Oversampling Techniques for Detecting Bitcoin Illegal Transactions

Jungsu Han[1], Jongsoo Woo[2], and Jame Won-Ki Hong[1]

[1]Department of Computer Science and Engineering, POSTECH, Korea.
{saw1515,jwkhong}@postech.ac.kr
[2]Center for Crypto Blockchain Research, POSTECH, Korea.
woojs@postech.ac.kr

***Abstract*— Bitcoin users are guaranteed to be anonymous, increasing the number of cryptocurrency trading related to crimes and fraudulent activities. While most studies about detecting illegal transactions try to distinguish trading patterns and classify them from legitimate ones, classification performance is poor since the class distributions of transaction data are highly imbalanced. In general, the Synthetic Minority Over-sampling TEchnique (SMOTE) is used to deal with class-imbalanced data, but SMOTE has a problem that it does not fully represent the diversity of the data. In this paper, we introduce another oversampling technique using Generative Adversarial Networks (GAN) to generate artificial training data for classification model. In order to verify similarity between artificial data and the actual one, oversampled dataset is evaluated with a classification model using XGBoost algorithm. We show classification performance is improved on average with synthetic data generated by both SMOTE and well-designed GAN model.**

***Index Terms*— Blockchain, Imbalanced data, Oversampling, Illegal detection, Classification, SMOTE, GAN**

I. INTRODUCTION

Cryptocurrency is a digital asset traded on blockchain networks using encrypted public keys, easily proving ownership with hash functions. Blockchain, the core technology of cryptocurrencies, uses distributed ledgers to disclose all transactions and the account information between individuals. However, most cryptocurrencies on public blockchains are difficult to identify users to ensure the anonymity of whole system [1]. Identifying individuals in blockchain network is possible only with the address of the account, which anyone can create and own freely. For this reason, bitcoin guarantees strong privacy. Therefore, someone can trade bitcoin with malicious purpose and there could be transactions associated with illegal activity. According to a survey conducted by Sean, around $35 billion illegal activities occur annually in the U.S. and Europe for weapons, drugs, and money laundering [2]. Therefore, detection and classification of illegal transactions from legitimate ones are essential for the blockchain technology to develop into a next-generation financial system.

A classification model can predict a class for new input data by analyzing patterns in existing data sets. Generally, classification models are developed through machine learning, where the training datasets are ideally well-balanced. But for most of the data actually collected, the classes are imbalanced. Imbalanced data typically refers to a training dataset where the number of observations per class is not equally distributed. If there are a number of data/observations for one class, it is called as the majority class, otherwise as minority class. In the case of bitcoin transaction, the dataset is also imbalanced due to the difficulty of securing illegal transactions. Since there is no explicit feature in the block record to express irregular transaction pattern, we collect transaction history with crawling system in Darknet [3]. A common way to collect illegal transaction data is to collect the hash value of transactions made on unofficial and prohibited trading sites (e.g., *Silkroad* in Darknet) [4]. A Darknet is an overlay network within the Internet that can only be accessed with specific software, configurations, or authorization. In 2017, bitcoin worth $770 million were traded through Darknet and most of them are linked to banned drugs and gun trades. Despite numerous criminal transactions, we cannot collect all illegal transactions because any Darknet operator including *Silkroad* does not provide trading log. Thus, the number of illegal transactions is less than the number of legitimate transactions traded on official exchanges.

Synthetic Minority Over-sampling TEchnique(SMOTE), a common resampling method, is used to address the imbalanced dataset which is degrading model performance [5]. But SMOTE is not suitable for high-dimensional data due to the problem of overfitting. Since SMOTE generates data regardless of the data distance, it could replicate the same instances over and over. Accordingly, we also propose Generative Adversarial Network (GAN) as a solution to solve the imbalanced data problem used in detecting bitcoin illegal transactions [6].

