Cluster Computing (2023) 26:927–943
https://doi.org/10.1007/s10586-022-03674-4

An evaluation of deep learning models for chargeback Fraud detection in online games

Yu-Chih Wei[1] · You-Xin Lai[1] · Mu-En Wu[1]

Received: 1 December 2021 / Revised: 16 April 2022 / Accepted: 22 June 2022 / Published online: 25 July 2022

Abstract
More and more gamers are willing to pay for games. It has been estimated that the global gaming market is worth nearly US$160 billion. Chargeback services offer gamers the convenience of refund mechanisms but are often used by malicious online gamers to commit fraud, causing huge adverse impacts on the online game industry. To combat chargeback fraud, some online game providers resort to manual checking and blocking of malicious accounts, which may incur huge labor costs in the process. In this research, various deep learning models, including recurrent neural networks, long short-term memory networks, and gated recurrent units, are evaluated on their accuracy and performance in detecting malicious chargebacks in online games. In addition, traditional models, such as decision trees, k-nearest neighbors, support vector machines, and random forests, are also evaluated for comparison. The evaluation results show that the Matthews correlation coefficients of the deep learning models range between 0.84 and 0.97. In addition, the gated recurrent unit and long short-term memory network models also outperform other traditional machine learning models in the experiments in this research. Furthermore, the practical feasibility is also taken into consideration in this research by calculating the time overhead of a single transaction to determine whether there is a significant increase in time costs. Although deep learning models are less efficient than traditional machine learning models, deep learning models remain competent in minimizing losses of online game companies.

Keywords Online Game · Malicious Chargeback · Deep Learning · Online Transactions

1 Introduction

The scale of the global gaming market reached US$160 billion in 2020 [1]. It was estimated that it could reach almost US$256.9 billion in 2025 [2]. With the rapid growth of the gaming market, there are more and more gamers who are willing to pay for games. Some would even pay tens of thousands of dollars in a game. Common payment gateways, such as Google Play and Apple Store, offer chargeback mechanisms to avoid losses that consumers may suffer as a result of top-ups of gaming credits or purchases of game products by mistake. Some malicious online gamers take advantage of the convenience of the chargeback mechanisms of store coins or tokens in in-game stores to carry out fraudulent acts. They first purchase a large sum of gaming credits on an online payment gateway, wait until they have obtained virtual treasures or virtual items with store coins or tokens in an in-game store, immediately use them or resell them privately, and then apply for chargebacks. In China, there are professional chargeback firms that specialize in arranging chargebacks for consumers. They charge 55% of chargeback amounts as their fees. It has been reported in the Korea Times that many Korean gamers profit from the loopholes in the chargeback mechanisms. When a Korean gamer purchases a virtual product on a payment gateway, a chargeback may be automatically obtained without any manual approval if the chargeback application is made within a certain period

✉ Yu-Chih Wei
vickrey@mail.ntut.edu.tw

You-Xin Lai
t108ab8016@ntut.org.tw

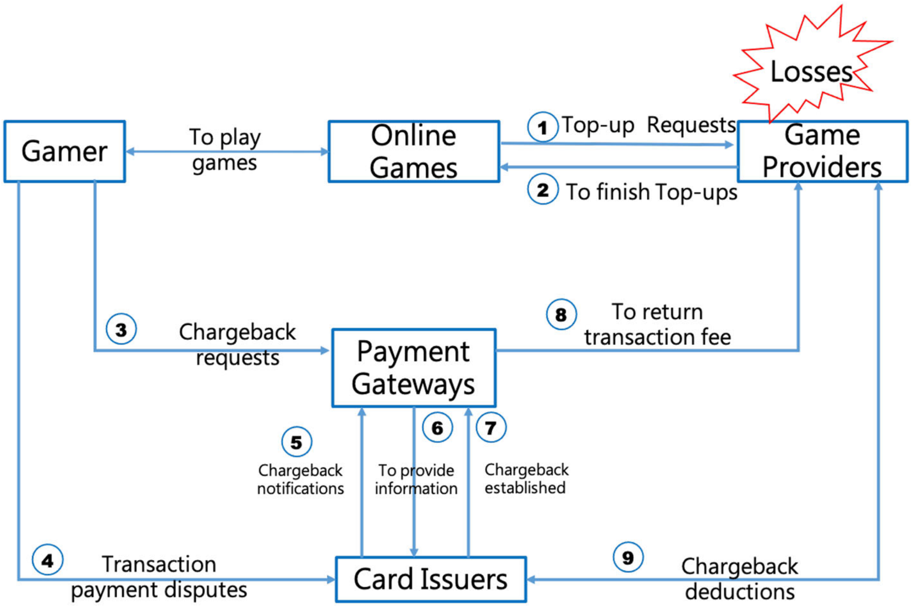
Mu-En Wu
mnwu@ntut.edu.tw

[1] Department of Information and Finance Management, National Taipei University of Technology, Taipei 10608, Taiwan R.O.C.

Table 1 Comparisons and analyses of related works in this research

References	Subjects	Models	Particulars	Evaluations
[3]	Online game chargeback fraud	Statistical model, RNN	An optimal window size based on combinations of features, such as purchases, was amended and found	F1, ROC
[15]	Credit card transactions	CNN, SLSTM, CNN-LSTM	PCA and undersampling were used for unbalanced datasets, and four sampling rates and numbers of three PCA features were compared	Accuracy, ROC, AUC
[16]	Credit card transactions, real time	NN, SVM, RF, CNN	Features based on the transactional disorder were proposed, and transactional features and time intervals were used to form two-dimensional arrays for machine learning	F1
[17]	Online transactions, real time	CNN, BP	A feature sorting layer was used before inputting data into a CNN model, because the authors believed that the sequence of inputting features into the CNN model would affect its results	Accuracy, Precision, Recall, F1
[13]	Credit card transactions	LSTM, RF	A feature aggregation method was proposed in their research	AUC
[14]	Credit card transactions	LSTM, GRU, 6 MLs	Imbalanced datasets were processed through three sampling methods, and their model combined a two-way LSTM with a two-way GRU	AUC, Precision, Recall, F1
[18]	Credit card transactions, real time	DNN, 4 MLs	A model based on a deep neural network technology and the auto-encoder was proposed	Accuracy, Precision, Recall, F1
[19]	Credit card transactions	DNN, RBM	The auto-encoder, an unsupervised learning algorithm, was trained by setting inputs equal to outputs and applying backpropagation	AUC
[10]	Credit card transactions	DNN	H2O, a deep learning package and an effective framework for processing large datasets and performing deep learning, was used to train deep learning models	MSE, RMSE, MAE, RMSLE
[11]	Credit card transactions	ANN, RNN, LSTM, GRU	The feature of domain expertise was proven to improve the prediction accuracy of the models for detecting credit card fraud on unbalanced data	Accuracy
[20]	Credit card transactions	LR, GBDT, DNN	The auto-encoder was used as an unsupervised feature engineering method, and three supervised classification models and six different feature sets were used for comparison	AUC
[12]	Credit card transactions	LSTM, SVM	The authors believed that instead of looking at a single transaction, it made more sense to look at the sequence of the entire transactions	AUC, MSE

Fig. 1 Gaming service providers do not participate in malicious chargeback processes on online gaming payment gateways



of time from the time of the purchase. The payment gateway companies state that to protect the privacy of users, a gaming service provider does not participate in the chargeback application process (as shown in Fig. 1), and it, therefore, cannot identify a user who has applied for a chargeback, nor can it trace the virtual items that the user has purchased to recall them. This enables users to obtain free virtual items. These malicious chargeback behaviors have affected the online gaming ecosystem and caused huge impacts on the business operations of online gaming service providers.

Chargebacks usually occur on application platform, such as Google Play and App Store, for reasons, such as purchases of wrong products, top ups of gaming credits by mistake or accidental purchases by children. Apart from these common reasons, taking Google Play as an example, "unauthorized chargebacks" often occurs. That is when a user tries to stop a chargeback process when his/her family and friends have applied for a chargeback on his/her behalf by mistake. Online gaming service providers are however unable to stop chargeback processes and can only let them continue to happen, which lead to losses in the gaming service providers.

