An Adaptive Approach on Credit Card Fraud Detection Using Transaction Aggregation and Word Embeddings

Ali Yeşilkanat(✉), Barış Bayram, Bilge Köroğlu, and Seçil Arslan

Applied AI and R&D Department, Yapi Kredi Technology, Istanbul, Turkey
ali.yesilkanat@ykteknoloji.com.tr

Abstract. Due to the surge of interest in online retailing, the use of credit cards has been rapidly expanded in recent years. Stealing the card details to perform online transactions, which is called fraud, has also seen more frequently. Preventive solutions and instant fraud detection methods are widely studied due to critical financial losses in many industries. In this work, a Gradient Boosting Tree (GBT) model for the real-time detection of credit card frauds on the streaming Card-Not-Present (CNP) transactions is investigated with the use of different attributes of card transactions. Numerical, hand-crafted numerical, categorical and textual attributes are combined to form a feature vector to be used as a training instance. One of the contributions of this work is to employ transaction aggregation for the categorical values and inclusion of vectors from a character level word embedding model which is trained on the merchant names of the transactions. The other contribution is introducing a new strategy for training dataset generation employing the sliding window approach in a given time frame to adapt to the changes on the trends of fraudulent transactions. In the experiments, the feature engineering strategy and the automated training set generation methodology are evaluated on the real credit card transactions.

Keywords: Fraud detection · Imbalanced data · Concept drift · Decision system · Character-level word embedding

1 Introduction

With the advances in information technology and electronic commerce, the use of credit cards raises in recent years. According to the fifth report of The Single European Payments Area (SEPA) report in 2016 [1], the total value of the fraudulent transactions was €1.8 billion, which 73% of this value comes from Card-Not-Present (CNP) payments. Comparing to the ATM and POS frauds, CNP fraud is the one which increases most by 2.1% over four years. Therefore, CNP fraud is a considerably serious problem in the credit card business.

Published by Springer Nature Switzerland AG 2020
I. Maglogiannis et al. (Eds.): AIAI 2020, IFIP AICT 583, pp. 3–14, 2020.
https://doi.org/10.1007/978-3-030-49161-1_1

While developing a fraud detection model, numerous problems are encountered such as:

- generation of training set from hundreds of millions of imbalanced transactions,
- selection of the most appropriate feature combination due to indistinguishable features between fraud and non-fraud instances,
- skewness of the data and cost of the false-positive samples,
- durability of the model being affected by trend variation called concept drift caused by changes on behaviors of the customers and fraudsters.

Data imbalance is defined as possessing an unequal distribution of classes within a dataset. The transactions from credit cards commonly form an imbalanced dataset, including very few fraud transactions comparing to legitimate ones. In order to solve this problem, various methods are provided, namely undersampling [15] and oversampling [4]; however, these techniques are problematic because the dataset is massively imbalanced and instances of the dataset individually carry important information (such as transactions belonging to the same credit card). In this work, we introduce a card-based equalization method on fraud and non-fraud cards to form a training dataset by incorporating all transactions of the sampled cards to provide a solution to the mentioned problems.

Properties of the fraudulent credit card transactions usually alter over time. The main reason for this is that fraudsters try to bypass fraud detection systems. Moreover, purchasing trends of online markets vary over time, such as the establishment of a new payment channel, new merchant, or merchant category. Those trend shifts are called concept drift, where the distribution of a data stream, is not stationary. The conventional fraud detection systems are developed using previous fraud transactions and cannot adapt to concept drift. In this work, the proposed system is retrained by itself over time to prevent concept drift adaptively. In this study, we propose a sliding window-based automated training dataset generation technique to solve this issue.

The transactions of a credit card can be performed from various sources, named terminals. These terminals transfer the data related to the transaction to the owner bank of the card in ISO 8538 Standard [9]. Generally, the sender terminal feeds the data in a structural and standardized format; yet, some attributes on the transactions are not sent correctly which may cause problems for fraud detection systems. For instance, the same merchant name are retrieved differently from the related property of the transaction, like, $FACEBOOK$ or $FACEBK$. Therefore, a character-level word embedding is required to map the name to a vector of real numbers. In this way, the name of the merchant can be used as a distinctive feature to detect fraudulent behaviour.