The rest of the paper is organized as follows. In Section II we briefly explain Bitcoin transaction analysis and theoretical background about oversampling and GAN. Then, in Section III we present experiments based on GAN models and SMOTE. We point out the limitations of dataset and models and include descriptions of additional experiments that have improved them. In Section IV, we summarize the experiment results and discuss possible future research.

II. RELATED WORK

A. Bitcoin transaction analysis

[7, 8] proposed a classification model to classify transactions related to crime and fraudulent activities. They

used clustering algorithm or machine learning to determine the characteristics of illegal transactions and identify users performing similar actions. [9] extracted 12 feature coefficients for anomaly detection in bitcoin network using an unsupervised learning method. Previous studies have only focused on classification method or model architecture, not on dataset. The main purpose of this study is to investigate the improvements in classification model performance while solving the imbalance in dataset.

B. Oversampling

Haixiang explained that Random Oversampling (ROS), Adaptive Synthetic Sampling (ADASYN) and SMOTE are used as representative oversampling techniques [10]. ROS is a method of randomly replicating minority class data samples and SMOTE selects K-nearest neighborhood between data samples to create new data samples following those points. ADASYN is an advanced technique in SMOTE that uses density distribution to create more realistic samples to prevent the distorted distribution of data created by SMOTE.

We note there are several attempts [11–15] using GAN-based model to oversampling for class-imbalanced dataset classification. GAN generates new data instances based on the existing data, and is typically made up of two deep networks: *generator* and *discriminator*. The purpose of the *generator* is to give random noise to the artificial data so that the data is similar to the actual data distribution, and the *discriminator* is to distinguish between the data produced by the *generator* and the actual data. The two models will compete in learning to generate more realistic data. conditional-GAN(CGAN) is an extended model of GAN, adding additional space such as class information to the data space [16]. For example, if you want to create a specific number of values in MNIST dataset as CGAN, you can learn the model by adding a label of that number as an additional space. This extra information y is fed into both the discriminator and the generator as additional input layer. In basic GAN, KL divergence or JS divergence was mainly used to measure the distance between data distributions, both of which had a problem with *mode collapse*. *Mode collapse* is that the generation model produce only certain values of the data (e.g., an image of MNIST labeling 2). To address this, Arjovski used Earth-Mover(EM) loss to understand the probability distribution of the actual data. In [17], Wasserstein-GAN (WGAN) was proposed to minimize the EM distance. WGAN, sensitive to the balance between generators and separators, can solve problems that may arise during learning. WCGAN is a conditional version of WGAN.

III. Experiment

In this section, we explain how we collected and preprocessed bitcoin transaction data. Then, we present oversampling techniques and validation method of synthetic data.

A. Data Preparation

Since Silkroad was closed in November 2014 (when bitcoin block height was about 330,000), the transactions in database were traded between 290,000 and 300,000 blocks, with 200,000 legitimate transactions and 10,000 illegal transactions being sampled. At this time, legitimate transactions were labeled as zero, otherwise as one. We gathered on-chain data such as bitcoin transmission volume, lifetime, fee and sibling number to extract features of the bitcoin transactions. We extracted the total list of extracted 62 features from bitcoin node. However, the data in high-dimensional space could cause problem like overfitting when training a classification model. Therefore, we used Principal Component Analysis (PCA) to transform the data with a total of 16 main component vectors including class label [18]. PCA is a data preprocessing method, usually applied to the feature collections to handle the curse of dimensionality by converting high-dimensional data into low-dimensional data.

B. Data Generation

We implement SMOTE with imblearn [1] python library and generate artificial data. We also developed GAN architectures implemented in Python using popular library like Keras and a Tensorflow-backend [19, 20]. Although GAN in image processing generally uses convolutional network, our dataset uses *densely connected layers*, which are connected to every input and output of the layer, due to the lack of any spatial structure among variables. Firstly, we built vanilla-GAN composed of generator network and discriminator network, using cross-entropy loss to train network. Secondly, we added class labels to the network layer to condition on both generator and discriminator (CGAN). Thirdly, we used the Wasserstein distance metrics to train network and lastly added class labels (WGAN/WCGAN). Each GAN models trained 5000 rounds and then generated illegal transactions.