The abovementioned malicious user behaviors have affected the online gaming ecosystem as well as caused huge impacts on the business operations of online gaming companies. Outside of Taiwan, online gaming companies set up risk management systems in accordance with rules set out by experts to deal with chargeback fraud. Unfortunately these rules cannot detect new fraudulent activities in real time, which lead to many loopholes in the risk management systems [3]. For now, the majority of gaming companies can only passively react to malicious chargeback behaviors, i.e. blocking the users' accounts after malicious chargebacks have been made. Malicious chargebacks continue to cause huge losses to online gaming companies.

Deep learning preventive measures for malicious chargebacks in the online gaming industry are proposed in this research. Through this research, online gaming companies will be able to move from passively to proactively dealing with malicious chargebacks. Our proposed predictive model can predict malicious chargeback activities and respond to them as they happen. For example, it can stop users from buying more gaming credits, reject potential malicious chargeback transactions, or stop users from buying virtual products etc. to prevent chargeback fraud. Although setting up rules and manual checking and blocking of fraudulent chargeback accounts may initially reduce some losses caused by fraudulent chargebacks, they require a huge amount of manpower to carry out the account checking and blocking processes. Sometimes errors occur where VIP accounts are blocked by mistake, causing dissatisfaction to these users. To save effort from manual checking and to reduce human errors, using deep learning detection technologies to detect anomalies can help discover suspected fraudulent activities early, so that suspicious malicious accounts can be blocked to reduce operational losses.

The contributions of this research are as follows:

1. setting up predictive deep learning models for malicious chargebacks;
2. increasing the efficiency in detecting malicious chargebacks; and
3. reducing error rate where normal top-ups of gaming credits are mistaken for malicious chargebacks.

2 Related works

2.1 Machine learning in detecting transactional anomalies

Awoyemi et al. [4] considered that credit card fraud represented a high percentage in online banking fraudulent transactions. Credit card fraud detections were highly challenging due to the fact that fraudulent behaviors evolved all the time and datasets collected from credit card fraud were highly imbalanced. 284,807 datasets were collected from European cardholders and used in their research. A mixed sampling method (mixing undersampling and oversampling methods) was used in their research so that the distributions of two sets of data were parallel. As the datasets in their research were highly imbalanced and biased towards the positive, a principal component analysis was used at the analytical stage to convert characteristics into 28 main components. The highly imbalanced datasets of credit card fraud were tested through the naïve bayes classifiers, the k-nearest neighbor method and the logistic regression classifiers. The performance of the test model was evaluated based on accuracy, sensitivity, specificity, precision, Matthews correlation coefficients and balanced classification rates. Their experimental results showed that the best accuracy of the naïve bayes classifiers, the k-nearest neighbor method and the logistic regression classifiers were 97.92%, 97.69% and 54.86%, respectively. Their comparison results showed that the k-nearest neighbor method performed better than the naive Bayes classifier and the logistic regression techniques.

Choi et al. [5] believed fraud and uses of e-payments continued to increase in a proportional manner, and many gamers had been abusing the return and chargeback policies on payment gateways to illegally obtain funds for gaming. Therefore, machine learning technologies were used, and the transactional data was provided by a world-renowned gaming company in their research, to predict fraudulent activities. The transactional data provided by the company was highly imbalanced. It was necessary to correct deviations in the original datasets. The authors proposed methods, such as resampling, new algorithms and feature selection, to solve the problems. In their experiments, decision trees and SVMs were used as the experimental models. Their experimental results showed that the SVMs performed better than other methods. The oversampling technique, SMOTE, could improve the performance by more than 30%.

Chen et al. [6] believed that category imbalance would be encountered in the information retrieval and filtering in the practical application of machine learning, data mining and credit card fraud detection. These often caused classifiers to perform less than expected. It is, therefore, essential to use feature selection methods for classifiers to achieve optimal performance. The authors proposed a new feature selection method, namely the feature assessment by sliding thresholds (FAST). FAST is based on the area under a ROC curve. For the imbalanced data classification, FAST and two other commonly used feature selection methods (correlation coefficients and relevance in estimated features) were compared in their research. Their experimental results obtained on the text mining, mass spectrometry and microarray datasets showed that this method was superior to the relief method and other related methods on imbalanced datasets and was comparable to balanced datasets. When the number of features is small, the classification performance of this method is significantly improved, compared with those related to and based on the relief method.

Carneiroa et al. [7] discovered that credit card fraud had caused billions of losses to online merchants. With the development of machine learning algorithms, researchers have discovered more and more sophisticated methods of detecting fraud. However, few actual implementations of these methods have been reported. The authors described the development and deployment of fraud detection systems in large e-commerce retailers and explored the combination of manual and automatic classification methods. To obtain more important variables to train the models, the authors designed a feature selection method for new variables through the abstraction and combination of variables. They provided insights into a complete development process and compared different machine learning methods. This research helps researchers and practitioners design and implement systems based on data mining for fraud detection or similar problems. Their project has not only contributed to the development of automated systems, but also provided suggestions for fraud analysts to improve their manual revision processes, resulting in their relatively superior performance overall.

Mao et al. [8] considered that the fact that e-commerce transaction fraud was constantly evolving was a major problem. The fast-changing fraud patterns had changed basic data generation systems and caused the performance of machine learning models to decline, making e-commerce merchants unable to obtain powerful machine learning models for fraudulent transaction detections. To overcome the “concept drift” problems in statistical modeling, the authors quantified the fluctuations in probability distributions of risk features from certain documents caused by concept drift and proposed a method to add dynamic risk features as model inputs. The dynamic risk functions are functions based on entities with fraudulent feedback. The authors also explained that the strategy could successfully deal with the impact of concept drift under the framework of statistical learning and verified the method in many ongoing businesses. They also verified that the proposed dynamic model had a better ROC curve than the static model based on the same data and training parameters.

de Sousa Tedim [9] did not consider fraud a brand new problem, as it had been discussed since the beginning of business developments. With the development of the Internet, it had become more and more advanced and had turned into a billion-dollar business. Traditional data analyses used to detect fraud involved different disciplines, such as the economics, the finance, and the law. The complexity of these disciplines quickly made traditional data analyses obsolete. As fraud was an adaptive crime, different techniques, such as data mining and machine learning, had been developed to identify and prevent fraud. The authors used regression models, neural networks, decision trees, ensemble neural network models, and other data mining and machine learning methods to develop their prediction models. Their results showed that the ensemble model could correctly predict 71% of the observation results of the validation sets with 74% accuracy. Their research has been used to help identify and prevent online financial fraud in the Portuguese betting market.

2.2 Deep learning in detecting transactional anomalies

In the past, it was relatively rare for deep learning to be applied to detection models for malicious chargebacks in online games. In South Korea, some scholars have put forward related studies and models. Lee et al. [3] proposed a detection model based on recurrent neural networks (RNNs) to deal with the malicious chargeback problems in the online games in South Korea. Traditional RNN models were used for training. Features, such as combinations of purchases of users, fee charges, amounts of purchases and country/regions, were also used for model training. The reason for using the RNNs was to take the time factor and the order of operations into consideration. They indicated that one of normal users' purchase habits was to make an initial payment (to purchase initial credits) first, and then make another payment (i.e. to top up more credits) when they had used up the credits they previously purchased, while fraudulent users would usually make a huge payment (to purchase initial credits) first, and did not pay again (i.e. to top up more credits) after they had used up the credits they previously purchased. This highlighted the differences in users' payment habits. Therefore, the team used a sequence model for evaluation. The end results of this method achieved 78% in recall and 0.057% of false positive in performance, improving the recall by about 35%, compared with traditional statistical models.

As mentioned at the beginning, it is uncommon to apply deep learning to detection models for malicious chargebacks in online games. Therefore, relevant literature about online transactions and credit card transactions are discussed in this paper. Pandey [10] stated that people and financial companies mostly relied on online services for transactions, which had led to an exponential increase in abnormal credit card transactions. Fraudulent credit card transactions had resulted in huge financial losses. To reduce the losses caused to customers and financial companies, an effective fraud detection system must be designed. As online game transactions are just like online transactions conducted through the Internet, their features are relatively similar, such as IP locations, regions, and the login and logout information etc. As for online game transactions, most of the top up transactions are completed through credit card payments, which contain a certain degree of similarities between online game and credit card transactions, including user transaction habits, amounts of transactions and order of transactions etc. Roy et al. [11] mentioned in their research that as digital payment platforms, such as Apple Pay, Android Pay and Venmo, became more common, losses caused by fraudulent activities on these platforms were estimated to increase. This is similar to online game transactions. Online game payment is usually made through online platforms, such as App Store and Google play, so that the literature on credit card transaction anomalies may help solve issues associated with online game transactions.

