Detecting Online Game Chargeback Fraud Based on Transaction Sequence Modeling Using Recurrent Neural Network

Namsup Lee[1], Hyunsoo Yoon[1], and Daeseon Choi[2(✉)]

[1] KAIST, Daejeon 34141, Korea
[2] Kongju National University, Gongju, Choongnam 32588, Korea
sunchoi@kongju.ac.kr

Abstract. We propose an online game money chargeback fraud detection method using operation sequence, gradient of charge/purchase amount, time and country as features of a transaction. We model the sequence of transactions with a recurrent neural network which also combines charge and purchase transaction features in single feature vector. In experiments using real data (a 483,410 transaction log) from a famous online game company in Korea, the proposed method shows a 78% recall rate with a 0.057% false positive rate. This recall rate is 7% better than current methodology utilizing transaction statistics as features.

Keywords: Online game chargeback fraud · Sequence modeling
Recurrent neural network

1 Introduction

Online game users charge "game money", a kind of virtual currency in the online game world that is spent on purchasing game items such as weapons. This conversion charge from real world to game money is done using online credit card payments.

Two kinds of anomalies are known in these game money charges when someone has lost his or her credit card or credit card information, and the stolen card is used to charge game money, and when a malicious game user charges game money to his or her credit card, spends this in the online game, and files a claim with the credit card company that his or her card or information has been stolen. In the first case, the victim requests a refund and the credit card company grants that request to refund the payments. In the second case, the malicious user can get a refund of the transaction if the game company cannot present evidence that it was intentionally fraudulent. This type of fraud, called *chargeback fraud*, has led to heavy losses for game companies, as they are not usually able to provide sufficient evidence [18, 19].

To detect chargeback fraud, online game companies run rule-based risk management systems defined by experts. Unfortunately, these rules leave out many fraud patterns and are unable to detect newly emerging patterns of fraud. Machine learning based classification methods have been proposed to overcome this limitation [5, 6]. These approaches classify users based on statistics such as the number of countries

B. B. Kang and T. Kim (Eds.): WISA 2017, LNCS 10763, pp. 297–309, 2018.
https://doi.org/10.1007/978-3-319-93563-8_25

where the user has been located and average amount of money charged according to the transaction log. Although these approaches have been able to improve accuracy or recall rate of detection, some normal and abnormal users continue to be misclassified when their transaction statistics are similar with each other. Even if those users' transaction statistics are similar, the sequence of each of their transactions could be quite different. Observing these differences can be used to improve the accuracy of such classification systems. Therefore, this study has proposed a method of user transaction sequence modeling capable of reflecting these differences. The contributions of this work are as follows:

- **First transaction sequence modeling in online game chargeback fraud detection.** We designed a sequence model based upon game money charges and item purchase transactions. To the best of our knowledge, this work is the first attempt to apply sequence modeling for fraud detection in game area. As a result, the proposed method provided a 7% improvement in fraud detection recall rate in experiments using real data.
- **Feature construction including both game money charge and game item purchase transactions for the recurrent neural network.** To model the transaction sequence, we constructed single feature vector that includes features from two different types of transactions and presented recurrent neural network structure for modeling transaction sequence using this feature vector.
- **Evaluation using real transaction data**. We conducted several experiments using real transaction log data provided by a famous online game company in Korea. This work was done to improve the company's chargeback fraud detection system, currently a rule-based detection system. We presented a set of features for this company's data. It is believed that this feature set could be applied to the other game services or similar content businesses. Further, we evaluated the classification accuracy of the proposed method using the real data.

This paper is structured as follows. In the second section, related works on fraud detection in finance and game are introduced. In the third section, the limitations of the statistical detection model are described and the proposed sequence detection model is introduced. In Sect. 4, the proposed model is evaluated based upon some experiments. Section 5 contains a discussion on some considerable points. The paper ends with concluding remarks in Sect. 6.

