A DQN-based Internet Financial Fraud Transaction Detection Method

Xiaoguo Wang
College of Electronics and Information Engineering, Tongji University, Shanghai,, China,
xiaoguowang@tongji.edu.cn

Zeguo Wan*
College of Electronics and Information Engineering, Tongji University, Shanghai, China,
1933042@tongji.edu.cn

Yin Zhang
College of Electronics and Information Engineering, Tongji University, Shanghai, China
1930790@tongji.edu.cn

ABSTRACT

The anti-fraud issue of Internet finance is a hot research topic in the industry. Aiming at the complex fraud problem of Internet finance, this paper proposes a fraudulent transaction detection method based on Deep Q Learning, and constructs a feasible electronic transaction fraud detection model. Based on reinforcement learning, this method makes the agent learn classification strategies, builds the environment with RFM model, and uses SmoothL1 as the loss function to improve the learning efficiency of the agent. The experiment uses a variety of evaluation metrics to verify the performance. The results demonstrated that the proposed DQN-based fraud detection method in this paper has improved some performance evaluation metrics compared with the traditional method.

CCS CONCEPTS

• **Computing methodologies**; • **Machine learning**; • **Learning paradigms**; • **Reinforcement learning**;

KEYWORDS

Internet finance, Fraudulent transaction detection, Electronic transaction

ACM Reference Format:
Xiaoguo Wang, Zeguo Wan*, and Yin Zhang. 2021. A DQN-based Internet Financial Fraud Transaction Detection Method. In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3487075.3487136

1 INTRODUCTION

Benefiting from the rapid development of the Internet, Internet finance based on emerging technologies such as big data has gradually revealed its important position. However, it is inevitable that the corresponding Internet financial frauds such as credit card fraud, financial statement fraud, insurance fraud, etc. have also brought losses to individuals and businesses [1, 2]. As an important part of Internet finance, electronic transactions bring convenience to people's lives, but transaction fraud seriously harms the interests of individuals and society. For Internet finance and electronic transactions, research on fraudulent transaction detection models and technologies is a hot spot in the industry [3-5].

In reinforcement learning, the interaction between agent and environment is a Markov Decision Process (MDP). A Markov decision process is composed of four tuples (S, A, P, R), that is, state, action, state transition function, and reward, and the Markov decision process should have such properties: The impact of actions taken in a state depends only on that state rather than previous history [6]. Reinforcement learning can regard the classification process as a series of decision-making problems, and transform the data into several states to carry out the training of the agent. Through training, the agent can learn the decision-making strategy, so as to improve the accuracy of model classification. On the anti-fraud issue of Internet finance, reinforcement learning has better coping ability for various fraud cases. Using reinforcement learning to carry out anti-fraud research provides a way to solve the problem of anti-fraud.

Based on the Deep Q Learning (DQN) reinforcement learning algorithm, this paper studies the detection method of Internet financial fraudulent transactions, designs and implements a fraudulent transaction detection model, and verifies the model performance on the transaction data set through experiments.

2 LITERATURE REVIEW

2.1 Internet Financial Fraudulent Transaction Detection

Methods based on rule matching, methods based on account transaction behavior features, and methods based on inter-account association are the main researches on Internet financial transaction fraud detection at home and abroad. Among them, the fraud detection method based on account transaction behavior features constructs account transaction behavior features from different angles according to the historical transaction data of accounts, and builds a detection model combined with machine learning and deep learning. The account transaction behavior features can be divided into: Recency Frequency Monetary (RFM) features based on time window, features based on transaction time distribution, features based on IP address, etc [7]. For example, Bashen et al. used the von Mises distribution to create new features and expanded the transaction aggregation strategy to evaluate the impact of different feature sets on fraud detection [8].