As to abnormal credit card transactions, methods based on time series have also been proposed. As early as in 2009, Wiese and Omlin [12] proposed to use the long short-term memory (LSTM) as a solution to detect fraud in credit card transactions. They believed that instead of looking at a single transaction, it made more sense to view the sequence of the entire transactions. Their results confirmed their hypothesis to a large extent. The LSTM, which detects subtle changes in shopping behavior through time series, has been proven to be a very successful method. Roy et al. [11] analyzed four different types of deep learning, i.e. the artificial neural networks (ANNs), the RNNs, the LSTM and the gated recurrent units (GRUs). Their analyses showed the importance of the time component. The LSTM and the GRU models were significantly better than the ANN model. This indicates that the sequence of account transactions is an important feature for distinguishing fraudulent from nonfraudulent transactions.

Jurgovsky et al. [13] mentioned that with the continuous increase in the number of electronic payments, the threat of credit card fraud had become a major challenge for financial institutions and service providers, forcing companies to continuously improve their detection systems. But in fact this is not the case. Although methods based on machine learning are quite popular in other fields, their growth in actual commercial applications is quite slow. J. Jurgovsky et al. considered solving the fraud detection problems a task for sequence classifications. Therefore, they used the LSTM to solve the problems and proposed a feature aggregation method. It was confirmed that the results of the models could be effectively improved after features were merged. After comparing with the random forests, their results showed that the LSTM could effectively improve the accuracy in identifying legitimate credit card transactions in online shops. In addition, Najadat [14] and his research team also applied the LSTM to abnormal credit card transactions, but unlike in the past, the LSTM they used was BiLSTM-MaxPooling-BiGRUMaxPooling, which was the combination of a two-way LSTM and a two-way GRUs. During the research process, they also used three sampling techniques to solve problems of imbalanced datasets. The three sampling techniques they used were the synthetic minority oversampling technique (SMOTE), the random oversampling and the random undersampling. During the modeling process, they used max pooling to extract relatively important feature values for machine learning. In the end, the research team compared six types of machine learning, including the naïve base, voting, the Ada boosting, the random forests, the decision trees and the logistic regression. Compared with machine learning classifiers, the BiLSTM-MaxPooling-BiGRU-MaxPooling model proposed by the team performed better and obtained an F1-Measure of 91.37%.

Heryadi [15] and his team even proposed the use of a variety of deep learning models to detect credit card transaction abnormalities, including convolutional neural networks (CNNs), the stacked long short-term memory (SLSTM) and a combination of CNN-LSTM for evaluation and analyses. It was also explained in their research how data imbalance and other issues are dealt with. The dimensionality reduction was carried out through the undersampling technology and the principal component analysis (PCA). The features used in their research included 50 features, such as daily transaction amounts, average transaction amounts, and minimum and maximum transaction amounts, etc. Their research results showed that under the AUC indicator, the CNNs had the best effect. The AUC came to 73–77%, followed by the CNN-LSTM (70–72%) and the SLSTM (65–72%). They believed that the transactional features they proposed could be obtained from most financial transaction models. Short-term financial transactions were mainly analyzed in their models, as typical fraudulent activities often occurred shortly after credit cards had been used for abnormal transactions. Based on a CNN method, Fu et al. [16] also proposed a relatively novel method by converting features and data in multiple time intervals into heat maps and using the CNNs for training. Therefore, unlike traditional CNNs, it did not use image classifications but included time information in the matrix. As it included time intervals (a day, two days, one week, one month and all transactional information within the time intervals), the behavior patterns of the transactions could be identified through the matrix and the prediction could be made through the CNNs. The scholars also explained that the CNNs could be used to avoid overfitting the model. Compared with the support vector machine (SVM), the random forests (RFs) and the neural networks (NNs), the results of real transactional data in commercial banks in their research showed that the proposed CNNs had a better performance than other existing methods.

After the research of Fu et al. [16], another group of scholars also used the CNN method for detecting abnormal credit card transactions. Zhang et al. [17] proposed to add a so-called feature sequencing layer before inputting data into a CNN model, because they believed that the order of features before the data was inputted into the CNN model would affect the results of the convolutional layer, the pooling layer and the final model. Subsequent experimental results also proved their hypothesis. The feature sorting layer also served as a feature selection function. Different sorting combinations allowed each feature to form a higher weighted or more important advanced feature after passing through the convolutional layer and the pooling layer. In this research, a one-dimensional matrix was used in a CNN model, which greatly reduced the calculation time and preprocessing time of the models. The final results showed that its performance was better than that proposed by K. Fu et al. with the recall of 94%.

In the research mentioned above, only Fu [16] and Zhang [17] consider real time detections. Their proposed detection frameworks are similar. The difference is that the latter adds a feature sorting layer, which rearranges the order of the features according to the different training data before inputting the data into the CNN model to optimize it. Abakarim et al. [18] explored the possibility of real-time detection in their research. They believed that although there were many solutions using machine learning, there were few studies using deep learning. According to them, many of these studies had not considered the importance of applying real-time methods to solve problems such as abnormal credit card transactions. Therefore, they proposed a real-time credit card fraud detection system based on the deep neural network technology. They proposed a model based on the auto-encoder, which included an encoder and a decoder. The encoder was used to compress inputs to the greatest extent possible, whilst the decoder rebuilt the compressed inputs, so that the autoencoder had a neural network which equaled the inputs and the outputs. The auto-encoder could instantly classify credit card transactions as legitimate or fraudulent.

The auto-encoder was also used in another study. Pumsirirat and Yan [19] believed that there were no set patterns for fraudulent activities, so unsupervised learning methods were required. Patterns of fraudulent activities changed rapidly because scammers imitated normal behaviors of ordinary consumers. The auto-encoder, known as AE and an unsupervised learning algorithm, which was trained by setting the input values equal to the output values and applying the backpropagation algorithm. Pumsirirat and Yan [19] compared the auto-encoder with the restricted boltzmann machine (RBP), focusing on cases where fraudulent activities could not be found from previous transactions or by supervised learning. The auto-encoder model found anomalies in these cases by reconstructing the transactions. In addition, Rushin et al. [20] used the auto-encoder as an unsupervised feature engineering method. They explored the impact of two feature engineering methods, the auto-encoder and domain expertise, on their models. They also used three supervised classification models and six different feature sets for comparison. The results showed that the use of domain expertise in building features could greatly improve the prediction performance of the models, whereas even though the auto-encoder only slightly improved the prediction performance, it could reduce the dimensionality of the data. Out of the three classification models, i.e. the logistic regression (LR), the gradient boosted tree (GBDT) and the deep learning models, the deep learning model accounted for four of the highest AUCs in six difference feature sets. This demonstrates again the feasibility of using deep learning models for fraud detection.

3 Methodology

The research framework of this paper is shown in Fig. 2. The original data in this research comprised the game records provided by a online game company. Features were then extracted from the original game data. After the