2 Related Work

FDS (Fraud Detection System) is a complex system designed to identify abnormal transactions by comprehensively analyzing various collected information. It formalizes the usage patterns of users and determines whether a transaction is normal or abnormal based on the established pattern. The previous standard FDS is usually a rule-based system designed by a data analysis expert who analyzes the patterns of abnormal transactions and sets up rules for detecting them [13]. Strong rules make it easier to detect abnormal transactions, but are also more likely to falsely identify normal transactions as abnormal. These rules are also unable to detect some frauds, especially

those following new patterns which will require the expert to make new rules for detecting them. As the volume of data increases, it is necessary to be able quickly and accurately discover usage patterns for users in big data; however, it is impossible for a few human experts to determine such patterns. Some works have applied machine learning to FDS as a feasible method of accomplishing such a task.

A great deal of research has been conducted on detecting fraud in payment transaction data in the financial area. Lim et al. [1] proposed a learning method by giving more heavily weights to recent transactions. Mahmoudi and Duman [2] proposed a method to maximize profits that can be obtained through detection of fraudulent transactions by modifying fisher discriminant function to be cost sensitive to credit card fraud detection problem. Coppolino et al. [3] proposed a fraud detection model for the case of taking and abusing accounts in the mobile banking system: a rule-based learning method and probability-based model are constructed around the logs generated by MMT (Mobile Money Transfer), which are then able to analyze intruders' behavioral patterns to succeed in when committing mobile payment fraud and calculate the possibility of an abnormal transaction. Schaidnageland et al. [4] proposed a model to detect abnormal transactions by analyzing the time series pattern of credit card transactions.

Moving away from purely financial business, research has also been conducted using actual game payment transaction data from MMORPG online games where payment fraud occurs frequently. Woo et al. [5] identified other types of fraud related to payment fraud, analyzed the payment data of each type, and extracted the features necessary for learning via statistical method. In this research, a period of 1 year was divided into 3 months intervals and the statistical features for each section were extracted as learning data, allowing them to present a detection model applying decision tree algorithm. Seo and Choi [6] also proposed a statistical fraud detection model. Their research chose the optimal feature selection for chargeback fraud detection using various methods for data set construction and applying various feature selection algorithms. However, statistical models are limited in their inability to reflect sequence information in data, such that even if the gradient information of some feature's definition of normal and abnormal users is different they will have the same value in a statistical model.

3 Proposed Method

3.1 Sequence Modeling

As mentioned above, statistical modeling is fundamentally limited in that no gradient information in data is reflected from previous transactions. As shown in Fig. 1, a feature for classifying a transaction can be statistically defined as a chunk of previous transactions. If the window size is 3, transaction T3's feature is <700, 1, 1, ..> which provides the statistical values of c1, c2 and c3. In the sample image, T6's feature is also <700, 1, 1, ..>. However, the value of charge amount decreases from 1000 to 550 in the u1 normal user case, whereas it increases from 550 to 1000 in the u2 abnormal

user case. Although the gradient of data is different between normal and abnormal users, a statistical model yields the same data for both. This may result in incorrect classification.

