A Graph-based Semi-supervised Fraud Detection Framework

Rongrong Jing[1,2], Xiaolong Zheng[1,2,3], Hu Tian[1,2], Xingwei Zhang[1,2] , Weiyun Chen[4], Dash Desheng Wu[2], Daniel Dajun Zeng[1,2,3]
[1]The State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3]Shenzhen Artificial Intelligence and Data Science Institute (Longhua), Shenzhen 518129, China
[4]School of Management, Huazhong University of Science & Technology, Wuhan 430074, China
{ jingrongrong2019, xiaolong.zheng, tianhu2018, zhangxingwei2019, dajun.zeng } @ia.ac.cn,
chenweiyun@mail.hust.edu.cn, dwu@ucas.ac.cn

Abstract—**Credit card payment has become one of the most commonly used consumption methods in modern society, yet risks of fraud transactions using credit cards also increased. Numerous methods have been proposed for credit card fraud detection during past decades. However, most of the existing frameworks mainly focus on directly processing structured data. While they lack inner relations between features of raw descriptions for credit owners, this could lead to information deficiency. Therefore, we proposed a graph-based semi-supervised fraud detection framework. In this work, the structured dataset is translated to graph format through the sample similarity in order to improve the effect of label propagation on the graph. We further adopt the GraphSAGE algorithm which has been demonstrated to show excellent performance on node classification tasks. Experimental results on the real-world dataset show that our graph-based model can outperform state-of-the-art baselines. We argue that our model could be extended to other classification tasks using structured data.**

Keywords—Credit card fraud Detection, Graph neural network, graph modeling, node classification

2019 4th IEEE International Conference on Cybernetics (Cybconf) | 978-1-7281-0030-2/19/$31.00 ©2019 IEEE | DOI: 10.1109/CYBCONF47073.2019.9436573

I. Introduction

In the past few years, the rapid development of mobile payment has made credit card payment more and more popular. But credit card fraud is also increasing in frequency, which makes global consumers and banking institutions suffer billions of dollars in losses every year [1]. Credit card fraud refers to the activities of non-cardholders to obtain illegal economic benefits in any way for the purpose of deception without the knowledge of the cardholder and the issuing bank. Therefore, for the financial system, it is very necessary to develop credit card fraud detection technology. There are various techniques available for credit card fraud detection such as Logistic Regression (LR), Decision Tree, rule based mining, fuzzy clustering approach, Support Vector Machine (SVM), Bayesian Network, Artificial Neural Network (ANN), Hidden Markov Model (HMM) or hybrid approach of these methods. [2].

Credit card transaction records are usually included identification information, location, phone number, income, and much additional details. Most of these data sets are structured data, where data is created using a pre-defined schema and is organized in a tabular format. Since graph may carry out with more information and depict complex relationships, translating structured data to graph format is a worthwhile job. Meanwhile, using various machine learning graph-based algorithms, we can mine a lot of useful hidden information from the graph.

Research on credit card fraud detection from the perspective of the graphic has attracted extensive attention. Belle et al. [12] introduced two representational learners: one is GraphSAGE, which is inherently inductive, and the other is a transductive representational learner based on Node2vec. How to build the graph is a challenge. Some researchers [9] [10] [12] [13] created graph through making advantage of transaction relations between holders and merchants for credit card fraud detection. Sadowksi and Rathle [8] discussed the graph databases that offer new method of uncovering fraud sophisticated rings. For example, two accounts which are nodes of graph can be connected because of sharing mutual e-mail address or home addresses. However, the number of publications available is limited. Meanwhile, most researchers introduced additional information of trading behaviors rather than the information inherent in the data itself.

To address these limitations, we modeled a correlation graph based on the sample similarity which is calculated by Pearson Product–moment Correlation Coefficient [15] in order to improve the effect of label propagation for node classification task. In this way, we turned the supervised learning to semi-supervised learning. Then we adopt an excellent and suitable model of node classification, GraphSAGE, to detect fraud risk. We compared our framework against the traditional approaches such as which are suitable for structured data. Experimental results demonstrate that our graph-based model can outperform the state-of-the-art methods on a real-world dataset.

The remainders of the paper are designed as follows. We introduce the dataset used in Section 2 and present the corresponding methods in the Section 3. Section 4 describes the dataset used and the demonstrates the main empirical results, including the description of the correlation graph and results of credit card fraud detection. In Section 5, we conclude the paper with the future work.