The online transactions are often real-time events, in which a fraud decision system has to determine if it is a fraud or not in milliseconds. In this work, with the purpose of designing a fast and reliable credit card fraud detection system, a Gradient Boosting Tree (GBT) based approach is developed with different types of features. Each transaction includes the typical numeric, hand-crafted numeric, categorical, and textual features.

The rest of the paper is organized as follows: In Sect. 2, the related work is discussed. The modules of the proposed credit card fraud detection system and the processing modules are described in Sect. 3. Section 4 is devoted to our results of the experiments to evaluate the importance of feature engineering and automated training set generation approach. Finally, the results are given in Sect. 5.

2 Related Work

For the detection of credit card frauds, various feature engineering strategies mostly based on an aggregation of transactions have been proposed in the literature. Also, Gradient Boosting Tree (GBT) has been investigated and compared with the other state-of-the-art algorithms.

Feature Engineering. To improve the performance of credit card fraud detection, Jha et al. [7] proposed a transaction aggregation strategy to derive certain aggregated features such as the total amount or the number of transactions initiated on the same day with the same merchant, currency or country for capturing the buying behaviors of customers. The features are created on real data of credit card transactions. In the study of Whitrow et al. [14], using a transaction aggregation process resulting in more robust to the problem of population drift, several features are derived at different aggregation periods. Afterwards, for the detection of credit card frauds, different algorithms are performed on transactional features including the derived ones. Also, a novel approach is proposed in this work [12] for the generation of a cardholder's profile by extracting the aggregated daily amounts of the cardholder's transactions. Comparison of the decision tree, artificial neural network, and logistic regression approaches for detecting credit card fraud transaction is studied by [13] on the combination of the raw features and eight aggregated features such as the total and average amount, the number of transactions and the failure number of transactions in the same day and the last five days, showing that neural network and logistic regression models over-perform decision tree. Bahnsen et al. [2,3] presented a transaction strategy for the capturing of the patterns for the periodic behavior of a transaction based on von Mises distribution and spending behavior belonging to a customer by exploiting the previous transactions of the customer. Also, it is revealed that spending patterns and time features improve the performance of the detection of fraudulent transactions. Moreover, Lim et al. [10] proposed a novel transaction-level based aggregation strategy using conditional weighted aggregation on the previous transactions. In this work, two weighting functions using the number of transactions and time are performed to assign more weights to the transactions.

In addition, the contribution of the aggregation on the detection of credit card frauds is investigated using various time-series based algorithms. Jurgovsky et al. [8] have used Long Short-Term Memory (LSTM) on the aggregated features extracted from the cardholder-present transactions. In another time-series based approach [11], Hidden Markov Model (HMM) is employed to extract descriptive aggregated features from a transaction sequence of a customer. For each transaction in the sequence, a likelihood value based on the previous transaction is computed, and then, this likelihood values are used as additional features to generate a Random Forest model.

Gradient Boosting Trees. Fang et al. proposed a Light Gradient Boosting Machine (LightGBM) for detecting credit card frauds [6]. In the work of [5], for the detection of credit card frauds, the boosting algorithms, Adaboost, Gradient Boost, and eXtreme Gradient Boosting (XGBoost) were implemented to evaluate the detection performances. It was found out that the XGBoost method outperforms the other algorithms.

3 Proposed Fraud Detection System

The proposed system for the detection of credit card frauds is composed of two main modules; offline training and real-time detection as shown in Fig. 1. There is a feature engineering step common in both of the components. The offline training phase covers the automatic generation of the training set to be used in the generation of a model for fraud detection. In the real-time detection phase, the feature vector is extracted from streaming transaction to decide whether it is fraudulent or not.