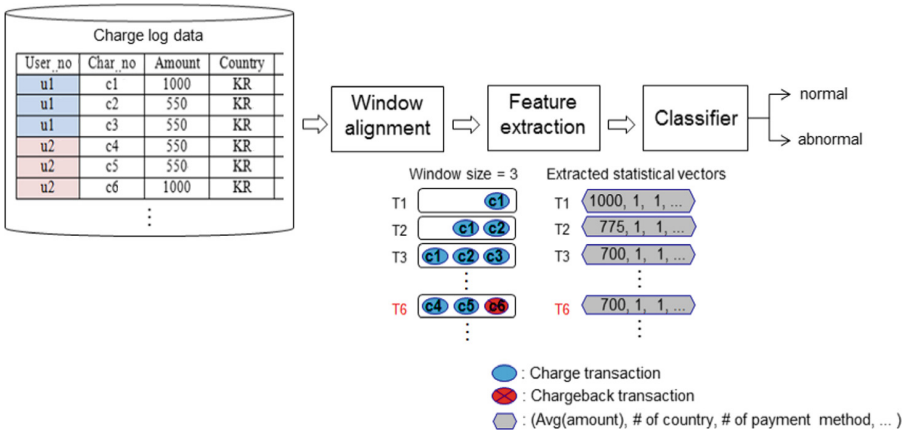


Fig. 1. Statistical modeling of charge data

Therefore, to improve the accuracy of classifications, differences in sequences should be modeled. In Fig. 2, T3's feature is <1000, ..> - <550, ..> - <550, ..> and it is tagged as normal. T6's feature is <550, ..> - <550, ..> - <1000, ..> and it is tagged as abnormal. An RNN (Recurrent Neural Network), which is well-known to show good performance for modeling sequential data, is used to model this sequence feature [7].

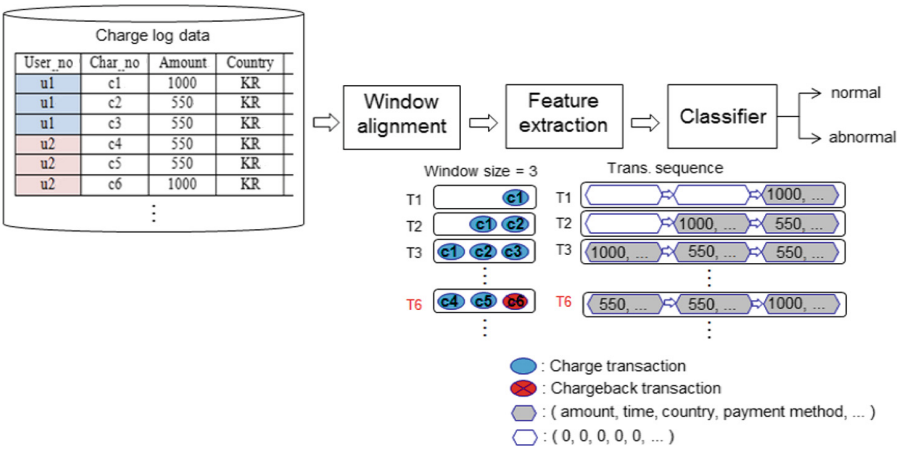


Fig. 2. Sequence modeling of charge data

Feature extraction. The original charge log data is shown in Table 1. 'Transaction_no' is used for sorting transactions in chronological order. 'User_no' is used for grouping transactions by user. These processes are a part of preprocessing for sequence modeling. 'Status' indicates whether or not the transaction has been charged back later, and this field is therefore a class tag.

Table 1. Charge data provided by a game company

Fields	Description
Transaction_no	Transaction identifier
User_no	User identifier
Datetime	Transaction date and time
Country	Country codes
Status	Present the steps of charging
Payment_method	Payment method
Amount	Charge amount
Ip_addr	IP address

Some fields are chosen and processed from the original log data to derive feature vectors. The selected features were 'Country', 'Datetime', 'Payment_method' and 'Amount' as shown in Table 2. Because the 'Country' and 'Payment_method' features are categorical data, the input vector for the features are constructed by one-hot encoding, a feature engineering technique. In the case of the 'Datetime' feature, trigonometric [8] functions such as sine and cosine were used to reflect cyclic characteristics. The 'Amount' feature was used as it is.