II. Related works

Card fraud detection has been a point of interest in the past. There are lots of problems such as feature selection, label imbalance and data missing that researchers have used a variety of methods to solve. Sánchez D et al. [5] applied Association rules to extract knowledge from the transactional credit card databases. R. Jing et al. [3] developed a spectral regularization algorithm and sampling approach to enhance the data quality to detect credit card fraud.

There are same patterns in the most fraudulent transactions generally and the transactions are classified as fraudulent by using any pattern recognition approaches such as SVM, ANN, Bayesian Network, KNN, HMM, Fuzzy Logic Based System and other approaches. To improve prediction accuracy, Shen et al. [4] used Logistic Regression, Decision Tree and Neural Network to analyze the credit card history business information. A similar study was conducted by Sahin and E.Duman [6], they compared the performance of Decision Tree and SVM in credit card fraud detection task. Abhinav Srivastava et al. [7] considered the operations processing sequence by HMM.

In recent years, some researchers are tackling this task from a network or graph perspective. Vlasselaer et al. [8] exploited a network of credit cardholders and merchants. They extracted features which combined both intrinsic and network-based attributes and then estimated LR, neural network and random forest method on these features. This study was followed by Lebichot et al. [10], they also built a network of credit card merchants and holders. But they adopted a semi-supervised graph-based classification with the Regularized Commute Time Kernel. Ramaki et al. [13] intended to provide an ontology graph model for credit card fraud detection on semantic connections between data stored for each transaction that a user basis fulfilled. Belle et al. [12] explored the relational and structural aspects of the transaction networks and utilized these in credit card fraud predictive models. Our framework utilized the intrinsic relationship of data to model graph rather than introducing a large amount of additional transaction information.

III. Methods

In this section, we demonstrate the graph modeling method using the structured credit card fraud dataset. Then we provide a brief introduction of GraphSAGE algorithm.

A. Graph modeling

The idea of graph modeling is that users connect to each other through feature similarity. A weighted graph between users and users is denoted as $G = \{U,\ F,\ E,\ W, L\}$. Here U is users vertex set. The node of user i is denoted as u_i. F is feature set. The features of u_i is denoted as $F_i = \{f_{i1}, f_{i2}, \dots, f_{in}\}$, where n is the amount of features of users. L is labels set. The label of user i is denoted as l_i. W is the set of edge weights. We use Pearson Product–moment Correlation Coefficient (PPCC) to compute the correlation coefficients between any two users. This correlation coefficient between -1 and 1 is used to find how strong a relationship is between data. The closer the absolute value of the correlation coefficient is to 1, the greater the correlation is. If the coefficient is negative, it means that their relationship has negative correlation. The particular weight $corr\_w_{ij}$ of user i and user j is calculated as follows:

$$corr\_w_{ij} = \frac{\sum_{i=1}^{n}(f_i-\bar{f_i})(f_j-\bar{f_j})}{\sqrt{\sum_{i=1}^{n}(f_i-\bar{f_i})^2}\sqrt{\sum_{i=1}^{n}(f_j-\bar{f_j})}^2}, \qquad (1)$$

We set a threshold value W_{limit} for every weight $corr\_w_{ij}$. Once $corr\_w_{ij} \geq W_{limit}$, there is a link named e_{ij} in graph G. Importantly, the linkages explicitly reflect the similarity between users. In this model, we not only convert structured data to graph which can be applied more abundant algorithms, we also add more additional correlation information which is beneficial to graph node classification model.

B. Graph Node Classification

In this article, we adopt a general framework for inductive node embedding, called GraphSAGE [14]. GraphSAGE, an inductive node embedding model, utilizes local node neighborhood characteristics. A function is learned to generate the new embeddings for invisible nodes. While GraphSAGE, designed for graphs with rich node attributes originally, can handle graphs without node features. Unlike transductive learning approaches as GCN, GraphSAGE leverages node properties to learn an embedded function that can generalize the unseen nodes. This approach is efficient to generate node embeddings for previously unseen data.

The key idea behind GraphSAGE is to learn the method of aggregating characteristic information from the local neighborhood of a node. The learning process of GraphSAGE is mainly divided into three steps including sampling neighborhood, aggregating feature information from neighbors, and predicting node labels by using the aggregated information.