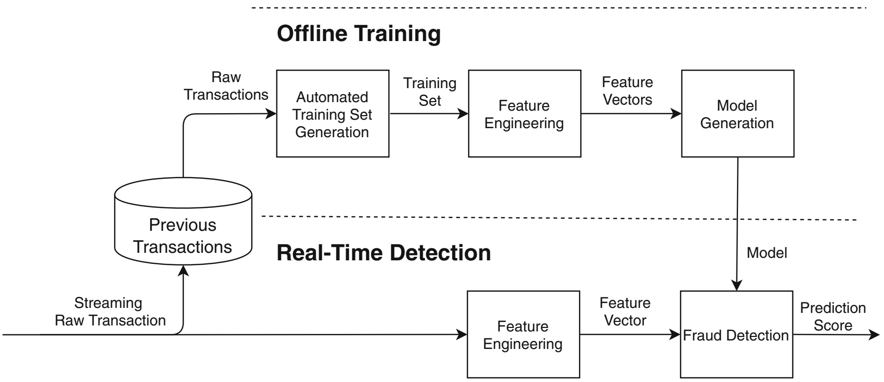

Fig. 1. The flowchart of the proposed system with offline training and credit card fraud detection in real-time.

3.1 Feature Engineering

A feature engineering process shown in Fig. 2 is required to improve the performance of model under the imbalanced class distributions, the noisy labels and features, skewness of the data, and overlapping transactions existing in both of the classes. Making combination of the different types of features, selection among the distinctive features, and generation of the additional hand-crafted features are used in feature engineering phase on the model generation component of our time-constrained fraud detection system.

Before the combination, the encoding of categorical features, generation of aggregated features, and extraction of word representations for the textual data are performed in the offline training and real-time detection stages. The categorical features are encoded in which a numeric value is set to a distinct categorical value. In addition, the vectors from the word embedding model of merchant names are included to compose the training instance for offline training component and to query the model on real time detection component.

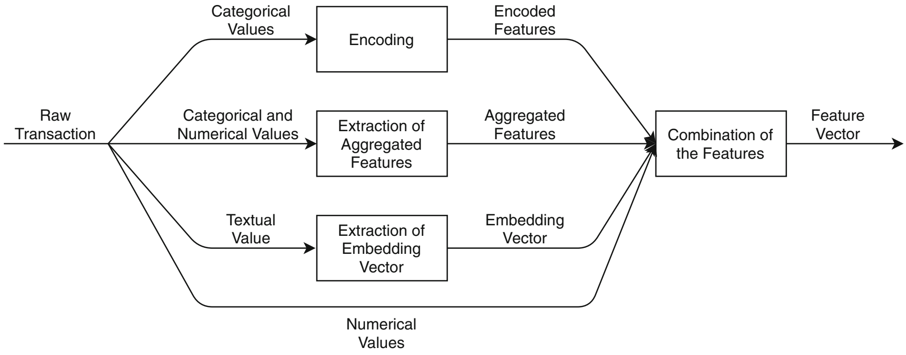

Fig. 2. The overview of the feature engineering process employed in the initial phases of the training and detection modules.

3.1.1 Encoding of Categorical Attributes

A few algorithms have been proposed to handle the data with categorical features. Therefore, an encoding process is required to transform them into numerical features. The values of some categorical attributes belonging to the transactions such as merchant category code, country code, etc. are directly encoded into a set of discrete integer values 0 through k where k is the number of unique categorical values.

3.1.2 Extraction of Aggregated Features

The properties of a credit card transaction do not represent the customer's financial behavior. In order to capture purchasing patterns to improve the model performance, an aggregation strategy is applied on raw transactional features such as currency, amount, merchant category. Utilizing the previous transactions of a credit card in a predefined time window to project the account's e-marketing behavior is the key point of our feature aggregation approach. Seeking the transaction history of a card in a time frame is needed because the recent transactions include much recent information about the customer. The day before the current day is decided as fixed time range. Various aggregated features per card, like the total amount of previous transactions, the total number of transactions which have the same merchant category code, total count of the same currency and country, etc., are computed in real-time from the transactions performed in the day before the current day.

3.1.3 Training a Word Embedding Model

For the use of the typical numeric, hand-crafted numeric, categorical, and textual attributes together, a set of a newly composed feature vectors are constructed to train a GBT model. Nevertheless, the attribute corresponding to merchant name is the textual one which have millions of different values including meaningless words and some phrases which have different number of characters. Thus, the same encoding technique used for the categorical attributes may not be useful on merchant names. Also, the list of the distinct names is being grown due to names of new merchants and mistyped names of the existing merchants, directly causing the out-of-vocabulary (OOV) problem.