Table 2. New features extracted from the original charge data

Features	Description
Country	Country codes
Payment_method_no	Identifier of payment methods
Time_x	x-coordinate of time in a cyclic form
Time_y	y-coordinate of time in a cyclic form
Amount	Charge amount

Recurrent Neural Net. The machine learning algorithms capable of training sequential model can be either HMM (Hidden Markov Model) based on probability theory or RNN (Recurrent Neural Network) based on an artificial neural network. It has already been proven that if there are enough units in the hidden layer, an RNN has the capability to map a sequential data to another sequential data as we want. Therefore, we adopt a RNN, actually LSTM RNN (Long-Short Term Memory Recurrent Neural Network) that is a variation of RNN to be able to train long sequence of data, as a learning algorithm.

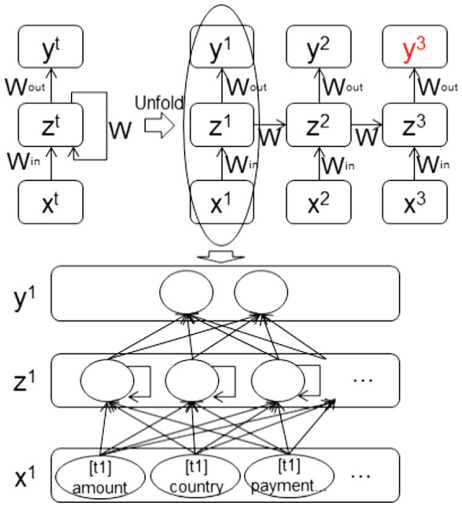


Fig. 3. Basic RNN architecture

The basic architecture of a RNN is that the result of calculating the current input vector becomes a piece of the calculation of the next steps' input vector. The problem dealt with in this work is accurately predicting whether or not the current transaction will be charged back in the future. Therefore, the window aligned dataset has to be modeled forward from previous transactions to the current transaction.

The internal procedure of calculating the output value is as follows. The status of hidden layer at $t(z^t)$ is calculated with the previous status (z^{t-1}) and current input vector (x^t), the result of feature engineering as mentioned above, as shown in Fig. 3. This step is repeated to calculate each time t steps. The formula of this procedure is as follows.

$$z^t = f\left(W^{(in)}x^t + \mathrm{W}z^{t-1}\right) \tag{1}$$

The output of the hidden layer at $t(\mathrm{y}^\mathrm{t})$ is calculated with z^t. We use the softmax function for 2-class parameters as the activation function (f), since the problem is a binary classification of whether or not each transaction will be charged back. We take the last step of output as a prediction value.

$$\mathrm{y}^\mathrm{t} = f^{(out)}\left(W^{(out)}z^t\right) \tag{2}$$

3.2 Purchase-Combined Sequence Modeling

As mentioned above, there are two types of transaction data: charge data and purchase data. These game-dependent types of transaction can present a considerable difference in behavioral pattern, making it possible to differentiate between normal and abnormal

users. Usually, a normal user will make one charge and then purchases game items until the charge amount has been depleted, whereas abnormal users are more likely to make many charges at one time and then purchases game items. The difference is shown in Fig. 4.

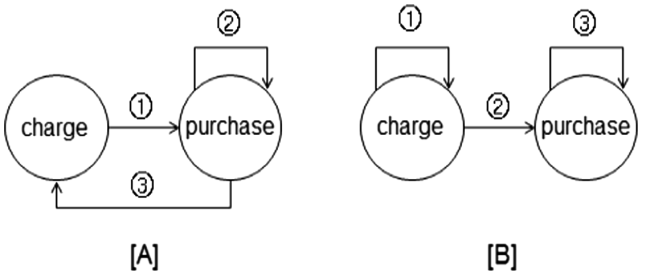


Fig. 4. State diagram of purchase behavioral pattern: [A] normal vs [B] abnormal user patterns

The result of analysis on real game transaction data is as follows: pattern 'A' is seen 68% for normal users and 33% for abnormal users, while the ratio for pattern 'B' is 32% normal and 67% abnormal users. Thus, sequential modeling that combines purchase transaction has a positive meaningful correlation to classification.