For a given network $G = (V,\ E,\ X)$. The adjacent nodes of each node are sampled, and the aggregation function shown in Formula (1) is used to aggregate the feature set.

$$h_N^k = AGGREGATE_k(\{h_u^{k-1}, \forall u \in N(v)\}) \qquad (2)$$

where h_N^k is composed of the neighbor information of adjacent nodes and the characteristic information of adjacent nodes, and h_N^{k-1} denotes a sample of neighbor nodes.

Since the neighbors of vertices in the graph are naturally disordered, the constructed aggregate function should be symmetric (that is, the order of inputs is changed, and the output of the function remains the same), with high expressive power at the same time. Unlike GCN, GraphSAGE offers three aggregation functions: LSTM, pool, and mean aggregation function.

GraphSAGE assumes that nodes that reside in the same neighborhood should have similar embeddings. In our graph modeling method, we construct the graph by the similarity of the nodes. By this way, we can make use of this powerful graph algorithm and improve the effect of label propagation.

IV. Experiment

A. Dataset

In order to estimate the performance of our proposed graph-based semi-supervised framework, the dataset from the UC Irvine Machine Learning Repository (https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/) prepared by Prof. Hofmann was selected. The dataset consists of 1000 records and each record has 20 categorial or symbolic attributes. In this data set, each record represents a person receiving credit from a bank. Each record is classified as a good or bad credit risk based on the attributes. 30% of transactions are fraud in the dataset. Because of its complex system of categories and symbols, we select the most important features which are job, sex, age, saving accounts, housing, credit amount, checking account, duration and purpose. Then we encode categorical features as a one-hot numeric way.

We modeled a correlation graph of credit card fraud detection using the data sets mentioned above. The nodes are representative of clients corresponding to the 1000 entries in the original dataset. The edges present the similarity degree of the two clients. A visualized subgraph with 15 nodes is shown in Fig. 1. Every node has its label 'At risk' (Binary variable). We divided 60% nodes as a training set, 20% nodes as a valid set, and 20% as test set. To avoid randomness and enhance credibility, we divided the data sets randomly with fixed proportion in every experiment. Our goal is predict the labels of nodes in test set.

TABLE I. described the statistical information of the graph. The presentation of graph will vary with the threshold value W_{limit} . While W_{limit} is zero, the graph is fully connected without any filter. It can be found that the mean weight of all edges is up to 0.93. It reflects the high similarity of the samples which make it harder to distinguish fraud. While W_{limit} is increasing, the mean of edge weights, density and average clustering coefficient is decreasing. By setting the threshold, we can retain the most critical information to improve the effect of label propagation for the downstream task for node classification. As shown in Fig. 1, nodes with similar types are more likely to be connected and clustered together.

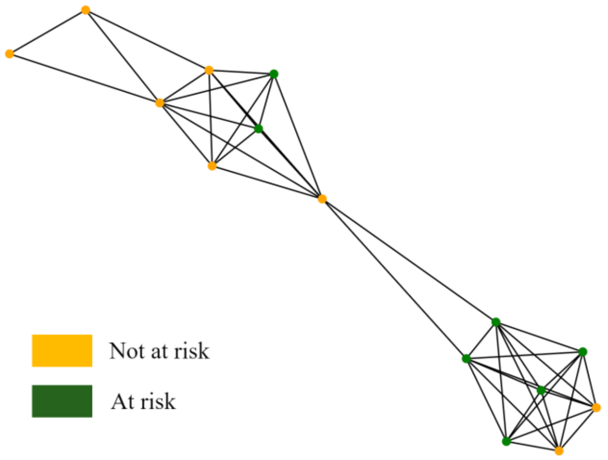

Fig. 1. Correlation Graph of credit card fraud detection

TABLE I. STATISTICAL INFORMATION OF THE GRAPH

Threshold Setting	**Number of Nodes**	**Number of Edges**	**Mean of edge weights**
$W_{limit} = 0.0$	1000	499500	0.9299
$W_{limit} = 0.5$	1000	498795	0.9305
$W_{limit} = 0.9$	1000	377866	0.9689
$W_{limit} = 0.95$	1000	288992	0.9816
Threshold Setting	**Density**	**Average degree**	**Average clustering coefficient**
$W_{limit} = 0.0$	1.0	749.25	1.0
$W_{limit} = 0.5$	0.9986	748.55	0.9987
$W_{limit} = 0.9$	0.7565	627.62	0.8874
$W_{limit} = 0.95$	0.5766	537.74	0.8371