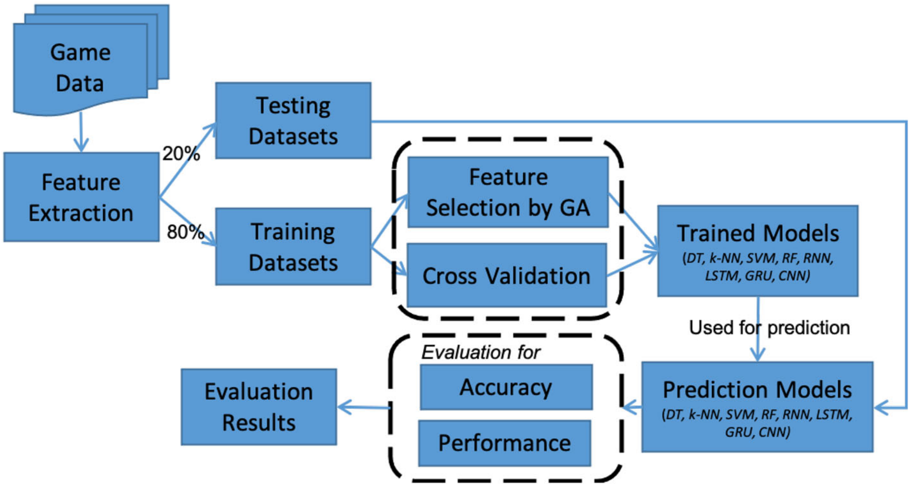


Fig. 2 The research framework of this research

Table 2 Symbols of features and their descriptions

Symbol	Type	Description
T_{time}	Transaction	The difference in time between an account registration date and a top-up time
$T_{billing}$	Transaction	The number of successful order requests
$T_{consume}$	Transaction	The number of successful consumptions
T_{coin}	Transaction	The number of acquired game mall coins
T_{price}	Transaction	The amount of top-up value
T_{store}	Transaction	The number of times a user has topped up
$T_{success}$	Transaction	The number of successful purchases
T_{first}	Transaction	Whether a transaction is a user's first top-up
E_{resume}	Event	The number of times a game is resumed
E_{init}	Event	The number of times a game is initialized
E_{login}	Event	The number of times a gamer logs in
$E_{mission}$	Event	The number of tasks completed in a game
E_{role}	Event	The number of game roles created by an account
E_{daily}	Event	The number of daily logins
E_{event}	Event	The number of game events
E_{level}	Event	The number of level-up times
E_{eval}	Event	The number of times a gamer rates a game
N_{3g}	Network	The number of times of mobile network connections
N_{ip}	Network	The number of times of IP connections
N_{wifi}	Network	The number of times of WiFi connections during a game initialization
N_{ios}	Network	The number of times of iOS connections
N_{andrd}	Network	The number of times of Android connections

features were extracted, the data was divided into training datasets and testing datasets. The training datasets were subjected to feature selection or a sampling technology. Traditional machine learning models and deep learning models in the training datasets were then trained. The trained models were evaluated through cross-validation. The best data pre-processing method for each of the models in the testing datasets was selected according to the cross-validation results. Finally, the results of the models in the testing datasets were compared. In this research, the traditional machine learning models were compared with the deep learning models. In addition to evaluating the models' accuracy, their time cost was compared, and their feasibility in practical applications was analyzed. The pros and cons of each model were evaluated and compared. The best model was proposed to predict malicious chargebacks.

3.1 Feature extraction

In this research, 22 features were extracted from the in-game log and the top-up log of an online-game. All features were classified as transaction, event, and network, which were denoted as *T*, *E*, *N* respectively as shown in Table 2. For better prediction results, the time factor was taken into consideration when establishing features. In addition to extracting the above features, features in different time intervals were separated. The game data was grouped into 14-day periods, starting from the 14th day before the date of a top-up of game credits and ending on the date of the top-up. For example, assuming a top-up of game credits took place at 12 noon on the 15th of a month, the features of this top-up transactions were converted from the game data from 12 noon of the 1st to 12 noon of the 15th of that month. The feature value outputs were contained in a 22 × 14 two-dimensional matrix. The two-dimensional matrix was used in the deep learning model, adding up the feature values of the game data in the 14-day period before the date of the top-up, and outputting them as a one-dimensional vector, i.e. as a 22 × 1 matrix.

3.2 Feature selection

It has been explained in the research in [21, 22] that feature selection is an important part of the data pre-processing in machine learning. Feature selection algorithms can be used to help choose important features among many features. Apart from that, feature selection algorithms can be used to select features that help improve model performance. They can also help increase the speed of model learning. Because if there are many features in datasets, they will slow down learning processes. The research in [6, 23] also shows that in the practical applications of machine learning or data mining, problems of imbalanced data or category imbalance are often encountered. Assuming that datasets contain two categories of results, and the number of results in one of the categories in the training sample is much larger than the other category, it will cause an imbalance in the datasets. In these circumstances, the performance of the models will be less accurate than expected, as the model will be biased towards the category contained the larger number of results and incorrectly classify the targeted category of the smaller number of results as the category of the larger number, reducing its accuracy. Feature selection can also be used to deal with problems of imbalanced datasets, as mentioned above. Feature selection is to find features that are highly related to results among all features, and these selected features enable a model to achieve better performance.

In the previous research, a feature selection method based on a genetic algorithm was proposed [24], which was applied on four machine learning algorithms, i.e. the decision trees (DT), the k-nearest neighbor algorithm (kNN), the support vector machine (SVM), and the random forests (RF), to detect malicious chargebacks in online games. The feature selection method proposed in this research converted *n* feature values into binary expressions by establishing and using F1-Measure as the fitness function in the genetic algorithm. In the genetic manipulation, 10 sets of chromosomes were randomly generated, and then the roulette selection method was used to select chromosomes. Two sets of offspring were taken out and configured with a random mating rate and a fixed mutation rate (10%) and repeated 200 generations. Finally, the best feature value combination was used as the output. Compared with existing methods, such as the information gain, the information gain ratio, the gini decrease, the reliefF, and the FCBF, the method proposed in this research showed that feature selection based on the genetic algorithm could increase the F1 scores of each machine learning model by 7–20% [24].

3.3 Deep learning models for chargeback detection

In this research, after the data was pre-processed, it entered the model training stage. The four machine learning models, i.e. the decision trees (DT), the k-nearest neighbor algorithm (k-NN), the support vector machine (SVM) and the random forests (RF), used in the research [24] were used in the training and prediction in this research. Comparisons and evaluation of the effectiveness of deep learning models in this research is discussed in this section.

3.3.1 Recurrent neural networks (RNNs)

The recurrent neural networks (RNNs) [25] are extension models of feedforward neural networks (FNNs) [11, 26]. After a FNN calculates outputs in a single layer, it inputs them into the next layer in a single direction only. That is to say that inputs and outputs of a single layer are independent. After calculating outputs of the same layer, an RNN can return values stored in the hidden layer to itself as inputs of the hidden layer. This is known as the Elman network. An RNN can also return outputs to itself as inputs of the hidden layer. This is known as the Jordan network. Regardless of which network, a recurrent network is formed in a network structure. Therefore, a RNN can handle inputs with variable length sequences. Taking the Elman network as an example, if the given sequence is $\mathrm{x} = (\mathrm{x}_1 x_2 \ldots \mathrm{x}_T)$, the RNN's hidden state ht is: $x = (x_1 x_2 \cdots x_T)$, the RNN's hidden state h_t is:

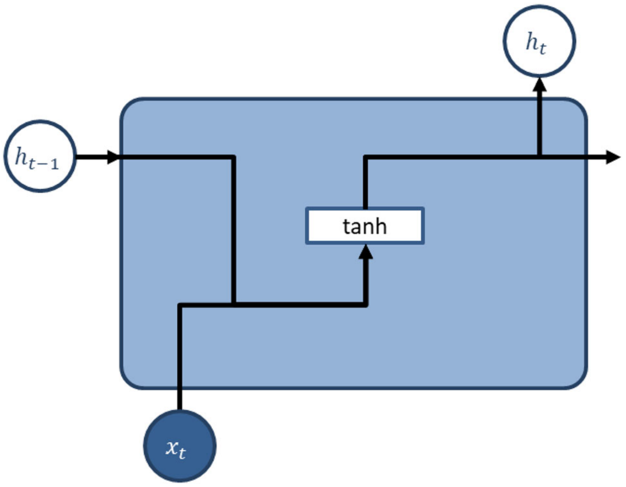


Fig. 3 The unit structure of RNN

$$h_t = \begin{cases} 0, & t = 0 \\ \phi(h_{t-1}x_t), & otherwise \end{cases} \tag{1}$$

Traditionally, the above formula is implemented as below, where g can be s-function or hyperbolic function:

$$h_t = g(Wx_t + Uh_{t-1}) \tag{2}$$

The common RNN structure is shown in Fig. 3:

However, the RNNs have a flaw that it is relatively difficult to learn a long-term time memory. In other words, the earlier the information is in the sequence, the less impact it has on the subsequent decision-making process, and as more time elapses, the less impact the earlier information has on the subsequent decision-making process. This effect is called the gradient disappearance. To solve the problem of the gradient disappearance, German scholars, S. Hochreiter and J. Schmidhuber [27] propose the long short-term memory networks (LSTMs).

3.3.2 Long short-term memory networks (LSTM)

The LSTM [27] is a special type of RNNs that can learn long-term dependencies. It contains a memory unit that maintains its state over time. It differs from the RNNs, in that the LSTM units in the hidden layer are replaced by memory units, and its memory units are controlled by three gates. The LSTM structure is shown in Fig. 4 [28], including an input gate, a forget gate and an output gate. The input gate controls whether inputs can enter the memory units, the forget gate determines whether to reset the memory units, and the output gate decides whether to output the state of the memory units to the output layer.