The basic concept of purchase-combined sequence modeling is as follows. The first purchase transaction per dedicated charge transaction is taken (this is sufficient to reflect the features described above), and uncommon features between charge and purchase transactions are solved by padding with 0 (an acceptable solution in RNN). A more detailed description is given in part 3.2.1, Feature Extraction. The whole description is shown in Fig. 5.

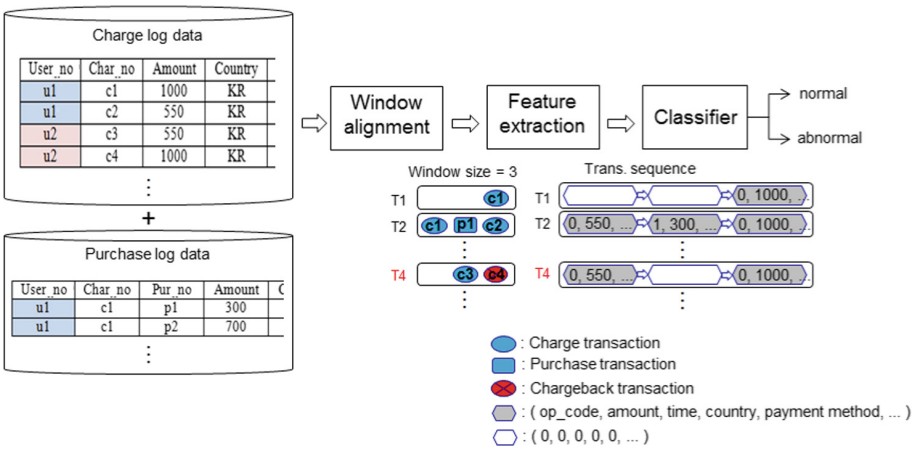


Fig. 5. Purchased-combined sequence modeling

Feature Extraction. Table 3 shows the extracted features for applying purchase-combined sequence modeling to RNN.

Table 3. New features extracted from the original charge and purchase data

Features	Description
Country	Country codes
Payment_method_no	Identifier of payment methods
Time_x	x-coordinate of time in a cyclic form
Time_y	y-coordinate of time in a cyclic form
Amount	Charge amount
Op_code	Identifier of whether type of input vector is for charge transaction or purchase transaction

These extracted features are made by combining some charge and purchase features, consisting of common, uncommon, and new features. We extract only one feature for each of the common features even though this represents two features, one for charge and another for purchase. In order to reflect the purchase behavioral pattern, we extract a new feature named 'op_code' for distinguishing types of input vector. When making an input vector for a charge transaction, purchase features are filled with 0 and vice versa.

4 Experiment and Evaluation

4.1 Dataset

Our original dataset, provided by a world-famous game company in South Korea, was collected over 34 months (May 2014 to February 2017). The transaction data is from European users participating in this online game. The data contained 93,520 normal users who never charged back and 621 abnormal users who maliciously charged back at least once. There were 483,410 normal transactions and 3,452 abnormal transactions, as in Table 4.

Table 4. Number of normal and abnormal users and transactions

	# of user	# of transaction
Normal	93,520	483,410
Abnormal	621	3,452

4.2 Evaluation Method

Prior work [6] has used data of purely abnormal users who chargeback all transactions and created a user-level detection system that classified users rather than transactions. However, the real data used in this work contained 379 impurely abnormal users (60% in total abnormal users) who charged back only some of their transactions. This shows

that the evaluation approach in previous work's evaluation is not appropriate for comprehensive fraud detection. Thus, we compare previous statistical model with the proposed sequence model in transaction-level so as to show that the sequence model is better. We also compare sequence model with purchase-combined sequence model in transaction-level to ascertain that purchase-combined sequence model is better.