CSAE 2021, October 19–21, 2021, Sanya, China

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487136

In recent years, domestic and foreign researches have extensive applications and theoretical support for detection methods based on account transaction behavior features, including K-nearest neighbors (KNN), Bayesian, decision trees, random forest, support vector machine (SVM), neural network, etc [9-11]. On the basis of existing research, Luis et al. [12] introduced reinforcement learning method to credit card fraud transactions detection, combining reinforcement learning with supervised learning and unsupervised learning, effectively improving the training speed and accuracy of the detection model. In view of the cost of acquiring features in some detection scenarios, Jaromir et al. [13] used Double DQN (DDQN), Dueling Architecture and other methods in reinforcement learning to build the model, which significantly reduced the number of parameters, and the model had robustness and scalability.

2.2 DQN Algorithm

The fraud situation faced by Internet finance is complex, and reinforcement learning has the ability to adapt to environmental changes, which can greatly help detect fraudulent transactions. The core of reinforcement learning is the Markov decision process [6]. The main process is carried out in the interaction between the agent and the environment. For a certain state $s \in S$ in the environment, the agent needs to take the corresponding action $a \in A$,so that the environment is transferred to the next state s' according to the state transition function $P(s'|s, a)$, and the reward $r(s, a)$ of environmental feedback is received.

DQN algorithm [14] is a classic algorithm in reinforcement learning. In traditional reinforcement learning methods such as Q_learning and Sarsa, the state transition function P appears in the form of state transition table. The algorithm needs to maintain a table to record the Q value under all combinations of states and actions. However, in some scenarios, the number of states is extremely large or even unpredictable, which makes the algorithm unable to maintain such a large table. DQN algorithm uses a value estimation function to replace the table, introduces the neural network in deep learning, and directly uses the neural network to generate the Q value. In this paper, the classification is based on DQN, the Q value of the fraud judgment action made by the agent on the transaction data is evaluated, and the better solution is taken as the judgment result to improve the accuracy of fraud detection.

The agent training algorithm of DQN is presented as Algorithm 1 in Appendix. In order to achieve the convergence of the value function, DQN uses experience playback [14] to store the state transformation (s_t, a_t, r_t, s_{t+1}) during the learning process. And during the learning process, the memory transformation data (s_j, a_j, r_j, s_{j+1}) is obtained by batch sampling from the memory bank, and the network parameters are updated with the memory data. DQN adopts ϵ-greedy strategy [14] in each learning process, in which the agent will randomly select actions according to a decreasing probability ϵ, so that the agent will try more different possibilities in the initial stage of learning.

3 RESEARCH METHODOLOGY

3.1 Fraudulent Transaction Detection Based DQN

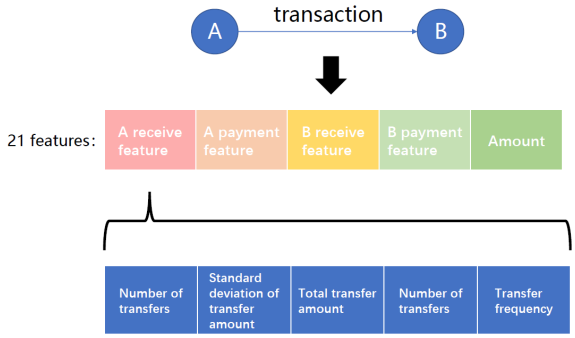

Figure 1: Corresponding Features of Each Transaction.

3.1.1 Environment Construction and Agent Training. For the transaction behavior of each account, the payment and receipt transactions may represent completely different features. As shown in Figure 1. In order to comprehensively consider the relationship between users' historical behavior features and transaction accounts, this paper constructs features based on RFM model. In the RFM model, Recency represents the time of the most recent transaction, which indicates the time between the user's most recent transaction and the present (or as of the statistical period). Frequency is the frequency of transactions, and consumption frequency refers to the number of transactions made by users in the statistical period. Monetary is the transaction amount, and the consumption amount refers to the total amount of transactions made by users during the statistical period [15, 16]. And in this paper, the number of transfers, the standard deviation of transfer amount, the total amount of transfers, the number of transfers and the frequency of transfers are calculated for each transaction account. Then we splice the payment and receipt transaction information of the paying and receiving account in each transaction and add the amount feature of the transaction to obtain the transaction data with 21-dimensional features.