B. Experiment setting

We designed two experiments to evaluate our model. In the first experiment, in order to prove whether the correlation intrinsic relationship of sample similarity can help the prediction. We applied our framework on three different graphs where there is no difference except the construction method. The first graph is a fully-connected graph with edges weighted $corr\_w_{ij}$. The second graph is constructed by our filter method except that all weights are 0. The last one is our graph model that filtered by $corr\_w_{ij}$. In all subsequent experiments, the graphSAGE is used to identify fraud here. We set learning rate 0.01 and adopt GCN and MEAN as the aggregator of two neural network layers.

In the second experiment, we compared the following baselines which are fit to tackle structured data with our framework using the same data set. The main goal of our experiment is to prove that translating the structured data to the format of graph by adding additional intrinsic information in our framework can help solving classification task.

We selected Logistic Regression (LR) [16], Support Vector Machine (SVM) [17], and XGBoost [11] as baselines. These baselines are the typical method in the areas of statistics, traditional machine learning, and decision tree accordingly. The iterations in these three baselines are all set to 1000. LR is a kind of classification statistical model, which uses maximum likelihood estimation to estimate parameters. LR can describe the data and model the probability of a certain event. We selected L2 regularization and newton-cg solver whose performance is better. SVM is a supervised learning model that aims to find the best hyperplane which has the maximum margin to distinguish the data points. SVM can be applied to solve both classification and regression problems. We selected the RBF kernel function because of the undivided linear samples and low complexity of the RBF function. XGBoost is a highly efficient approach which uses a gradient boosting framework to enhance the weak learner principle to be a strong classifier. In this task, the max depth of tree, learning rate, and value of estimators is set to 3, 0.1, and 100.

C. Result

In the first experiment, as described in the previous section, we applied our framework on three different graphs.

The accuracy of training is 0.3 or 0.7 in Fig. 2 which equals the proportion of labels. At meanwhile, the precision and recall of 'Risk' category is both 0 in TABLE II. As for the first graph, the classifier can't classify nodes and regard all the nodes as the same category. There are too many edges that the influence of nodes is dispersed and weakened. Although every edge has a different weight, labels are hard to spread because there are too many nodes to choose.

TABLE II. THE PERFORMANCE COMPARISON OF VARIOUS METHODS.

Graph	**At Risk**	**Precision**	**Recall**	**F1-score**	**Accuracy**
fully-connected + $corr\_w_{ij}$	0	0.68	**1.00**	0.81	0.68
	1	0.00	0.00	0.00	
Edge filter+ $w_{ij} = 0$	0	**0.73**	0.91	0.81	0.71
	1	0.56	**0.25**	0.34	
Edge filter+ $corr\_w_{ij}$	0	**0.73**	0.97	**0.84**	**0.73**
	1	**0.75**	0.20	**0.31**	

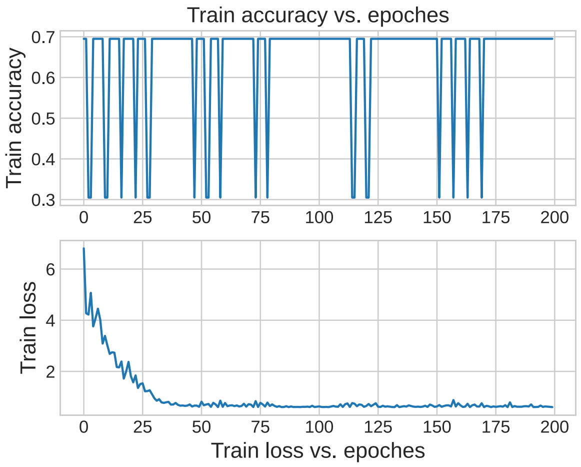

Fig. 2. Training accuracy and training loss of full-connected graph with edges weighted $corr\_w_{ij}$

Although the final training result is stable in Fig. 3, the oscillation in the middle process is severe. Surprisingly, this approach has also yielded good results. However, this method is inferior to our graph modeling method on the whole. It is found that the structure of the graph is the main influence factor on the result. It's worth noting that this situation may due to the close weights in this dataset where the weights are all around 0.9.

