rate measures the detection rate of all fraud cases and precision rate is a measure of the result of prediction. F1 score is a weighted harmonic mean of accuracy and recall, with a nonnegative weight:

$$F1 \quad socre = \frac{(\beta^2 + 1) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{11}$$

β is commonly set to 1.

C. Experiment I

The dataset for this experiment is described in Subsection A. The data is processed through simple conventional method and each original record is not converted into a feature matrix. A set of classifiers including Support Vector Machines (SVM), Random Forest (RF) and Neural Networks (NN), are used in the experiment. The purpose of this experiment is to provide a benchmark for further experiments. Tab.II shows the results produced by SVM, RF and NN. The recall rates of the three algorithms are 75.00%, 90.14%, and 83.89%, respectively, and RF achieves the highest F1 score and the highest precision. NN has the lowest precision, only 19.00%, which leads to the lowest F1 score. So it is obvious that, RF achieves the best results, followed by SVM.

D. Experiment II

The dataset of this group of experiments also uses the data set introduced in Subsection A, but the feature processing is different from the experiment I. The process of fraud detection

TABLE II
RESULT OF EXPERIMENT I

Classifier	Recall	Precision	Accuracy	F1 score
SVM	75.00%	37.74%	97.09%	50.21%
RF	90.14%	44.84%	97.63%	59.88%
NN	83.89%	19.00%	95.93%	30.99%

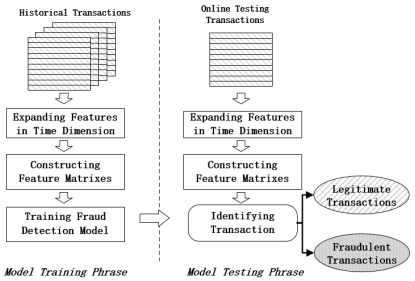

Fig. 4. Credit card fraud detection system.

is shown in Fig.4. Each record in the data set is expanded according to the time window to form a feature matrix, as described in Section IV. The three algorithms SVM, RF and NN are still used. Tab.III presents the results. And their recall rate does not change too much. However, the precision rates are much higher than that in Tab.II, increasing about 30%. Their F1 scores also increases by 22.09%, 18.40% and 32.04%, respectively. In addition to the above three classifiers, we also used two additional methods. One is the CNN mentioned in [4], and a other one is the capsule network we introduced in Section III. Capsule network achieves the highest recall, precision, accuracy, and F1 score. The CNN model gets the second highest recall rate. Fig.5 shows the CapsNet and CNN ROC curves, and AUC of CapsNet is slightly higher than CNN's.

TABLE III
RESULT OF EXPERIMENT II

Classifier	Recall	Precision	Accuracy	F1 score
SVM	73.15%	71.46%	98.90%	72.30%
RF	87.32%	70.93%	99.05%	78.28%
NN	86.00%	49.75%	98.03%	63.03%
CNN	92.40%	67.35%	98.99%	78.09%
Capsule Network	95.20%	72.67%	99.21%	82.41%

E. Experiment III

In the capsule network experiment, the training phase is iterated for a total of 50 times. Fig.6 shows the loss values during the training phase and Fig.7 shows the accuracy values of the training and validation sets for different iterations. There are three loss curves in Fig.6, namely loss, capsnet_loss, and decoder_loss. Loss is the sum of margin loss and reconstruction loss introduced in the third part of the paper. Capsnet_loss is margin loss. After 50 iterations, decoder loss is close to 0. In Fig.8, the top half is the feature matrices corresponding to the

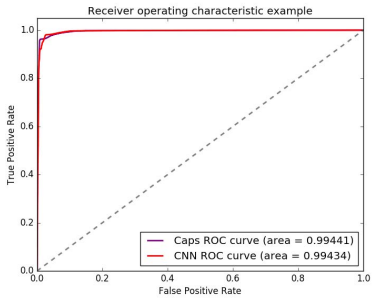

Fig. 5. ROC curve of CNN and CapsNet.

four real transaction records. The bottom half is the feature matrices after the decoder layer. The decoder can complete the reconstruction of the input image. After 30 iterations, the losses basically remain unchanged. Fig.7 shows that after 10 iterations, the accuracy of the validation set does not change. However, the accuracy of the training set rises ocillatingly.