For the data obtained by feature construction, we correspond each data to a state in the environment. As shown in Figure 2. The data containing the features of RFM model is introduced into the environment as a series of states. For each such state, the agent needs to take corresponding actions, that is, to judge whether each transaction record is fraudulent or not.

In this paper, Algorithm 1 is used to train the agent. The state conversion during the learning process is stored. And the memory data is sampled from the memory bank in batches during the learning process, and the neural network is updated according to the memory data. In the training process, in order to update the parameters in time, if the environment finds that the agent is wrong, the round of training is ended and gradient descent is performed to update the model parameters, otherwise the agent continues to use the ϵ-greedy strategy for action selection.

3.1.2 Network Structure. In reinforcement learning, the agent uses deep learning neural network to calculate the Q value, and selects actions based on the Q value. The network structure uses a fully connected layer with 25 units as the hidden layer. The network

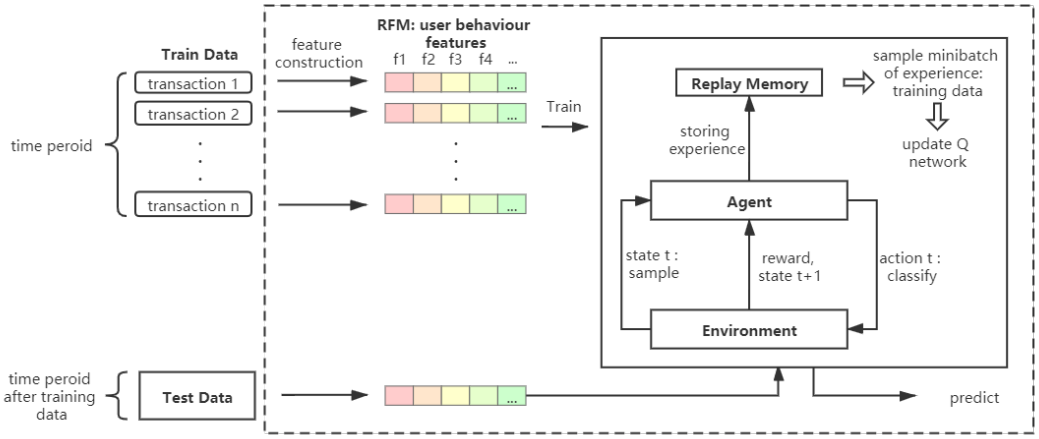

Figure 2: Description of Environment and Agent Training Process.

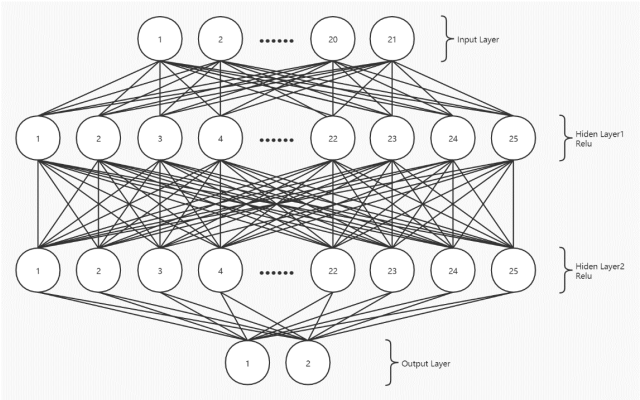

Figure 3: Agent Neural Network Structure.

structure is shown in Figure 3. The activation function used in this paper is Relu function. This is because the Relu function greatly reduces the amount of calculation when deriving the derivation during back propagation and speeds up the training process. And Relu will make the output of some neurons be 0, which causes the sparsity of the network and reduces the interdependence of parameters, thereby alleviating the occurrence of over-fitting problems [17, 18]. The Adam algorithm [19] is selected as the optimizer to update the agent network during the training process, which makes the model training process more stable.