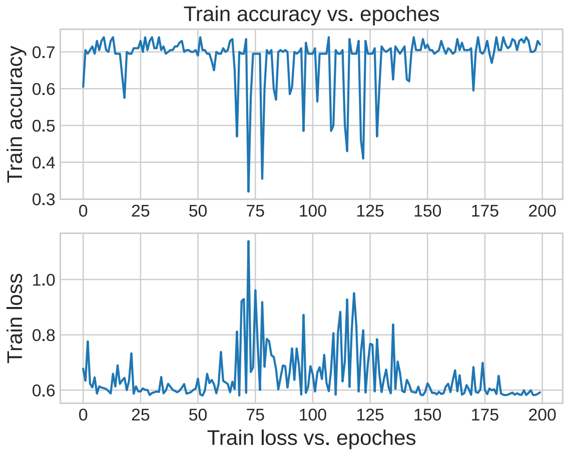

Fig. 3. Training accuracy and training loss of f graph constructed by our fileter method except that all weights are 0

By comparing the training accuracy as well as training loss of three graphs, it is so obvious that the our graph model can converge very quickly in Fig. 4. As meanwhile, the classifier can distinguish the two categories well with accuracy over 0.7. It proved that our graph modeling method by correlation intrinsic relationship of sample similarity can help downstream classification tasks. It provides an effective method to model graph based on structured data.

In the second experiment, as described in the previous subsection A, we compared these following baselines which are fit to tackle structured data with our framework using the same data set. As shown in Fig. 5 and TABLE III. , SVM can't classify node and regard all the nodes as the same category. The precision and recall of 'Risk' category are both 0 in TABLE II. Although SVM is a powerful tool for classification task, this tool is not suitable for solving this problem. The mean similarity of samples is up to 0.93 which we mentioned above, so it is hard to divide.

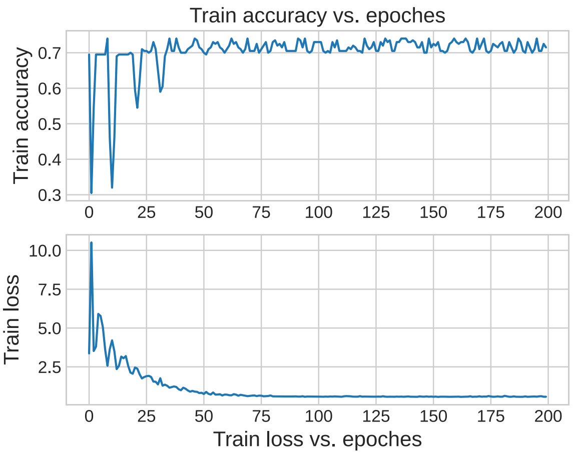

Fig. 4. Training accuracy and training loss of our graph model that filtered by $corr\_w_{ij}$

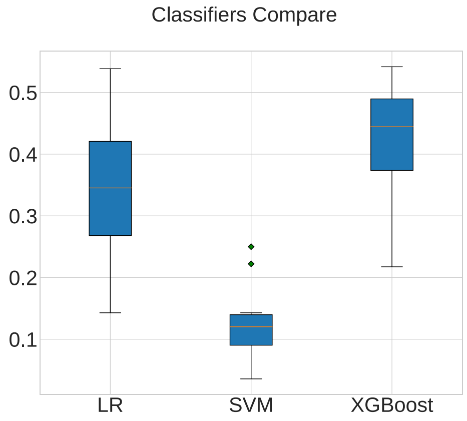

Fig. 5. Cross valid scores of baselines

TABLE III. THE PERFORMANCE COMPARISON OF VARIOUS METHODS.

Classifier	At Risk	Precision	Recall	F1-score	Accuracy
LR	0	0.72	0.87	0.79	0.69
	1	0.55	0.32	0.40	
SVM	0	0.68	1.00	0.81	0.68
	1	0.00	0.00	0.00	
XGBoost	0	**0.76**	0.81	0.78	0.66
	1	0.55	0.48	0.52	
Ours (Corraletion weight)	0	0.73	0.97	0.84	**0.73**
	1	**0.75**	0.20	0.31	

The performance of classifier LR, XGBoost and our framework all exceed 0.65. However, the results of the two categories are different. The precision and recall of 'At risk' category are both below requirements especially in the credit card fraud detection which need more frauds detected. In particular, although the precision of label 'Not at risk' in

Xgboost is up to 0.76, but it can only detect halt of frauds. By contrast, our framework performs well in both categories with best accuracy. It indicates that our model where we translate structured data to graph format with additional information can figure out this classification task effectively.