In this study, due to the OOV problem and strict time constraint for fraud detection per transaction, we require a fast solution to convert such information into the numeric form to be used in the corresponding feature vector. Thus, to overcome these problems, fastText library that provides a character-level word embedding approach is exploited for the extraction of embedding vectors from the names. Among the last year's transactions, trigram character sequences of all the unique names of the merchants in which at least one transaction and contain only alphabetical characters, are considered to include the model. We train an embedding model on a corpus with millions of these filtered unique names. The model provides a 16-D embedding vector generated for each merchant name. The vector for the name of an incoming transaction is basically computed by summing all vectors extracted for each trigram sequence in the name, and included into the other features. Due to the restriction for the prediction time, retraining of the fastText model on the merchant names, including the new ones, is not applicable.

It is required to estimate how the embedding vector should be efficiently utilized. Combining the entire vector with the other features may make the name attribute dominate the other transactional features. To select more distinctive features and also reduce the dimensionality, Principled Component Analysis (PCA) is performed on the vectors of the names in training set like in Fig. 3. This technique removes the correlation and preserves the orthogonality between the word representations. Using PCA, the 16-D vectors are projected onto 4-D subspace determined by experiments. It is observed that the PCA based reduction prevents the dominancy of the vector on the other transactional features, enhances the fraud detection performance, and decreases the prediction time.

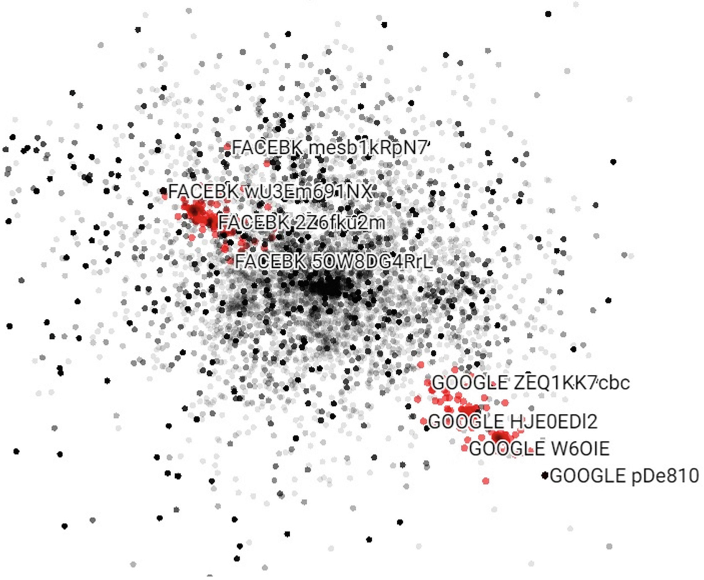

Fig. 3. A set of embedding vectors projected onto 2-D space generated on a set of merchant names in the training set.

3.2 Training of Credit Card Fraud Detection Model

In this study, the training set is constructed within several stages to create a data set in which classifiers can have stability against class skewness problems. It is desired to equalize fraudulent and legitimate credit card counts and to make sure that the transactions of each credit card covers the equal time range.

We identify credit cards as a fraudulent card if it has at least one fraud transaction in a given time range. If the credit card has no fraud transaction, then we denote it as a legitimate card. Instead of equalizing fraud and non-fraud transaction count, such as undersampling, we take the number of training instances the same for fraud and non-fraud ones. Although there is still massive class imbalance; however, we consider detecting fraud credit cards instead of the transaction more important.