3.1.3 Loss Function. The loss function used in Algorithm 1 is mean square error (MSE). However, in the fraud detection problem, for the case where the distance between some outliers is large, the MSE function may lead to gradient explosion. The SmoothL1 loss function solves the unsmooth problem of the L1 loss function, which is more robust and less susceptible to outliers. It can control the magnitude of the gradient to make the gradient descent more stable. Taking these advantages into account, this article uses the SmoothL1 function instead of the mean square error function.

From the Bayman Equation [6], the value estimation function in Algorithm 1 can be expressed as:

$$Q(s,a) = E[r_t + \gamma Q(s_{t+1}, a_{t+1}) | s_t = s, a_t = a] \quad (1)$$

Then, suppose the target estimated value of the Q function in the training process is $\boldsymbol{y}$:

$$y = r + \gamma \max_{a'} Q(s', a'; \theta') \quad (2)$$

And the overall loss function of this article is as Equation 3):

$$L(\theta) = \begin{cases} 0.5(y - Q(s,a;\theta))^2, if \ |y - Q(s,a;\theta)| < 1 \\ |y - Q(s,a;\theta)| - 0.5, \ otherwise \end{cases} \quad (3)$$

3.2 Comparison Methods

In order to verify the effectiveness of fraudulent transaction detection method based on DQN algorithm on transaction data, this paper selects methods including support vector machine (SVM) [20], random forest [21], KNN (K-Nearest Neighbour) [22] and logistic regression [23] as a comparative measurement for experimentation.

3.3 Data Description

The simulated transaction data set PaySim [24] is used in this paper, which is widely used for the performance evaluation of fraudulent transaction detection methods. The data set contains 6,362,620 transaction data. Among all transactions, 8,213 transactions are fraudulent transactions, and the rest are normal transactions. The step feature in the data set is a time unit. A step corresponds to one hour in reality, ranging from 1 to 743, which corresponds to about one month in reality. The amount feature represents the amount of each transaction. The maximum amount in the data set is 92445517, and the minimum is 0. In addition, there are five types of transaction types: CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER. The fraud transactions included in each transaction type are shown in Table 1. Fraud is the number of fraudulent transactions, and fraudPER represents the proportion of fraudulent transactions in the total.

In the course of the study, considering that there is no payment in the merchant account, the merchant account has no classification

Table 1: Dataset Transaction Type and Corresponding Fraud Ratio

type	fraud	fraudPER	total
CASH_IN	0	0.000000000	1399284
CASH_OUT	4116	0.001839553	2237500
DEBIT	0	0.000000000	41432
PAYMENT	0	0.000000000	2151495
TRANSFER	4097	0.007687992	532909

Table 2: Confusion Matrix

	Positive(1)	**Negative(0)**
Positive(1)	TP	FP
Negative(0)	FN	TN

value in the detection of fraudulent transactions, and the transaction data of the merchant accounts is removed.

This paper uses time as the criterion to divide the data set. From the overall data set, all transfer data with a step value of 201-299 are selected as the training set, and all transfer data with a step value of 718-743 are selected as the test set.