Fig. 1: Difference between critic loss in WGAN and WCGAN

In WGAN and WCGAN architectures, the discriminator calculates Wasserstein (EM) distance to learn data distribution and be optimized to tell artificial data. If the network is ideally learned, the EM distance between artificial and actual illegal data will be measured close to zero. It means that the critic network loss will not be lower, then no matter how further training may not be helpful. From the critic loss in Fig. 1, however, there still seems to be room for further research in both WGAN and WCGAN.

[1] https://imbalanced-learn.readthedocs.io/en/stable/

We generated data when each model is best iteration with the highest classification score. To investigate the distribution of synthetic and actual data, we visualized them in 2 K-Means classes, Class 1 for legitimate transactions and Class 2 for illegal transactions. We plotted with 2 dimensions (V2, V12 transformed by PCA) that determine which features had a significant impact on classification using Extreme Gradient Boosting (XGBoost) [22]. In other words, XGBoost algorithm calculates and scores importance of input features based on how useful they are for classification. Fig. 2 illustrates the comparison of synthetic data with GAN models.

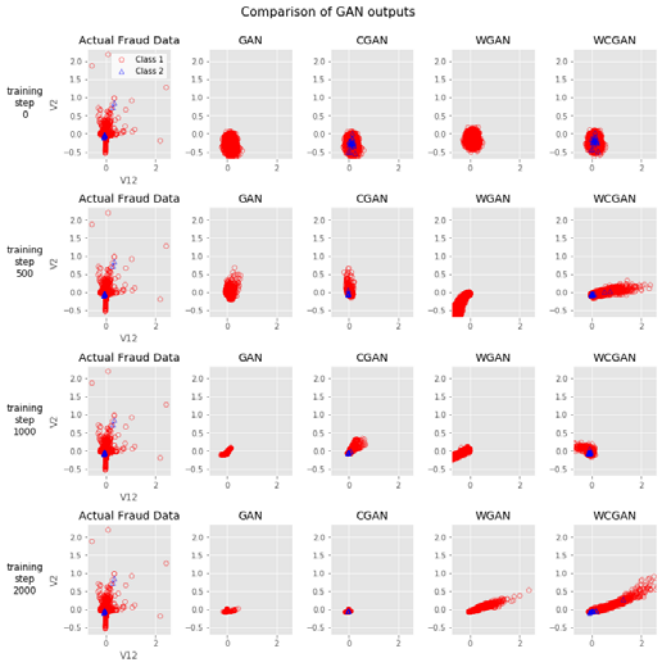

Fig. 2: Comparison of Generated Data with GAN models

C. Evaluation

Synthetic data is produced similar to the actual data in the initial learning phase in vanilla GAN. As the learning phase began to exceed 1000 steps, however, it began to converge into a single point which is not optimal. This seems to be due to mode collapse, where learning converges on a non-optimal sample distribution. In the case of CGAN, the class values each converge to a specific distribution. CGAN does a little better produce than GAN but the mode collapse sets in the end. On the other hand, in WGAN and WCGAN which analyze the distribution between data using EM distance, it was found that all of them did not show mode collapse regardless of class information.

For training set, we used 70% of the legitimate data (140,000 cases) and 20% of the illegal data (2,000 cases). Then we added different amounts of real or generated transaction data to this training set, up to 7,000 cases (70% of the fraud data). For the test set, we used the rest 30% of the legitimate cases (60,000 cases) and illegal (3,000 cases). We tried adding generated data from an untrained GAN to test the performance improvement of random noise. To mitigate overfitting, the results are reported under 5-fold cross-validation. From our tests, it appears that our best architecture was WCGAN at 5000 step. Thus, we compare SMOTE and WCGAN using XGBoost classification scores. The evaluation of the XGBoost was examined quantitatively with data generated by the untrained-WCGAN model, the trained-WCGAN and SMOTE.