The input gate formula is:

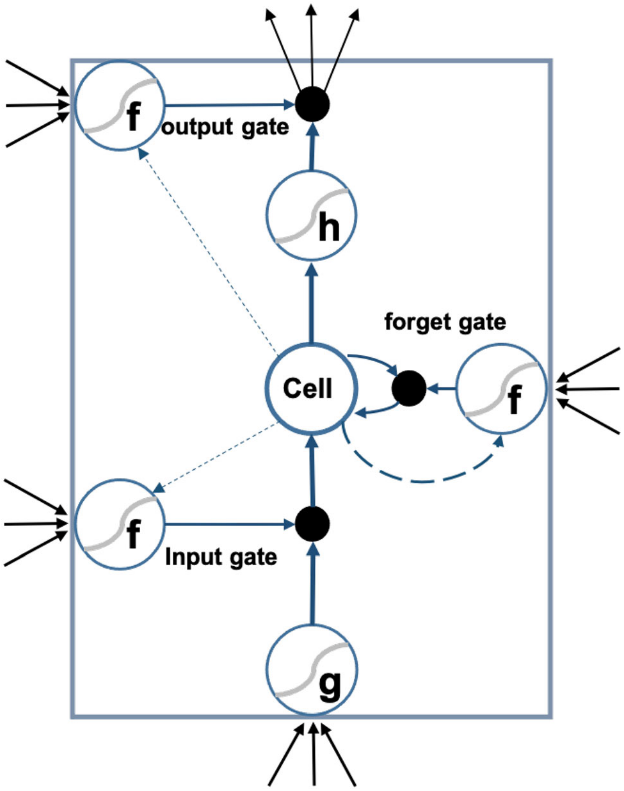


Fig. 4 The unit structure of the LSTM

$$b_\iota^t = f\left(\sum_{i=1}^{I} w_{i\iota}x_i^t + \sum_{h=1}^{H} w_{h\iota}b_h^{t-1} + \sum_{c=1}^{C} w_{c\iota}s_c^{t-1}\right) \tag{3}$$

The forget gate formula is:

$$b_\phi^t = f\left(\sum_{i=1}^{I} w_{i\phi}x_i^t + \sum_{h=1}^{H} w_{h\phi}b_h^{t-1} + \sum_{c=1}^{C} w_{c\phi}s_c^{t-1}\right) \tag{4}$$

The memory unit state formula is:

$$s_c^t = b_\phi^t s_c^{t-1} + b_\iota^t g\left(\sum_{i=1}^{I} w_{ic}x_i^t + \sum_{h=1}^{H} w_{hc}b_h^{t-1}\right) \tag{5}$$

The output gate formula is:

$$b_\omega^t = f\left(\sum_{i=1}^{I} w_{i\omega}x_i^t + \sum_{h=1}^{H} w_{h\omega}b_h^{t-1} + \sum_{c=1}^{C} w_{c\omega}s_c^{t}\right) \tag{6}$$

The memory unit output formula is:

$$b_c^t = b_\omega^t h\left(s_c^t\right) \tag{7}$$

The problem that the RNNs cannot learn the long-term time memory can be improved using the LSTM. Through the hidden layer of the LSTM and its memory units, the information at the first point of time can be stored well, and the information can be transmitted to required outputs, so that the network maintains a relatively long short-term memory capacity.

3.3.3 Gated recurrent units (GRUs)

Although the LSTM can solve the long-term time memory problem of the RNNs, the LSTM takes a relatively long time to execute. Therefore, in 2014, Korean scholars, D. Bahdanau et al. [29] proposed gated recurrent units (GRUs). GRUs can be used to accelerate the speed of execution and reduce the amount of the memory used. The structure of a GRU is shown in Fig. 5, which is similar to the LSTM, but something known as the update gate in a GRU replaces the input gate and forget gate in the LSTM. It also combines the unit state and the hidden state. The calculation formulas in the GRUs are therefore different from the LSTM. In terms of the number of the gates, the GRUs have one less gate than the LSTM, and as such, they have more advantages than the LSTM in terms of calculation and time.

The formula for the update gate is:

$$z_t = \sigma(W_z \cdot [h_{t-1}x_t]) \tag{8}$$

The formula for the reset gate is:

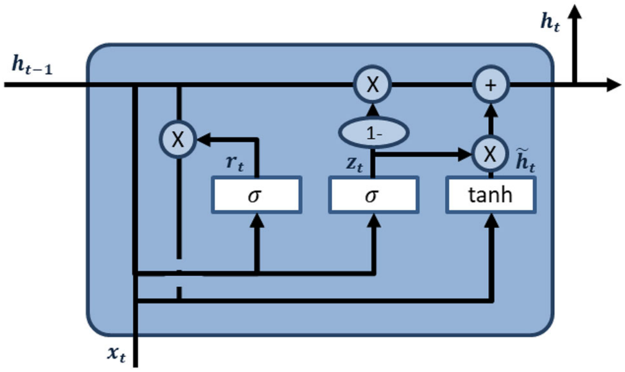


Fig. 5 The unit structure of GRU

$$r_t = \sigma(W_r \cdot [h_{t-1}x_t]) \tag{9}$$

The formula for the current hidden state is:

$$\tilde{h_t} = tanh(W \cdot [r_t * h_{t-1}x_t]) \tag{10}$$

The formula for the output hidden state is:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h_t} \tag{11}$$

3.3.4 Convolutional neural networks (CNN)

Convolutional neural networks, proposed by LeCun et al. [30], are also extension models of the FNNs. CNNs have become successful models for solving classification problems. The CNN models are also suitable for training a large amount of data. There is a mechanism in the CNNs to avoid model overfitting [16]. In general, The CNNs mainly consist of two parts [15], the feature extraction part and the classification part. The feature extraction part is consist of one or more convolution layers and pooling layers. It then carries out classification processes after features are passed through a fully connected layer. The convolutional layer is used to extract the relationship between adjacent pixels, and find important features, such as the boundaries of objects in the pictures, and so on. The pooling layer is used to greatly reduce the dimension of data and suppress noises. Finally, the fully connected layer flattens the results of the previous convolution and pooling layers and then inputs themto a general neural network for further calculation. A general architecture of a CNN is shown in Fig. 6.

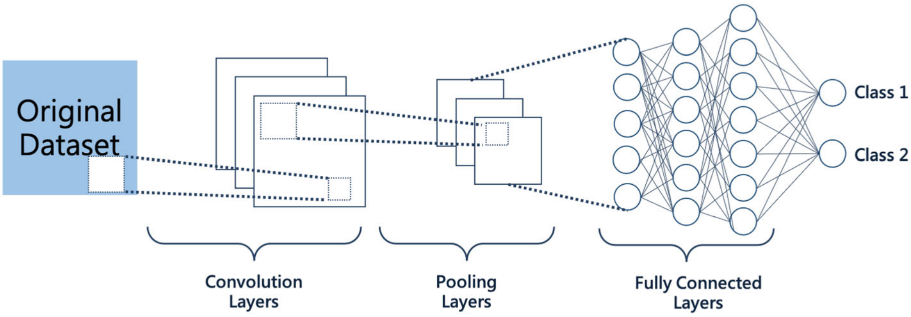
4 Experiments and model evaluation

4.1 Model evaluation

4.1.1 Accuracy evaluation

The models in this research were evaluated through Matthews correlation coefficient (*MCC*) and F1-Measure (*F*1). In the process of evaluating abnormal events, the largest

Fig. 6 A general architecture of CNN



number of abnormal events are usually predicted with the highest accuracy [3]. The main explanation is that two indicators must be considered when predicting abnormal events: precision and recall. Their formulas are as follows, where TP represents true positive, FP false positive, and FN false negative:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

Precision is used to gauge whether models have misclassified abnormal events. That is, whether they have classified non-abnormal events as abnormal events. In this research, Precision was used to check whether legitimate top-ups of game credits had been treated as malicious chargebacks. Recall is used to gauge whether abnormal events have been missed, i.e. abnormal transactions have been classified as normal events. In this research, recall was used to check whether malicious chargebacks had been classified as legitimate top-ups of game credits.