In order to accelerate the convergence of the gradient descent solution, these data also need to be normalized to limit the value range of the attribute to between 0 and 1. The form is shown in Equation 4):

$$X = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4}$$

3.4 Evaluation Metrics

In the fraud transaction detection problem, simply using the detection accuracy rate (OA) cannot accurately reflect the model detection performance. Therefore, this paper uses evaluation metrics based on the confusion matrix (Table 2), such as the recognition rate of fraud samples (Sensitivity), the recognition rate of normal samples (Specificity), and G-mean as the criteria for the comparison test.

$$OA = \frac{TN + TP}{TP + FP + FN + TN} \tag{5}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{6}$$

$$Specificity = \frac{TN}{TN + FP} \tag{7}$$

$$G - mean = \sqrt{Specificity * Sensitivity} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$F1\ Score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{11}$$

4 EXPERIMENT RESULTS

The results of the experiment are shown in Table 3. The fraudulent transaction detection method based on DQN algorithm is superior to traditional fraudulent transaction detection methods such as support vector machine and random forest in Specificity, G-mean, Precision, F1 Score evaluation metrics, and also has better performance in OA index. This comes from the advantages of this method at the data level and algorithm level. At the data level, this method converts the input RFM feature data into states, and learns by way of experience playback after storage, which not only learns the historical transaction features of users, but also breaks the correlation of empirical data. At the algorithm level, the model terminates the training once the fraud samples are misclassified, which improves the model's attention to the fraud samples.

5 CONCLUSION

For the problem of fraudulent transaction detection in Internet finance, this paper proposes a fraudulent transaction detection method based on the DQN algorithm. This method is based on reinforcement learning. Through the interaction between the agent and the environment, the agent can learn the decision-making strategy. With the help of RFM model, the user's historical transaction behavior information is abstracted, and 21-dimensional features are

Table 3: Experimental Results

Metrics	SVM	RandomForest	KNN	LogicRegression	Fraudulent Transaction Detection Based DQN
OA	0.74483	0.71379	0.71034	0.74828	0.73448
Sensitivity	0.73203	0.64762	0.64055	0.73684	0.65888
Specificity	0.75912	0.88750	0.91781	0.76087	**0.94737**
G-mean	0.74545	0.75813	0.76675	0.74876	**0.79007**
Recall	0.73203	0.64762	0.64055	0.73684	0.65888
Precision	0.77241	0.93793	0.95862	0.77241	**0.97241**
F1 Score	0.75168	0.76620	0.76795	0.75421	**0.78552**

constructed based on the user's historical behavior information. In the learning process, the feature data is corresponding to the state in the environment, and the agent is adjusted to learn the parameters in time when there is a misclassification. Within the agent, the SmoothL1 loss function is used to improve the learning efficiency of the agent. The experimental results show that proposed Internet financial fraud transaction detection method based on the DQN algorithm in this paper has a good performance on the PaySim data set, and is superior to the traditional detection methods in terms of accuracy and other evaluation metrics. In future work, we will study how to further improve the loss function to accelerate the convergence of the model, and consider applying the improvement method of the DQN algorithm to the model.

REFERENCES

[1] Wang C, Wang C Q. (2020). An Automated Feature Engineering Method for Online Payment Fraud Detection. Chinese Journal of Computers, 43(10), 1983-2001.

[2] Zhang X, Han Y, Xu W, *et al.* (2019). HOBA: A Novel Feature Engineering Methodology for Credit Card Fraud Detection with a Deep Learning Architecture. Information Sciences.

[3] De Sá A, Pereira A, Pappa G. (2018). A Customized Classification Algorithm for Credit-Card Fraud Detection. Engineering Applications of Artificial Intelligence. 72, 10.1016/j.engappai.

[4] Dhankhad S, Mohammed E, Far B. (2018). Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study. Proceeding of the International Conference on Information Reuse and Intergration IEEE Computer Society. 122-125.

[5] Li Y N. (2018). Application of Neural Network Model in Anti-fraud in Banking Internet Finance. The era of financial technology, 276(8), 24-28 (in Chinese).

[6] Sutton R, Barto A. 1998. Reinforcement Learning: An Introduction. MIT Press,.

[7] Whitrow C, Hand D J, Juszczak P, *et al.* (2009). Transaction aggregation as a strategy for credit card fraud detection. Data Mining and Knowledge Discovery, 18(1), 30-55.