TABLE I: Peformance of XGBoost classification at Best Iteration

	auc	precision	recall	roc_auc
Untrained	0.9282	**1.0**	0.2826	0.9833
WCGAN	0.9280	0.9978	0.2808	0.9843
SMOTE	**0.9291**	0.9989	**0.2919**	**0.9848**

Fig. 3: Effects of Additional Data on Classification

As shown in Table I, classification performance metrics show recall value is extraordinarily lower than other metrics. Recall, also known as sensitivity in statistics, is the fraction of illegal transaction samples accurately identified in the test set. In other words, recall value is calculated as the percentage of synthetic data by XGBoost algorithm that predicts illegal transaction. If the model trained data distribution properly(i.e., the value of recall is increased), it means that the synthetic data is similar to the actual data. Fig. 3 shows the change in the value of recall as the generated data is added.

In the case of untrained WCGAN, there was not much change in recall value while the synthetic data was added. This means that the data only generated by random noise is not similar to the actual data. This is reasonable and an expected result. Unfortuately, no significant difference in trained WCGAN was compared with untrained WCGAN. With the addition of trained WCGAN data in classification model, the performance has fallen further. SMOTE shows the highest recall value in best iteration and looks relatively good, but it does not show noticeable performance improvements. In summary, the overall performance of the data generation model (random noise, WCGAN) and SMOTE was not different from around 0.3 baseline.

There are two main reasons why data generation models are not performing well in previous experiments. The first problem is the fundamental limitation of GAN architecture. In the case of GAN architecture, it is highly likely that the parameters of the model will oscillate and destabilized during learning. For instance, if the discriminator gets too successful, the generator gradient vanishes and learns nothing. Thus, the parameters will not converge on optimal values. Especially, GAN is difficult to identify decision variables

in the learning process due to a black-box method such as neural network. For this reason, we expect that there is a possibility of improvement in the loss of WGAN/WCGAN like Fig. 1. Secondly, the quality of bitcoin transaction dataset is a problem. We can can readily spot a lot of overlap and the distribution is clearly concentrated on certain values. This seems to be due to the fact that the dataset was composed using only on-chain data. On-chain data refer to transaction information which occur on the blockchain and remain dependent on the state of the blockchain for their validity. The problem with on-chain data is that key information about the transaction is missing, such as market price and where the trading occurred. To address this problem, off-chain data such as the exchange or price of each transaction is necessarily required.

We provide an additional experiment using DRAGAN and resampled training set [23]. DRAGAN was used to solve training difficulty and low-performance of basic GAN/WGAN architecture. DRAGAN enables faster training, achieves improved stability with fewer mode collapses, and leads to generator networks with better modeling performance across a variety of architectures and objective functions. Furthermore, To reduce dependency on off-chain data, the scope of transaction blocks has been narrowed from 295,000 block to 300,000 block(previously, from 290,000 block to 300,000 block). The number of legitimate transaction decreases from 200,000 to 100,000 and so does illegal transaction decreases about half. Fig. 4 illustrates the effects of additional data on classification with resampled training set in SMOTE and DRAGAN.

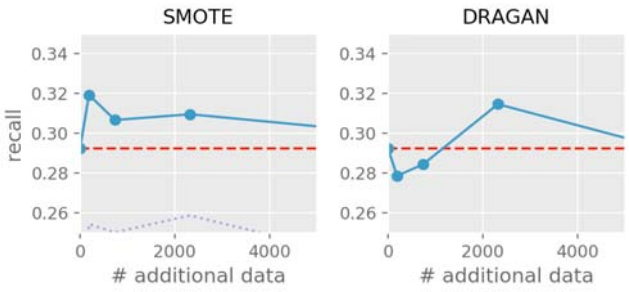

Fig. 4: Effects of Additional Data on Classification with Resampled Training Set

Fig. 4 indicate that DRAGAN shows clear improvement in recall value compared to basic GAN and WGAN architecture. In particular, we can check an average performance improvement of 10 percent during SMOTE doubles its minority-class data. In the imbalanced data, we found that SMOTE shows better performance than GAN-based frameworks.