$F1$ is the harmonic average of the precision and recall rates. It also includes the precision and recall rates of a classifier model, so its scores are used as evaluation indicators. The formula of $F1$ is:

$$F1 = 2 * \frac{Precision * Recall}{Precisiom + Recall} \tag{14}$$

D. Chicco et al. believed that when evaluating binary models, MCC was better than accuracy and $F1$. They believed that MCC could produce a better reference and more authentic assessment than the other two [31], the reasons being that the other two might produce misleading results on unbalanced datasets. Their research proved they were correct in that MCC could only produce high scores when a binary model correctly predicted most of the true positives and most of the true negatives. Therefore, MCC scores were used in this research as the main evaluation model indicators. The range of MCC scores was [− 1, +1], which was negative 1 and positive 1 in cases of complete error and perfect classification respectively. The formula of MCC is:

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \tag{15}$$

4.1.2 Performance evaluation

After each model was determined in this research, the time used for program execution by each model in pre-processing, training, and data prediction was measured and compared to determine which model achieved higher efficiency in terms of performance and time. Below is the formula for evaluating the efficiency of the models:

$$PE = \frac{MCC}{\text{Computing Duration}} \tag{16}$$

In order to have more consistent evaluation results, all evaluations were performed with the same hardware and operation system. The detail descriptions of the experimental environment and related package versions are shown in Table 3.

Table 3 The experimental environment of a hardware, an operation system, and a software library used by python

Item	Specifications
CPU	Intel Xeon Silver 4110 8C 2.1GHz *2
GPU	NVidia RTX 2080 Ti
RAM	384 GB
System OS	Linux 4.12.14–197.45 × 64
Python	3.6.9
Scikit-learn	0.24.1
Pandas	1.1.5
Tensorflow-gpu	2.2.2
Scipy	1.5.4
Imbalanced-learn	0.8.0
Numpy	1.18.5

4.2 Evaluation results

The data used in this research was the actual game data obtained from a game company in Taiwan. The game company had already published quite a few games in Taiwan. The game company provided the data from a game for this research, including the number of times of the game played, top-ups, chargebacks, blacklist accounts and delisted blacklist accounts, and the registration dates of game accounts between April 1, 2019 and March 31, 2021. These records are shown in Table 4. During this period, a total of more than 67,000 accounts were registered, there were over 260,000 valid top-ups, and over 550,000,000 times of the game played. Even though we obtained a large number of records from the game company, the actual data we could use was from the 54 blacklist users out of the 7087 valid top-ups that had been blocked out of log-ins.

As to the data pre-processing, how to label the data and establish features from the original datasets is first discussed in this paper. 88 out of 96 users of the blacklist accounts were marked out, 54 of which were blocked by the game company from logging into their accounts.

Table 4 The game data provided by the game company

	Top-ups	Chargebacks	Blacklist accounts	Delisting blacklist	Players
Total records	694,520	582	96	18	67,316
Valid users	269,124				7087
Blocked users	3243	435	61	10	54

Blocking a user from logging into his/her account was the most severe punishment the game company could impose on a user for his/her malicious chargeback behaviors. The dates of the registration of 3 of the 54 users were missing, causing the features for these three accounts being omitted. The data of these three accounts was therefore excluded from this research. 9 of the remaining 51 users applied for delisting from the blacklist and the amounts of chargebacks were returned to them. The transactions of these 9 users were therefore treated as legitimate/normal top-up transactions. The chargeback transactions of the rest of the 42 users were treated as malicious chargebacks. After marking out the transactions of the 51 users, there were 230 malicious chargeback transactions and 1469 legitimate/normal transactions, totaling 1699 transactions.

After the features were created, the datasets were divided in to two parts, 80% of which were used in training and 20% in testing. The datasets for training were used for feature selection or sampling, as discussed below. Each pre-processing was cross validated by the training datasets. The cross-validation was performed using 5-folds. According to the results of the cross-validation, the best data pre-processing methods were selected to apply on each model. 20% of the testing datasets were pre-processed according to the selected processing methods, and then the results of the models in the testing datasets were compared. After the datasets were segmented, there were 1359 training datasets and 340 testing datasets.

From Table 4, it is observed that there were imbalanced datasets. In order to handle the imbalanced datasets, the sampling method was used to solve the imbalance datasets. The sampling method can be roughly classified into oversampleing [32, 33], under-sampling [34, 35] and comprehensive sampling [36]. The oversampling method is to increase the number of data of a minority class through a sampling algorithm. In contrast, the undersampling method is to reduce the number of the majority class through a sampling algorithm. And the comprehensive sampling method is the simultaneous use of both oversampling and undersampling, which not only increases the number of minority classes but also reduces the number of majority classes. The synthetic minority over-sampling technique (SMOTE) [37] is the most popular sampling method to handle imbalance datasets. The SMOTE is an oversampling technique that increases the amount of data by synthesizing data in a few categories. Many studies are based on SMOTE extensions or improvements. The SMOTE method for sampling was therefore used in this research.

4.2.1 Results of feature selection

After the features were extracted and the datasets were divided, the best feature combinations of each model were selected through the feature selection method based on the genetic algorithm proposed in Sect. 3.2, The results were compared with those without performing feature selection. The fitness function in the genetic algorithm was substituted with the *MCC* evaluation method in this research. The feature selection results of each learning model are shown in Table 5. As seen in Table 5, three features, N_{3g}, T_{time} and N_{ip} are all selected by 8 models after feature selection. Take the feature N_{3g} as an example, 137 top-ups are malicious chargebacks, and 160 top-ups are normal. According to the results, users who carry out maliciously chargeback activities rarely play games on mobile networks. Furthermore, the feature, T_{time}, which refers to the difference in time between an account registration date and a top-up time, shows that the average T_{time} of malicious chargeback users is 155.38 days, however that average T_{time} of normal users is 332.48 days. This means that malicious chargeback users usually top-up within a short period of time after registering an account, while normal top-up users usually play games for a while before making top-up transactions. In addition, E_{eval} is not adopted in the models. This may indicate that there is no significant difference in the behaviors of malicious chargeback users and normal top-up transaction users in the rating of a game.

4.2.2 Results of accuracy and performance evaluation

After each model was trained by the training datasets, the testing datasets were processed by the best pre-processing methods, including the GA-based feature selection and the SMOTE method. The comparison results of the *MCC* scores of the testing dataets are shown in Fig. 7. The GRU model obtains the highest score of 0.97 among all the models, followed by LSTM of 0.944, DT of 0.928 and k-NN of 0.918. The lowest score of 0.772 is obtained by SVM.

As seen in Fig. 7, the features proposed in this research for deep learning (a 22 × 14 two-dimensional matrix) can be used to effectively improve the scores of deep learning

Table 5 The selected features by the best feature combination for each model after feature selection

Feature	DT	k-NN	SVM	RF	RNN	LSTM	GRU	CNN	Count
N_{3g}	✔	✔	✔	✔	✔	✔	✔	✔	8
T_{time}	✔	✔	✔	✔	✔	✔	✔	✔	8
N_{ip}	✔	✔	✔	✔	✔	✔	✔	✔	8
E_{resume}	✔	✔		✔	✔	✔	✔	✔	7
E_{init}	✔	✔	✔	✔	✔		✔	✔	7
E_{login}		✔	✔	✔	✔	✔	✔		6
$E_{mission}$		✔	✔	✔	✔	✔	✔		6
E_{role}	✔	✔	✔	✔	✔			✔	6
E_{daily}	✔		✔		✔	✔	✔		5
$T_{billing}$			✔	✔		✔	✔	✔	5
E_{event}	✔	✔				✔	✔	✔	5
N_{wifi}			✔	✔	✔	✔			4
$T_{consume}$	✔	✔				✔		✔	4
E_{level}			✔		✔	✔	✔		4
N_{ios}			✔	✔		✔		✔	4
T_{coin}	✔			✔			✔		3
T_{price}				✔	✔		✔		3
T_{store}	✔					✔	✔		3
$T_{success}$			✔		✔			✔	3
T_{first}					✔	✔	✔		3
N_{andrd}					✔	✔			2
E_{eval}									0
Count	11	10	13	13	15	16	15	11	-

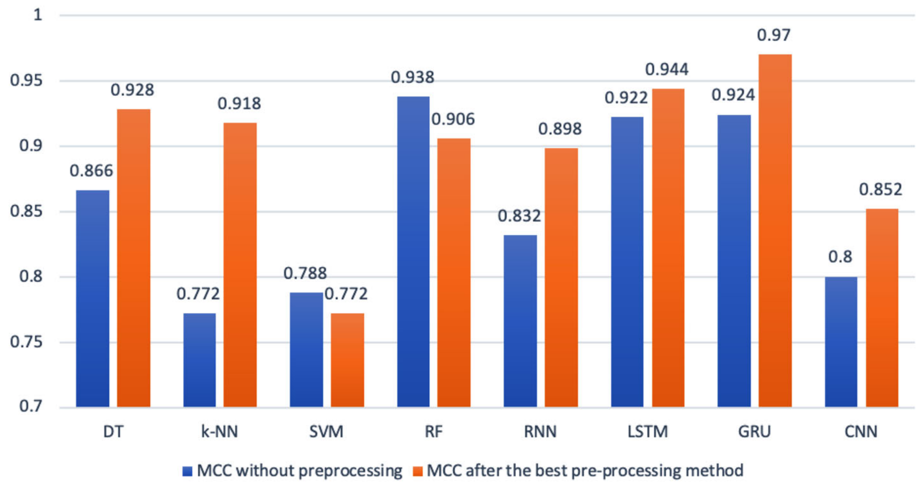


Fig. 7 A comparison of *MCC* scores with and without the best pre-processing methods

models when combined with the research process of this research. The scores of the four deep learning models are improved by 2% to 8%, and the final scores of the GRU and the LSTM models are higher than those of the machine learning models.