Table 5. Confusion matrix

		Prediction	
		True	False
Observation	True	True Positive (TP)	False Negative (FN)
	False	False Positive (FP)	True Negative (TN)

The confusion matrix defined in Table 5 was used as a performance estimation tool. The performance metric was set as the f1-score, since the goal of abnormal transaction detection is to detect the greatest number of abnormal transactions with the highest accuracy. The formula of f1-score is below:

$$f_1 = 2 * \frac{precision * recall}{precision + recall} \tag{3}$$

In the previous works, a decision tree was used as a classifier. However, a decision tree has a disadvantage of large variation in the results or performance. Additionally, it is difficult to regard a decision tree as a generalized model, as a decision tree generated can be different every time. Thus, a random forest that overcomes these shortcomings and has good generalization performance is applied to the statistical model for estimating performance. Scikit-learn [14], the python machine learning library, was used to implement this random forest. For the sequence model, RNN is applied as mentioned above, and the number of units in the hidden layer is set to 250 while the epoch is set to 10 (based on the result of finding optimal epoch, shown in Fig. 6). Keras [15], the python deep learning library, was used to implement the RNN. This is a high-level neural networks API, capable of running on top of TensorFlow, CNTK or Theano. The deep learning framework used in this work is TensorFlow.

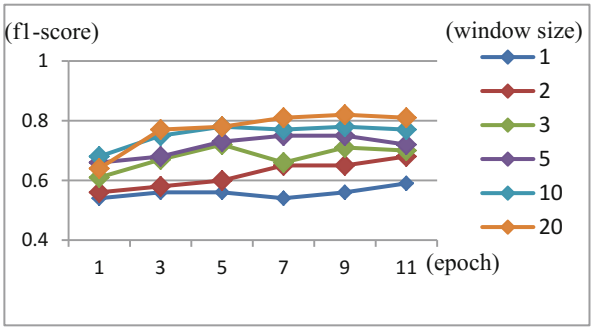
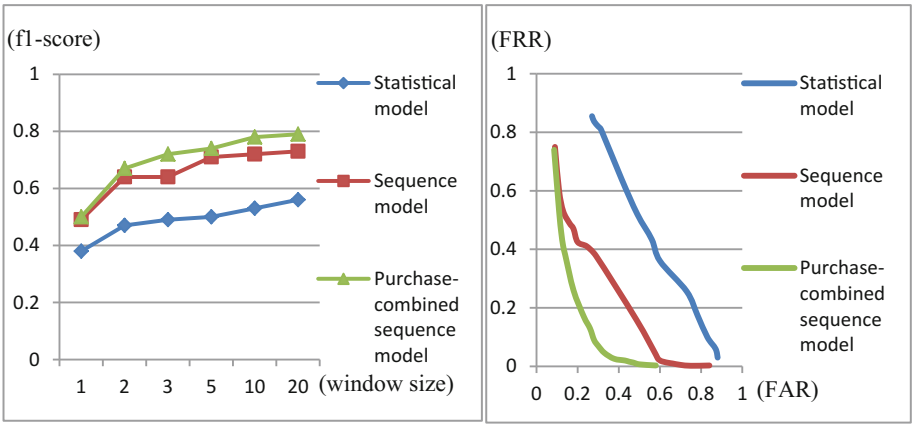


Fig. 6. Epoch of abnormal class for window size

4.3 Experimental Results

The performance for each model was measured with increasing window size. The experimental results are shown in Table 4. Since there is a class imbalance problem between normal and abnormal transactions, all models have a good performance for the normal class '0' as shown in Table 4. Therefore, model evaluation was conducted by extracting the f1-scores of the abnormal class according to window size increase, as shown in Fig. 7(a), and the ROC (Receiver Operating Characteristic) curve of the abnormal class, shown in Fig. 7(b).