The training set is constructed in a period of T, with m number of consecutive months. The last C number of months of this period is defined as the card selection zone, and the beginning B months denotes the transaction buffer zone. In Fig. 4, m is chosen as 10 to define the number of months in the period T. In the first place, containing the same count of the fraudulent credit cards, K, legitimate credit cards are sampled for each t_i month in the C card selection zone. This ensures that the dataset contains the same number of training instances for fraud and legitimate credit card transactions. Following the card selection process, all transactions of the selected cards in t_i, are obtained from the range of t_{i-n}, where n is the sliding window parameter. This operation needs reservation

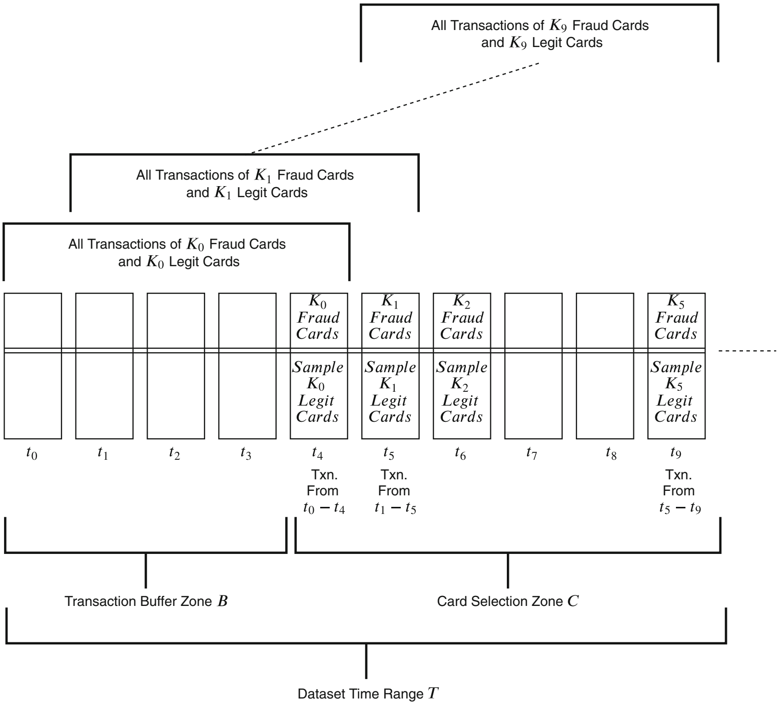

Fig. 4. The illustration of the automatic training set generation using transaction (txn) between T_i and T_{i+4}, where T denotes the month. The sliding window n parameter is chosen as 4 for this figure.

of the first n months only to fetch transactions of the t_{i+n} cards. As a result, the training set makes each card to have transactions from $(\text{n}+1)$ months.

In addition, after a new training set is constructed, a PCA model for the embedding vectors of the merchant names is also updated using the vector representations of the merchant names. However, the same fastText model is utilized since the update of the word-embedding model like the fraud detection model may not enhance the vector representation.

4 Experiments

4.1 Experimental Setup

The experiments were performed using two training sets and test sets incorporating credit card transactions in a private bank. Training sets are formed using the approach described in Table 4. These two training sets contain transactions from 2018 October to 2019 July (Oct.'18–Jul.'19) and 2018 November to 2019 August (Nov.'18–Aug.'19), respectively. For the test sets, we use all CNP transactions in the next month for each training set that are August (Aug.'19) and

September (Sep.'19). Also, 30% of each training set has been utilized as a validation set, to be used in the training of the detection model. More details about the datasets can be found in Table 1.

The proposed system for credit card fraud detection was developed in Python 3.6.8. For the GBT model generation, XGBoost library 0.71 presenting an efficient Gradient Boosted Tree (GBT) implementation is used on these datasets. Also, Redis, an in-memory data structure store based on a server-client model with TCP sockets, is utilized for the real-time transaction aggregation.

4.2 Evaluation Metrics

Experiments are evaluated by the following metrics; the False-Positive Rate (FPR), recall, precision, and Area Under Curve (AUC). We also calculated the Equal Error Rate (EER), which determines the threshold values for its false acceptance rate and its false rejection rate when the rates are equal. Using the determined threshold values from EER calculation, we evaluated the performance of the credit card fraud detection model. Moreover, we experimentally found that 0.3 is decided as a fixed FPR to obtain the optimum threshold value.

Table 1. Overview of the training and test sets (Transactions = Txn).