IV. Conclusion

This study proposed an oversampling method for detecting bitcoin illegal transaction data. It was expected that data augmentation in minority class would solve the problem of imbalanced dataset, thereby improving the performance of classification model. We showed evidence of improved classification performance using an oversampling method.

ACKNOWLEDGMENT

This work was supported by the ICT RD program of MSIT/IITP [No.2018-000539, Development of Blockchain Transaction Monitoring and Analysis Technology] in Republic of Korea.

References

[1] Nakamoto, S. Bitcoin: A peer-to-peer Electronic Cash System (2008)
[2] Foly, S. Karlsen, J. R., & Putniņ,T. J. Sex, Drugs, and Bitcoin: How Much Illegal Activity Is Financed through Cryptocurrencies?. The Review of Financial Studies, pp. 1798–1853. (2019)
[3] Buxton, J., & Bingham, T. The rise and challenge of dark net drug markets. Policy Brief, 7, 1-24. (2015).
[4] Lacson, W., & Jones, B. The 21st Century DarkNet Market: Lessons from the Fall of Silk Road. International Journal of Cyber Criminology, 10(1). (2016).
[5] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. SMOTE: synthetic minority oversampling technique. Journal of artificial intelligence research. pp. 321-357. (2002)
[6] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. & Bengio, Y. Generative ad-versarial nets. In Advances in neural information pro-cessing systems. pp. 2672-2680. (2014)
[7] Zambre, D., & Shah, A. (2013). Analysis of bitcoin network dataset for fraud. Unpublished Report, 27. (2013)
[8] Toyoda, K., Ohtsuki, T., Mathiopoulos, P.T.: Identification of high yielding investment programs in Bitcoin via transactions pattern analysis. In: 2017 IEEE Global Communications Conference, GLOBECOM 2017, Singapore, pp. 1–6 (2017)
[9] Pham, T., & Lee, S. Anomaly detection in bitcoin network using unsupervised learning methods. arXiv preprint arXiv:1611.03941. (2016).
[10] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. Learning from class-imbalanced data: Re-view of methods and applications. Expert Systems with Applications. pp. 220-239. (2017)
[11] Wu, E., Wu, K., Cox, D., & Lotter, W. Conditional infilling GANs for data augmentation in mammogram classification. In Image Analysis for Moving Organ, Breast, and Thoracic Images. pp. 98-106. Springer, Cham. (2018).
[12] Tanaka, F. H. K. D. S., & Aranha, C. Data augmentation using GANs. arXiv preprint arXiv:1904.09135. (2019).
[13] Oh, J. H., Hong, J. Y., & Baek, J. G. (2019). Oversampling method using outlier detectable generative adversarial network. Expert Systems with Applications, 133, 1-8.
[14] Mullick, S. S., Datta, S., & Das, S. Generative adversarial minority oversampling. In Proceedings of the IEEE Inter-national Conference on Computer Vision. pp. 1695-1704. (2019)
[15] Ba, H. Improving Detection of Credit Card Fraudulent Transactions using Generative Adversarial Net-works. arXiv preprint arXiv:1907.03355. (2019).
[16] Mirza, M., & Osindero, S. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784. (2014).
[17] Arjovsky, M., Chintala, S., & Bottou, L. Wasserstein gan. arXiv preprint arXiv:1701.07875. (2017).
[18] Wold, S., Esbensen, K., & Geladi, P. Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52. (1987).
[19] Chollet, F., & others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras
[20] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. Tensorflow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16) (pp. 265-283). (2016).
[21] Ali-Gombe, A., & Elyan, E. (2019). MFC-GAN: class-imbalanced dataset classification using multiple fake class generative adversarial network. Neurocomputing, 361, 212-221.
[22] Chen, T., & Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discov-ery and data mining. pp. 785-794. (2016)
[23] Kodali, N., Abernethy, J., Hays, J., & Kira, Z. On convergence and stability of gans. arXiv preprint arXiv:1705.07215. (2017).