[8] Correa Bahnsen A, Aouada D, Stojanovic A, *et al.* (2016). Feature engineering strategies for credit card fraud detection. Expert Systems with Application, 51(Jun.), 134-142.

[9] He H, Hawkins S, Graco W, *et al.* (2000). Application of Genetic Algorithm and K-Nearest Neighbour Method in Real World Medical Fraud Detection Problem. Journal of Advanced Computational Intelligence and Intelligent Informatics,2000,4(2).

[10] Juliet M, Jonah K. (2018). Credit Card Fraud Detection using Bayes Theorem. International Journal of Computer and Information Technology, 7(4).

[11] Zhang W, Ntoutsi E. (2019). FAHT: An Adaptive Fairness-aware Decision Tree Classifier. Twenty-Eighth International Joint Conference on Artificial Intelligence, {IJCAI-19}

[12] Zhinin L, Chang O, Valencia-Ramos R, *et al.* (2020). Q-Credit Card Fraud Detector for Imbalanced Classification using Reinforcement Learning, 12th International Conference on Agents and Artificial Intelligence.

[13] Janisch J, Tomáš P, & Viliam L. (2017). Classification with Costly Features using Deep Reinforcement Learning. https://arxiv.org/abs/1711.07364v1.

[14] Mnih V, Kavukcuoglu K., Silver D, *et al.* (2015). Human-level control through deep reinforcement learning. Nature 518, 529–533. https://doi.org/10.1038/nature14236.

[15] Ernawati E, Baharin S S K, Kasmin F. (2021). A Review of Data Mining Methods in RFM-based Customer Segmentation. Journal of Physics: Conference Series, 1869, 012085.

[16] Yong H, Mingzhen Z, Yue H. (2020). Research on Improved RFM Customer Segmentation Model Based on K-Means Algorithm. 2020 5th International Conference on Computational Intelligence and Applications (ICCIA), 24-7.

[17] Glorot X, Bordes A, Bengio Y. (2011). Deep Sparse Rectifier Neural Networks. Journal of Machine Learning Research, 15, 315-323.

[18] Krizhevsky A, Sutskever I, Hinton G E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM, 60(6), 84-90.

[19] Kingma D P, Ba J. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[20] Chih-Chung C, Chih-Jen L. (2011). LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2(3, article 27).

[21] Breiman L. (2001). Random Forests[J]. Machine Learning, 45(1)(-), 5-32.

[22] Cover T, Hart P. (2003). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1):21-27.

[23] Hosmer D W, Lemeshow S. (2000). Applied Logistic Regression (2nd. ed.). 91-142.

[24] https://www.kaggle.com/ntnu-testimon/paysim1

APPENDIX

Algorithm 1 DQN Training Agent

Input : Train dataset D = $\{x_1, x_2, \ldots, x_T\}$

1. Initialize replay memory M to capacity N
2. Initialize Episode number K
3. Randomly initialize parameters θ
4. Initialize simulation environment
5. **for** episode k = 1 to K **do**
6. Initialize state $s_1 = x_1$
7. **for** t = 1 to T **do**
8. With probability ϵ select an random action
9. otherwise pick an action: $a_t = argmax_a Q(s_t, a; \theta)$
10. Execute action a_t in environment and observe reward r_t and data x_{t+1}
11. Set $s_{t+1} = x_{t+1}$
12. Store (s_t, a_t, r_t, s_{t+1}) to M
13. Randomly sample minibatch of transitions (s_j, a_j, r_j, s_{j+1}) from M
14. Set $y_j = \begin{cases} r_j, if\ episode\ terminates \\ r_j + \gamma max_{a'} Q(s_{j+1}, a_j; \theta),\ otherwise \end{cases}$
15. Perform a gradient descent step: $L(\theta) = (y_j - Q(s_j, a_j; \theta))^2$
16. **end for**
17. **end for**