	Training set		Test set	
Overview	Oct.'18–Jul.'19	Nov.'18–Aug.'19	Aug.'19	Sep.'19
# of txn.	1,3 M	1,4 M	20,2 M	21,3 M
# of cards	31,158	49,002	3 M	3,1 M
# of fraud txn.	61,130	52,610	6,767	7,045
# of fraud cards	15,579	24,501	3,626	3,997

4.3 Experiments of Feature Engineering

In the experiments regarding feature engineering process, only encoded features, encoded with aggregated ones, and all of them including word embedding features have been used to train a GBT model. In Table 2, the evaluation is shown on the test set Aug.'19 composed of all the CNP transactions on August, 2019. Also, on the test set, Sep.'19 composed of all the CNP transactions in September, 2019, the fraud detection results of these three models trained using three different feature sets are shown in the Table 3.

4.4 Experiments of Automated Training Set Generation

Two different datasets are formed by sliding window approach using transactions in Oct.'18–Jul.'19 & Nov.'18–Aug.'19. For each set, a brand new GBT model

Table 2. Success measurement results of our fraud detection system trained on Oct.’18–Jul.’19 training set and tested on Aug.’19 test dataset.

		Fixed FPR		EER			
Features	AUC	FPR	Recall	FPR	Recall	Error	Threshold
Encoded	0.947	0.3	0.962	0.120	0.879	12.08	0.064
Agg. + Encoded	0.951	0.3	0.962	0.115	0.883	11.68	0.056
Emb. + Agg. + Encoded	0.958	0.3	0.974	0.110	0.889	11.02	0.055

Table 3. Success measurement results of the fraud detection models which are trained on Nov.’18–Aug.’19 training sets and tested on Sep.’19 test set.

		Fixed FPR		EER			
Features	AUC	FPR	Recall	FPR	Recall	Error	Threshold
Encoded	0.960	0.3	0.984	0.104	0.895	10.48	0.018
Agg. + Encoded	0.964	0.3	0.960	0.099	0.900	9.90	0.017
Emb. + Agg. + Encoded	0.968	0.3	0.989	0.091	0.908	9.15	0.017

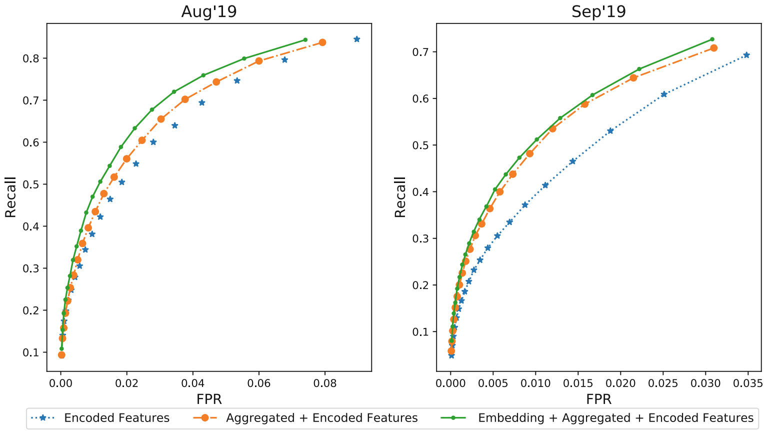

Fig. 5. Recall-FPR curve of the detection models trained on these features for test set composed of all CNP transactions in August (figure on the left), and test set composed of all CNP transactions in September (figure on the right)

Table 4. Results of the models trained on the sets generated by sliding window

		Fixed FPR		EER			
Training sets	AUC	FPR	Recall	FPR	Recall	Error	Threshold
Oct.’18–Jul.’19	0.968	0.3	0.989	0.091	0.908	14.16	0.038
Nov.’18–Aug.’19	0.940	0.3	0.960	0.141	0.858	9.15	0.017

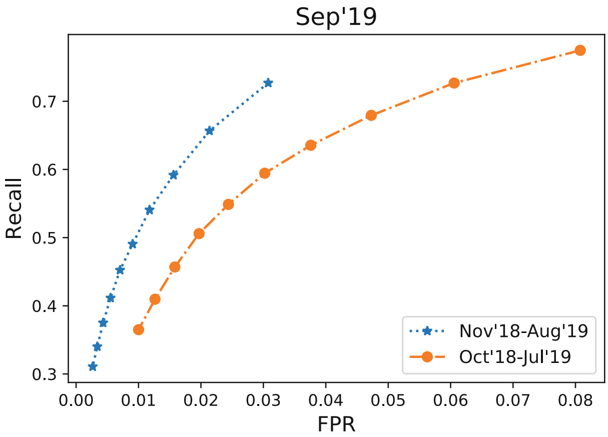

Fig. 6. The contribution of the sliding window based approach being employed for training set generation.

is trained and their detection performance are evaluated on the same test set Sep.'19 (Fig. 5).