V. Conclusions

In this paper, we have proposed a novel graph-based framework to detect credit card fraud. Graph can carry more information and depict complex relationships. We translated original structured data to graph format through intrinsic information of features. The method of graph modeling can help improve the performance of GraphSAGE algorithm on node classification task. We turned a normal classification task to a semi-supervised node classification task which can be solved with more efficient graph algorithms. Experimental results show excellent performance on the real-world fraud card dataset. Our framework provides a novel perspective that structured data could be processed using the rapidly developed graph structures.

Acknowledgement

This work is supported by the Natural Science Foundation of China under Grant No. 71621002, 71602184 and 71472175, the Ministry of Health of China under Grant No. 2017YFC1200302 and 2017ZX10303401-002, and the National Key Research and Development Program of China under Grant No. 2020AAA0108401, 2019QY(Y)0101 and 2016QY02D0305.

References

[1] Delamaire, L., Abdou, H., & Pointon, J. (2009). Credit card fraud and detection techniques: a review. Banks and Bank systems, 4(2), 57-68.

[2] Jain, Y., NamrataTiwari, S., & Jain, S. (2019). A comparative analysis of various credit card fraud detection techniques. Int J Recent Technol Eng, 7(5S2), 402-407.

[3] R. Jing et al., "Improving the Data Quality for Credit Card Fraud Detection," 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), Arlington, VA, 2020, pp. 1-6, doi: 10.1109/ISI49825.2020.9280510.

[4] A. Shen, R. Tong and Y. Deng, "Application of Classification Models on Credit Card Fraud Detection," 2007 International Conference on Service Systems and Service Management, Chengdu, 2007, pp. 1-4, doi: 10.1109/ICSSSM.2007.4280163.

[5] Sánchez, D., Vila, M. A., Cerda, L., & Serrano, J. M. (2009). Association rules applied to credit card fraud detection. Expert systems with applications, 36(2), 3630-3640.

[6] Şahin, Y. G., & Duman, E. (2011). Detecting credit card fraud by decision trees and support vector machines.

[7] A. Srivastava, A. Kundu, S. Sural and A. Majumdar, "Credit Card Fraud Detection Using Hidden Markov Model," in IEEE Transactions on Dependable and Secure Computing, vol. 5, no. 1, pp. 37-48, Jan.-March 2008, doi: 10.1109/TDSC.2007.70228.

[8] Sadowski, G., & Rathle, P. (2014). Fraud detection: Discovering connections with graph databases. White Paper-Neo Technology-Graphs are Everywhere, 13.

[9] Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M., & Baesens, B. (2015). APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions. Decision Support Systems, 75, 38-48.

[10] Lebichot, B., Braun, F., Caelen, O., & Saerens, M. (2016, November). A graph-based, semi-supervised, credit card fraud detection system. In International Workshop on Complex Networks and their Applications (pp. 721-733). Springer, Cham.

[11] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016: 785-794.

[12] Van Belle, R., Mitrović, S., & De Weerdt, J. (2019, September). Representation learning in graphs for credit card fraud detection. In Workshop on Mining Data for Financial Applications (pp. 32-46). Springer, Cham.

[13] Ramaki, A. A., Asgari, R., & Atani, R. E. (2012). Credit card fraud detection based on ontology graph. International Journal of Security, Privacy and Trust Management (IJSPTM), 1(5), 1-12.

[14] Hamilton, W. L., Ying, R., & Leskovec, J. (2017, December). Inductive representation learning on large graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems (pp. 1025-1035).

[15] Weisstein, E. W. (2006). Correlation coefficient. https://mathworld. wolfram. com/.

[16] Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., & Klein, M. (2002). Logistic regression. New York: Springer-Verlag.

[17] Joachims, T. (1998). Making large-scale SVM learning practical (No. 1998, 28). Technical report.