The $F1$ scores of the final test results in this research are shown in Fig. 8. The performances of the deep learning models on the $F1$ scores are also in line with the *MCC* scores. The $F1$ scores of the GRU and the LSTM are still higher than all machine learning models. Compared with the original samples, the scores of the four deep learning models are improved by 2% to 7%. It also proves once again that the features and process proposed in this research can be used to effectively improve the scores of deep learning models.

As to the machine learning, the *MCC* score of the SVM is reduced from 0.788 to 0.772, compared with the original samples, whereas the $F1$ score of the SVM rises slightly

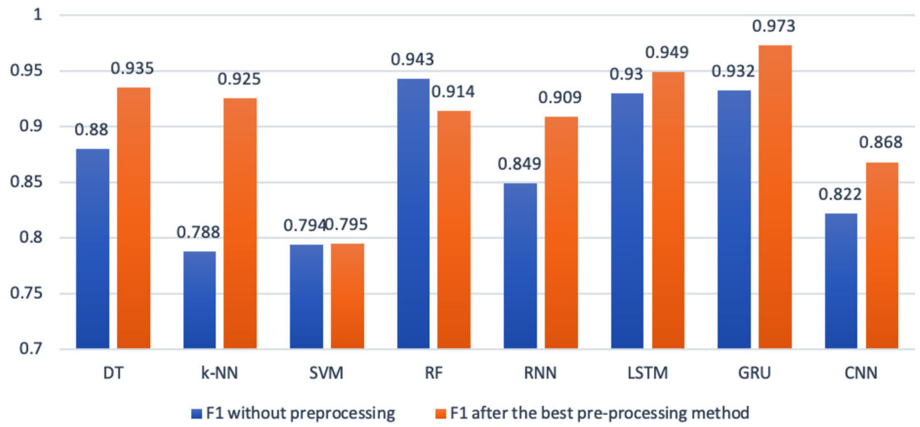


Fig. 8 A comparison of $F1$ scores with and without the best pre-processing methods

from 0.794 to 0.795. These are interesting results. *MCC* can, on the other hand, strike a balance between true positives and true negatives. Only when most of the true positives and most of the true negatives are correctly predicted can a higher *MCC* score be generated. It can also avoid using accuracy on imbalanced datasets to produce overly optimistic scores. This is the reason why the *MCC* scores were selected as the main evaluation indicators in this research. *MCC* can provide a better reference and more authentic scores.

During the experiments in this research, the training time and prediction time required by different models with and without the best pre-processing methods were used to analyze the efficiency of each model. The results are shown in Table 6. All the time units in the Table are in seconds.

As seen in Table 6, the training time of the DT and CNN models after the samples have been processed by the best pre-processing method of feature selection is shorter than the training time without processing, the reasons being that feature selection can effectively reduce data dimensions, enabling the models to learn more quickly. In the cases of the RNN and GRU models, where only the sampling method is used, the training time significantly increases because the number of the samples increases after sampling. Both pre-processing methods are used in the kNN, SVM, RF and LSTM models. As their training time is obviously affected by the sampling method, the training time is still longer than that without processing.

In addition, it can be found that the training time and prediction time required for deep learning are significantly longer than the time required for machine learning. Therefore, if the model efficiency evaluation equation, *PE*, (in Equation 16) is used for evaluation, it is found that the best model at this time is no longer the GRU but the DT, even the GRU has the highest MCC score. The ranking is shown in Table 7. In terms of the total number of seconds

Table 7 The efficiency ranking of the models in the experiments

Rank	Model	*MCC*	*PE*
1	DT	0.928	175.170
2	k-NN	0.918	34.643
3	SVM	0.772	5.204
4	RF	0.906	2.763
5	CNN	0.852	0.114
6	GRU	0.970	0.105
7	LSTM	0.944	0.099
8	RNN	0.898	0.051

Table 6 The training time and prediction time of the models in seconds

Model	Without preprocessing			With the best pre-processing method		
	Training	Prediction	Total	Training	Prediction	Total
DT	0.0086	0.0001	0.0087	0.0052	0.0001	0.0053
k-NN	0.0003	0.0245	0.0248	0.0102	0.0163	0.0265
SVM	0.0353	0.0149	0.0502	0.1195	0.0289	0.1484
RF	0.2445	0.0133	0.2578	0.3141	0.0138	0.3279
RNN	11.7132	0.2029	11.9161	17.431	0.1979	17.6289
LSTM	7.377	0.4678	7.8449	8.8744	0.6628	9.5371
GRU	6.2079	0.3829	6.5908	8.6543	0.5998	9.2542
CNN	12.2191	0.2359	12.455	7.217	0.2315	7.4485

required to turn the models from training to prediction in the experiments, the DT is the most efficient model.

From a practical point of view, the views of N. Lee et al. [3] in solving malicious chargebacks in online games in South Korea was mentioned earlier in this paper. Their research discussed the time cost of using the models in actual scenarios. They used the model prediction time of a single transaction to gauge whether the time cost might be increased based on average daily transaction volumes. Trial calculations were also conducted in this research based on this approach. There were 269,124 valid transactions of game credit top-ups and an average of 369 transactions of game credit top-ups a day in a two-year period in this research. The processing time for each game credit top-up transaction was 223 seconds a day. Taking the LSTM as an example. It took 0.6628 seconds to predict 340 test sets, the longest time for prediction in the experiments, meaning 0.00195 seconds for a single prediction. Even if the process of establishing the features for game credit top-up transactions takes about 7.48 seconds per transaction, there is ample time for it to execute a prediction. As seen from the example of the LSTM, none of the models in this research significantly increased the time cost. Therefore, it was recommended to use the GRU model in this research, which had a higher evaluation score, to prevent malicious chargebacks in online games.

5 Conclusion and future work

Following the development of the technology, more and more users join online games. Not only the number of people join games has increased, the number of users who are now willing to pay for games has also gradually increased. This results in considerable gains in the game industry. Due to the rapid development of the technology, privacy issues have also appeared one after another. To protect users' privacy, online game companies are not allowed to identify users who have made chargeback requests on game payment platforms, causing the game companies not being able to recall services or items sold. Scammers take advantages of this mechanism to commit malicious chargebacks. This causes huge losses to online game companies. These are major challenges facing online game companies these days.

This research is dedicated to solving the above-mentioned problems. The following results and contributions have been obtained:

1. The game data is analyzed to establish features of the deep learning models to prevent malicious chargebacks in online games, enabling online game companies to understand the differences in behaviors between malicious chargeback users and normal top-up transaction users according to the user behavior trajectories in games.
2. It is proposed in this research to use deep learning on the prevention of malicious chargebacks in online games and online chargeback processes. Deep learning shows good MCC results, which are improved by about 2% to 8%. Among all deep learning models, the GRU obtains the highest MCC score in our research, reaching 0.97, followed by the LSTM at 0.944, the decision trees at 0.928 and the k-NN at 0.918. The scores of the GRU and the LSTM in the experiments are also higher than those of the traditional machine learning models in the past.
3. In terms of practical feasibility, although the deep learning models are not as efficient as the machine learning models, they are still competent when applied in real cases. The method proposed in this research can still help prevent malicious chargebacks in online games, minimizing losses caused to online game companies.