Figure 6 and Table 4 reveals that, sliding the training set enhances the fraud detection performance in terms of AUC by 0.028%, and in fixed 0.3 FPR, Recall is increased by 0.029%. Also, we realized that joining recent transactions and decreasing more distant transactions to the training set increases the success of our credit card fraud detection model.

5 Conclusion

In this work, for the real-time detection of credit card fraud, a new approach is proposed for training dataset construction to make usable and valuable of different types of attributes of a transaction by combining numerical, hand-crafted numerical, categorical and textual features. Also, a character-level word embedding method is utilized to generate embedding vector for each merchant name. Also, there is a major problem affecting the durability of the model. Therefore, automatic generation of a transaction dataset has been developed to extract feature vectors used as training set to generate a Gradient Boosting Tree (GBT) model for the detection of the fraud transactions. In the future work, new types of aggregated features will be created and an ensemble method will be developed by combining of a number of models trained on the transactions of long- and short-term periods.

References

1. Fifth report on card fraud (2018). https://www.ecb.europa.eu/pub/cardfraud/html/ecb.cardfraudreport201809.en.html. Accessed 03 Feb 2020

2. Bahnsen, A.C., Aouada, D., Stojanovic, A., Ottersten, B.: Detecting credit card fraud using periodic features. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp. 208–213. IEEE (2015)
3. Bahnsen, A.C., Aouada, D., Stojanovic, A., Ottersten, B.: Feature engineering strategies for credit card fraud detection. Expert Syst. Appl. **51**, 134–142 (2016)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
5. Divakar, K., Chitharanjan, K.: Performance evaluation of credit card fraud transactions using boosting algorithms. Int. J. Electron. Commun. Comput. Eng. IJECCE **10**(6), 262–270 (2019)
6. Fang, Y., Zhang, Y., Huang, C.: Credit card fraud detection based on machine learning. Comput. Mater. Continua CMC **61**(1), 185–195 (2019)
7. Jha, S., Guillen, M., Westland, J.C.: Employing transaction aggregation strategy to detect credit card fraud. Expert Syst. Appl. **39**(16), 12650–12657 (2012)
8. Jurgovsky, J., et al.: Sequence classification for credit-card fraud detection. Expert Syst. Appl. **100**, 234–245 (2018)
9. Korman, B.R., Bergman, D.J.: Multi-transactional architecture. US Patent 6,308,887, 30 Oct 2001
10. Lim, W.-Y., Sachan, A., Thing, V.: Conditional weighted transaction aggregation for credit card fraud detection. In: Peterson, G., Shenoi, S. (eds.) DigitalForensics 2014. IAICT, vol. 433, pp. 3–16. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44952-3_1
11. Lucas, Y., et al.: Multiple perspectives HMM-based feature engineering for credit card fraud detection. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 1359–1361 (2019)
12. Seyedhossein, L., Hashemi, M.R.: Mining information from credit card time series for timelier fraud detection. In: 2010 5th International Symposium on Telecommunications, pp. 619–624. IEEE (2010)
13. Shen, A., Tong, R., Deng, Y.: Application of classification models on credit card fraud detection. In: 2007 International Conference on Service Systems and Service Management, pp. 1–4. IEEE (2007)
14. Whitrow, C., Hand, D.J., Juszczak, P., Weston, D., Adams, N.M.: Transaction aggregation as a strategy for credit card fraud detection. Data Min. Knowl. Disc. **18**(1), 30–55 (2009)
15. Yen, S.J., Lee, Y.S.: Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In: Huang, D.S., Li, K., Irwin, G.W. (eds.) Intelligent Control and Automation. LNCIS, vol. 344, pp. 731–740. Springer, Heidelberg (2006). https://doi.org/10.1007/978-3-540-37256-1_89