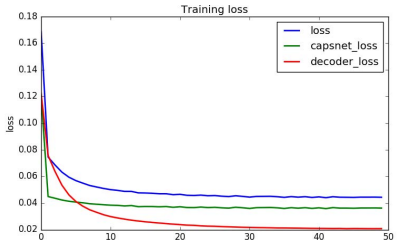

Fig. 6. Training loss.

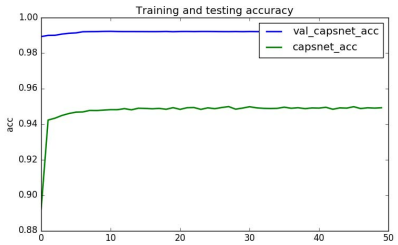

Fig. 7. Training and validation accuracy.

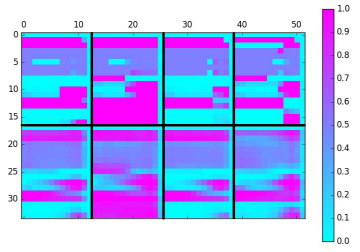

Fig. 8. real and reconstructed feature matrix.

F. Discussion

By comparing the SVM, RF, and NN algorithms in Experiments I and II, the transaction feature matrix can significantly improve the precision of the experiment, thus increase the F1 socre, while the recall does not change too much. In Experiment II, the CapsNet has the best performance. Each capsule can represent an attribute of an entity. It can classify normal and fraudulent transactions in a more conscious way. After calculating the correct digital capsule activity vector, we feed a perturbed version of this activity vector to the decoder network and see how the perturbations affect the reconstruction. Fig.9 indicates that a certain capsule of one transaction is disturbed. It can be seen that this capsule can represent the "drift" of a certain transaction pattern of a user.

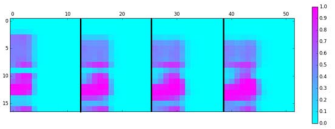

Fig. 9. perturbed feature matrix.

VI. CONCLUSION

This paper presents a feature extension method to describe users' payment behavior more expressively. Simultaneously, a powerful feature extracting model Capsule Network is applied to improve the effectiveness for fraud detection. Through detailed experiments on a real B2C transaction dataset from a financial company in China, both the efficacy of our feature extension method and the superiority of Capsule Network for detecting fraudulent transactions are verified. Although the Capsule Network achieves the best performance, it still has some limitations, e.g., big time consumption. In the future work, we will focus on the adjustment of Capsule Network to fit the fraud detection problem.

ACKNOWLEDGMENT

This paper was supported in part by National Key R&D Program of China (Grant No. 2017YFB1001804), Shanghai Science and Technology Innovation Action Plan Project (Grant No. 16511100900) and the National Natural Science Foundation of China (Grant No. 61332008). Corresponding authors are G.J. Liu and C.J. Jiang.

REFERENCES

[1] Thomas, Hugh, Amit Jain, and Michael Angus. "Measuring progress toward a cashless society." MasterCard Advisors (2013).

[2] Alex West (23 October 2016). "One in seven Brits no longer carries cash, as we become increasingly reliant on card and smartphone payments". thesun.co.uk. The Sun. Retrieved 12 November 2016.

[3] Lima, Rafael Franca, and Adriano CM Pereira. "Feature Selection Approaches to Fraud Detection in e-Payment Systems." International Conference on Electronic Commerce and Web Technologies. Springer, Cham, 2016.

[4] Fu, Kang, et al. "Credit card fraud detection using convolutional neural networks." International Conference on Neural Information Processing. Springer, Cham, 2016.