(a) **(b)**

Fig. 7. (a) F1-score for each model (b) ROC curve for each model at window size 20

In terms of f1-score, the performance of the sequence model is seen to be better than that of statistical model while the performance of purchase-combined sequence model is still better than that of sequence model. As the window size increases, the performances of all models increase as additional historical data is available for reflection. The performance of the two sequence models is shown to be bounded at some window size, whereas the performance of the statistical model is shown to be unbounded in Fig. 6. In other words, it is possible to misread the data and assume that the performance of statistical model may be above that of the sequence models if window size increases. Therefore, the f1-score of the statistical model was measured at window size of 50, and it was found that it does not increase above 0.6.

The purchase-combined sequence model is also better than the others in terms of the ROC curve. The ROC curve of the purchase-combined sequence model is closer to a good ROC curve than those of the statistical model and basic sequence models. It is therefore shown that the purchase-combined sequence model would be optimal for a service provider to give an option that which one he will focus on between rejection and acceptance service.

Table 6. Test results of confusion matrix for each model at window size 20.

Model		True	False
Statistical model	True	239797	182
	False	992	734
Sequence model	True	239822	157
	False	432	1294
Purchase-combined sequence model	True	239842	137
	False	381	1345

Table 6 shows the test results of the confusion matrix for each model at a window size of 20. It can be seen that the false positive value decreases dramatically between statistical model and sequence model. It is shown that the purchase-combined sequence model has the smallest false positive value (0.057%) and the highest detection ratio of abnormal transactions at 78%.

5 Discussion

There remain several issues that should be considered for applying the proposed methodology in a real service environment.

The first issue is the time required for classification. In our experiment, the time consumed for 241,705 transactions was 322 s in the server using a Xeon E5-2609 1.7 GHz CPU. For each transaction, 0.0013 s is required. The average daily transaction count given in the experimental data was 703. Therefore, the proposed method will not critically increase the time overhead.

The model construction period is another potential issue. The time required during the experiment for training 241,705 transactions in a sequence of 20 steps each in a RNN model was approximately 1,770 s, when the training epoch is 10. It would therefore be possible to update the model daily in a real world environment.

A third issue worth discussing is the potential of false classifications. In the results of this experiment, the false positive rate of fraud detection for the proposed method is 0.057%. This is lower than the previous method's 0.076%. In game money charge operations, the game service provider does not directly abandon the transaction. They require additional procedures such as secondary authentication for the suspicious transaction. Thus, a 0.057% false positive rate is so trivial that could be neglected.

The final issue is about resistance to attacks for deceiving machine learning based classifiers. Attacks such as the poisoning attack [9, 10] and evasion attack [11, 12] were proposed. The poisoning attack is performed by injecting poisoned data (e.g. a fraud transaction with normal tag) as training data to a classifier, causing the classifier to produce a false result. In the performed experiment, a charge that is refunded within 6 months (after that the credit card company does not refund the transaction) is defined as abnormal. The attacker should pay for such charge transactions and not get refunded in order to make deceiving transactions that have shape of typical their transaction patterns and are not refunded. As such attackers are in the minority, each has to make

many transactions by themselves to affect the classifier's decision. A classifier is trained with more than 200,000 transactions. In [9], it was shown that about 10% of data must be poisoned in order to degrade the classifier's accuracy by 10%. Therefore the poisoning attack is not practical in the respect of cost effectiveness. In evasion attack designed by making data such that a classifier produces false results, the confidence value of any decision should be required with the classification result of any test data. Of course, the proposed system does not provide any confidence value to users, so that the evasion attack could be impossible. However, in the case of normal users that turn to charging back transactions, they could easily make an evasion attack with only the classification result because they have a sequence of normal transactions that were classified to normal. Also, a classifier is not able to detect an attack that doesn't exist within a given dataset. Therefore, additional research is required for making a classifier capable of defending against those attacks: and making malicious data artificially via GAN (Generative Adversarial Network) [16] and training the data may be one of the solutions that improve the capability of a classifier so that it can defend such attacks.