In the future, there are many areas in this research that await further research and exploration, such as the collection and use of data. Although there is a large amount of data collected for this research, the amount of data that can be actually used as features in this research is low. Apart from continuing to collect data for the research in the future, 32 status indicators that have not yet been included in this research can be added to discover more malicious chargeback users' behavior trajectories and further reduce malicious chargebacks. The datasets provided for this research was limited. Only the data of a single game was used in this research. If datasets collected from various games can be included in the research in the future, more general models for a variety of games can be developed to prevent malicious chargebacks to minimize potential losses that may be caused to the online game industry. Finally, due to the length restriction, the imbalance datasets have not been investigated and evaluated in detail. In the future, we intend to compare the results under oversampling, undersampling, and comprehensive sampling to make the research more complete.

Acknowledgements This research was partially funded by Ministry of Science and Technology (No. 110-2637-H-027-004-). We would like to thank the anonymous reviewers for their valuable feedback. We are also grateful to Lien Wang and another anonymous individual for proofreading the manuscript.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by YCW, YXL and MEW. The first draft of the manuscript was written by YCW and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work was partially supported by Ministry of Science and Technology, Taiwan (Grant Numbers 110-2637-H-027-004-).

Data availability The datasets generated during and/or analysed during the current study are not publicly available due to the trade secrets consideration.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Wijman, T.: Newzoo's games trends to watch in 2021, 19 (2019)
2. Intelligence, M.: Gaming industry–growth, trends, and forecast (2020–2025) (2020)
3. Lee, N., Yoon, H., Choi, D.: Detecting online game chargeback fraud based on transaction sequence modeling using recurrent neural network. In: International Workshop on Information Security Applications, pp. 297–309. Springer
4. Awoyemi, J.O., Adetunmbi, A.O., Oluwadare, S.A.: Credit card fraud detection using machine learning techniques: a comparative analysis. In: 2017 International Conference on Computing Networking and Informatics (ICCNI), pp. 1–9. IEEE
5. Seo, J.-H., Choi, D.: Feature selection for chargeback fraud detection based on machine learning algorithms. Int. J. Appl. Eng. Res. **11**(22), 10960–10966 (2016)
6. Chen, X.-W., Wasikowski, M.: Fast: a roc-based feature selection metric for small samples and imbalanced data classification problems. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 124–132
7. Carneiro, N., Figueira, G., Costa, M.: A data mining based system for credit-card fraud detection in e-tail. Decis. Support Syst. **95**, 91–101 (2017)
8. Mao, H., Liu, Y.-W., Jia, Y., Nanduri, J.: Adaptive fraud detection system using dynamic risk features. arXiv:1810.04654 (2018)
9. Tedim, M.D.S.: Predicting fraud behaviour in online betting. Thesis (2019)
10. Pandey, Y.: Credit card fraud detection using deep learning. Int. J. Adv. Res. Comput. Sci. **8**(5), 981–984 (2017)
11. Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., Beling, P.: Deep learning detecting fraud in credit card transactions. In: 2018 Systems and Information Engineering Design Symposium (SIEDS), pp. 129–134. IEEE
12. Wiese, B., Omlin, C.: In: Bianchini, M., Maggini, M., Scarselli, F., Jain, L.C. (eds.) Credit card transactions, fraud detection, and machine learning: modelling time with LSTM recurrent neural networks, pp. 231–268. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04003-0_10
13. Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.-E., He-Guelton, L., Caelen, O.: Sequence classification for credit-card fraud detection. Expert Syst. Appl. **100**, 234–245 (2018)
14. Najadat, H., Altiti, O., Aqouleh, A.A., Younes, M.: Credit card fraud detection based on machine and deep learning. In: 2020 11th International Conference on Information and Communication Systems (ICICS), pp. 204–208. IEEE
15. Heryadi, Y., Warnars, H.L.H.S.: Learning temporal representation of transaction amount for fraudulent transaction recognition using CNN, Stacked LSTM, and CNN-LSTM. In: 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), pp. 84–89 (2017). https://doi.org/10.1109/CYBERNETICSCOM.2017.8311689
16. Fu, K., Cheng, D., Tu, Y., Zhang, L.: Credit card fraud detection using convolutional neural networks. In: International Conference on Neural Information Processing, pp. 483–490. Springer
17. Zhang, Z., Zhou, X., Zhang, X., Wang, L., Wang, P.: A model based on convolutional neural network for online transaction fraud detection. Secur. Commun. Netw. **2018** (2018)
18. Abakarim, Y., Lahby, M., Attioui, A.: An efficient real time model for credit card fraud detection based on deep learning. In: Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications, pp. 1–7
19. Pumsirirat, A., Yan, L.: Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine. Int. J. Adv. Comput. Sci. Appl. **9**(1), 18–25 (2018)
20. Rushin, G., Stancil, C., Sun, M., Adams, S., Beling, P.: Horse race analysis in credit card fraud—deep learning, logistic regression, and gradient boosted tree. In: 2017 Systems and Information Engineering Design Symposium (SIEDS), pp. 117–121. IEEE
21. Uçar, M.: Classification performance-based feature selection algorithm for machine learning: P-score. IRBM (2020)
22. Claypo, N., Jaiyen, S.: A new feature selection based on class dependency and feature dissimilarity. In: 2015 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), pp. 1–6. IEEE
23. Pant, H., Srivastava, R.: A survey on feature selection methods for imbalanced datasets. Int. J. Comput. Eng. Appl. **9**(2), 197–204 (2015)
24. Lai, Y.-X., Liao, T.-Y., Wu, Y.-S., Wei, Y.-C.: Based on Genetic Algorithm for Feature Selection of Chargeback Fraud Detection in Online Games (2020)
25. Elman, J.L.: Finding structure in time. Cogn. Sci. **14**(2), 179–211 (1990)
26. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555 (2014)
27. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
28. Graves, A.: Long Short-Term Memory, pp. 37–45. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24797-2_4
29. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv:1409.0473 (2014)
30. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
31. Chicco, D., Jurman, G.: The advantages of the Matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC Genomics **21**(1), 1–13 (2020)
32. Mathew, J., Pang, C.K., Luo, M., Leong, W.H.: Classification of imbalanced data by oversampling in kernel space of support vector machines. IEEE Trans. Neural Netw. Learn. Syst. **29**(9), 4065–4076 (2018). https://doi.org/10.1109/TNNLS.2017.2751612
33. Zheng, Z., Cai, Y., Li, Y.: Oversampling method for imbalanced classification. Comput. Inf. **34**(5), 1017–1037 (2015)
34. Aridas, C.K., Karlos, S., Kanas, V.G., Fazakis, N., Kotsiantis, S.B.: Uncertainty based under-sampling for learning Naive Bayes classifiers under imbalanced data sets. IEEE Access **8**, 2122–2133 (2020). https://doi.org/10.1109/ACCESS.2019.2961784
35. Lin, W.-C., Tsai, C.-F., Hu, Y.-H., Jhang, J.-S.: Clustering-based undersampling in class-imbalanced data. Inf. Sci. **409–410**, 17–26 (2017). https://doi.org/10.1016/j.ins.2017.05.008

36. Agrawal, A., Viktor, H.L., Paquet, E.: Scut: Multi-class imbalanced data classification using smote and cluster-based undersampling. In: 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), vol. 01, pp. 226–234 (2015)
37. 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). https://doi.org/10.1613/jair.953

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Yu-Chih Wei is an Assistant Professor in the Department of Information and Finance Management at the National Taipei University of Technology. He holds a Ph.D. in Information Management from National Central University, and a B.S. and a M.S. in Information Management from YuanZe University. His research interests include FinTech security, ISRA, RegTech, VANET security, information security management, and business continuity management. Before pursuing an academic career, Dr. Wei was a researcher at the Information & Communication Security Laboratory of Chunghwa Telecom Co., Ltd.

You-Xin Lai received his B.S. and M.S. degree in Department of Information and Finance Management, National Taipei University of Technology in 2021. His research interests include information security and machine learning.

Mu-En Wu is an Associate Professor at Department of Information and Finance Management at National Taipei University of Technology, Taiwan. Dr. Wu received his Ph.D. degree with major in computer science from National Tsing Hua University, Taiwan, in 2009. After that, he joined Institute of Information Science, Academia Sinica at Taipei City, Taiwan as a postdoctoral fellow during 2009 2014. During February 2014 to July 2017, he served as an assistant professor of Department of Mathematics at Soochow University. He has a wide variety of research interests covering cryptography, information theory, prediction market, money management, and financial data analysis.