[5] Bahnsen, Alejandro Correa, et al. "Feature engineering strategies for credit card fraud detection." Expert Systems with Applications 51 (2016): 134-142.

[6] Fadaei Noghani, F., and M. Moattar. "Ensemble Classification and Extended Feature Selection for Credit Card Fraud Detection." Journal of AI and Data Mining 5.2 (2017): 235-243.

[7] Ise, Masayuki, Ayahiko Niimi, and Osamu Konishi. "Feature selection in large scale data stream for credit card fraud detection." (2009).

[8] Lpez, Victoria, et al. "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics." Information Sciences 250 (2013): 113-141.

[9] Dal Pozzolo, Andrea, et al. "Credit card fraud detection: a realistic modeling and a novel learning strategy." IEEE transactions on neural networks and learning systems (2017).

[10] Alippi, Cesare, Giacomo Boracchi, and Manuel Roveri. "Just-in-time classifiers for recurrent concepts." IEEE transactions on neural networks and learning systems 24.4 (2013): 620-634.

[11] Gama, João, et al. "A survey on concept drift adaptation." ACM computing surveys (CSUR) 46.4 (2014): 44.

[12] Leonard, Kevin J. "Detecting credit card fraud using expert systems." Computers & industrial engineering 25.1-4 (1993): 103-106.

[13] Aleskerov, Emin, Bernd Freisleben, and Bharat Rao. "Cardwatch: A neural network based database mining system for credit card fraud detection." Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997. IEEE, 1997.

[14] Bahnsen, Alejandro Correa, Djamila Aouada, and Björn Ottersten. "Example-dependent cost-sensitive decision trees." Expert Systems with Applications 42.19 (2015): 6609-6619.

[15] Whitrow, Christopher, et al. "Transaction aggregation as a strategy for credit card fraud detection." Data Mining and Knowledge Discovery 18.1 (2009): 30-55.

[16] Srivastava, Abhinav, et al. "Credit card fraud detection using hidden Markov model." IEEE Transactions on dependable and secure computing 5.1 (2008): 37-48.

[17] Randhawa, Kuldeep, Chu Kiong Loo, and Manjeevan Seera. "Credit card fraud detection using AdaBoost and majority voting."

[18] Pumsirirat, Apapan, and Y. Liu. "Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine." International Journal of Advanced Computer Science & Applications 9.1(2018).

[19] Wiese, Bnard, and Christian Omlin. "Credit card transactions, fraud detection, and machine learning: Modelling time with LSTM recurrent neural networks." Innovations in neural information paradigms and applications. Springer, Berlin, Heidelberg, 2009. 231-268.

[20] Fiore, Ugo, et al. "Using Generative Adversarial Networks for Improving Classification Effectiveness in Credit Card Fraud Detection." Information Sciences (2017).

[21] Lu, Qibei, and Chunhua Ju. "Research on credit card fraud detection model based on class weighted support vector machine." Journal of Convergence Information Technology 6.1 (2011).

[22] Jurgovsky, Johannes, et al. "Sequence classification for credit-card fraud detection." Expert Systems with Applications 100 (2018): 234-245.

[23] Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic routing between capsules." Advances in Neural Information Processing Systems. 2017.

[24] Kokkinaki, Angelika I. "On atypical database transactions: identification of probable frauds using machine learning for user profiling." Knowledge and Data Engineering Exchange Workshop, 1997. Proceedings. IEEE, 1997.

[25] Chen, Chao, Andy Liaw, and Leo Breiman. "Using random forest to learn imbalanced data." University of California, Berkeley 110 (2004): 1-12.

[26] Syeda, Mubeena, Yan-Qing Zhang, and Yi Pan. "Parallel granular neural networks for fast credit card fraud detection." Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on. Vol. 1. IEEE, 2002.

[27] Kundu, Amlan, et al. "Blast-ssaha hybridization for credit card fraud detection." IEEE transactions on dependable and Secure Computing 6.4 (2009): 309-315.