6 Conclusion

A sequence model for online game fraud detection at the transaction-level was proposed which overcomes the limitations of the currently used statistical model, including misclassification of normal and abnormal users who have different transaction sequences. In addition, a difference in purchasing behavioral patterns was found between normal and abnormal users. Based on this, a purchased-combined sequence model was suggested to reflect the difference. The proposed models were tested with real game transaction data provided by a world-famous game company in South Korea, and results showed that the performance of both developed sequence models was better than that of the existing statistical model, with the purchase-combined sequence model showing the best performance overall.

This work focused on explaining the evaluation of superiority in our proposed model. However, there is a class imbalance problem in the data, which will also require research work for performance improvement. Therefore, future work will try to solve this class imbalance problem by applying a GAN as a novel approach and comparing the performance with existing techniques such as SMOTE [17] or under-sampling. Further research will also be required to defend against such attacks as mentioned in the discussion; and applying GAN as a method may also be capable of making a classifier defendable against such an attack.

Acknowledgments. This work was partly supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (B0717-16-0139, Security Technologies for Financial Fraud Prevention on Fintech) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2016R1A4A1011761).

References

1. 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
2. Mahmoudi, N., Duman, E.: Detecting credit card fraud by modified Fisher discriminant analysis. Expert Syst. Appl. **42**(5), 2510–2516 (2015)
3. Coppolino, L., D'Antonio, S., Formicola, V., Massei, C., Romano, L.: Use of the Dempster-Shafer theory for fraud detection: the mobile money transfer case study. In: Camacho, D., Braubach, L., Venticinque, S., Badica, C. (eds.) Intelligent Distributed Computing VIII. SCI, vol. 570, pp. 465–474. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-10422-5_48
4. Schaidnagel, M., Connolly, T., Laux, F.: Automated feature construction for classification of time ordered data sequences. Int. J. Adv. Softw. **7**(3), 632–664 (2014)
5. Woo, J.Y., Kim, H.N., Kwak, B.I., Kim, H.K.: Abnormal transaction detection model based on online game payment data analysis. Korea Inst. Inf. Secur. Cryptol. **26**(3), 38–44 (2016)
6. Seo, J.H., Choi, D.: Feature selection for chargeback fraud detection based on machine learning algorithms. Int. J. Appl. Eng. Res. **11**, 10960–10966 (2016)
7. Hammer, B.: On the approximation capability of recurrent neural networks. Neurocomputing **31**(1–4), 107–123 (2000)
8. Trigonometric Function. https://en.wikipedia.org/wiki/Trigonometric_functions. Accessed June 2016
9. Mozaffari-Kermani, M., et al.: Systematic poisoning attacks on and defenses for machine learning in healthcare. IEEE J. Biomed. Health Inform. **19**(6), 1893–1905 (2015)
10. Poison attacks against machine learning, Security and spam-detection programs could be affected (2012). http://www.kurzweilai.net/poison-attacks-against-machine-learning
11. Vaidya, T., Zhang, Y., Sherr, M., Shields, C.: Cocaine noodles: exploiting the gap between human and machine speech recognition. In: 9th USENIX Workshop on Offensive Technologies (WOOT 2015) (2015)
12. Szegedy, C., et al.: Intriguing properties of neural networks, preprint arXiv, arXiv:1312.6199 (2013)
13. Patcha, A., Park, J.M.: An overview of anomaly detection techniques: existing solutions and latest technological trends. Comput. Netw. **41**(12) (2007)
14. Scikit-learn Homepage. http://scikit-learn.org/. Accessed 15 June 2017
15. Keras Homepage. https://keras.io/. Accessed 15 June 2017
16. Goodfellow, I.J., et al.: Generative Adversarial Networks, preprint arXiv, arXiv:1406.2661 (2014)
17. Chawla, N.V., et al.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
18. https://chargeback.com/events/money2020-2016/. Accessed 15 June 2017
19. http://gametoc.hankyung.com/news/articleView.html?idxno=42921. Accessed 15 June 2017