**Indonesian Journal of Electrical Engineering and Computer Science**
Vol. 41, No. 2, February 2026, pp. 700~709
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v41.i2.pp700-709 

# Predicting non-performing loans in Vietnam's financial sector: a deep Q-learning approach

**Luyen Anh Do, Huong Thi Viet Pham, Thinh Duc Le, Oanh Thi Tran**
International School, Vietnam National University, Hanoi, Vietnam

**Article Info**

***Article history:***

Received Sep 4, 2025
Revised Jan 3, 2026
Accepted Jan 11, 2026

***Keywords:***

Deep Q-learning
Machine learning
Non-performing loans
Reinforcement learning
Risk management

**ABSTRACT**

Non-performing loans (NPLs) prediction is a very important task in risk management of financial institutions. NPLs often lead to substantial losses when loans are not paid back on time. While traditional machine learning (ML) models have been conventionally exploited for credit risk assessment, they frequently face challenges with handling imbalanced data. To deal with this problem, this paper introduces a novel approach using deep reinforcement learning (DRL), specifically deep Q-learning, to enhance the prediction of NPLs. To verify the effectiveness of the method, we introduce a new dataset comprising 83,732 customer records (each described with 22 key features) from one of Vietnam's largest financial entities. Our method is compared with standard ML techniques such as random forest, decision tree, logistic regression, support vector machine, LightGBM, and XGBoost. Experimental results on this dataset demonstrate that deep Q-learning outperforms these traditional models in handling imbalanced data and boosting prediction accuracy. This research highlights the potential of DRL as a robust risk management tool, helping financial institutions make credit assessments more efficiently and reducing decision-making costs.



***Corresponding Author:***

Oanh Thi Tran
International School, Vietnam National University
144 Xuan Thuy Street, Cau Giay Ward, Hanoi, Vietnam
Email: tranthioanh@vnu.edu.vn

## 1. INTRODUCTION

Non-performing loan (NPL) is a loan where the borrower has not paid back principal and interest for at least 90 days. Unfortunately, there is no standard way to identify bad loans in advance. Currently, in most financial and banking companies, the conventional approach to the NPLs prediction problem is to use the risk modeling method. In this approach, they first clean and prepare the data, then apply traditional statistical methods like weight of evidence and logistic regression, and finally turn the results into a credit score.

Along with the development of machine learning (ML) and artificial intelligence (AI), traditional statistical methods show inefficiency in making forecasts and predictions in the decision-making process for businesses. Because of that, there have been a number of research papers on the application of ML in NPL prediction; however, the common point of most of these papers is that they only focus on the application of ML algorithms to predict the output without noticing that the data in most financial and banking problems are faced with problems of imbalanced data. This means that the number of samples in one group (like cancer patients) can be 1,000 times smaller than in another group (like healthy patients), but most ML methods work better when the data is balanced. Typically, in the NPL prediction problem, this is a typical binary classification problem with two classes, NPL and PL. In which the number of observations of the NPL class (minority class) is many times less than the number of observations of the performing loans class (majority class), leading to the prediction of the minority class has many errors, while the prediction of that class is the most important.

Over the past two decades, there has been research on methods to solve this imbalanced-data problem. These methods are divided into two main techniques: over-sampling (add more samples to the smaller group until it has the same number as the bigger group), under-sampling (reduce the samples in the bigger group so they match, or are fewer than the samples in the smaller group). Synthetic minority oversampling technique (SMOTE) is a typical algorithm using the over-sampling technique, first introduced in 2002 by [1]. It works by creating synthetic samples that are similar to existing minority class samples. To this end, it selects a random minority class instance and finds its k-nearest neighbors (KNN). It then chooses one of the neighbors randomly, computes the difference between the feature vectors, and multiplies it by a random number between 0 and 1. This difference is then added to the selected minority instance to create a new synthetic example. By creating synthetic examples, SMOTE helps to balance the proportion of instances between the minority and majority class, making the training data more representative. This can improve the performance of ML algorithms, especially in cases where the minority class has critical information and needs to be well-represented [2].

NearMiss is a typical algorithm using the under-sampling technique proposed by Mani and Zhang [3]. The authors observed that the KNN algorithm tends to classify examples from the majority class more accurately than the minority class in imbalanced datasets. This is because the majority class has a larger representation in the dataset, making it more likely for the k nearest neighbors to be majority class instances. To address this issue, the NearMiss algorithm focuses on selecting representative examples from the majority class that are in close proximity to the minority class instances. However, these methods also have weaknesses. As SMOTE makes training very expensive because with large data sets, increasing the number of observations of the minority class equal to the majority class will cause the data to be greatly increased in size and time consuming to train, leading to memory lake [4]. As NearMiss, deleting the observations of the majority class will cause the data to lose a lot of information and lead to a decrease in the performance of the model [5].

Deep reinforcement learning has been successfully used in recent years to apply in computer games, robot control, self-driving cars, and other systems. Deep reinforcement learning has greatly improved classification performance for classification issues by deleting noisy data and studying better features. A proposed approach of deep reinforcement learning is indeed the great effective method for learning from imbalanced data because of how easily it can focus more attention on the smaller class by its rewards function or penalty's function [6]. The main idea of deep Q-learning is that try to memory the previous study by using replay buffer and use that memory for training, the agent will interact with environment by action, action will be determined based on policy. Environment will return agent reward or penalty if action is true or false. The goal of deep Q-learning is to achieve as many rewards as it can.

The objective of our study is to be designed as a proposed approach to handle imbalanced data and predicting NPLs using deep Q-learning algorithm. With the use of this method and a special, exclusive dataset of a Vietnamese lending service company, we are able to make significant advances to this field of study. Our experimental results show that deep Q-learning significantly improves NPL detection accuracy by effectively handling imbalanced data and learning optimal classification strategies.

In brief, this study has the contributions as follows:

i) Introduces deep Q-learning as an alternative to traditional ML models for predicting NPL, addressing the limitations of existing methods to handle imbalanced data by dynamically adjusting its focus on the minority class using reward and penalty mechanisms.
ii) Introduces an exclusive dataset of 83,732 customer records from a leading Vietnamese financial institution (2019–2022), ensuring practical relevance and applicability.
iii) Extensively conduct experiments to prove the effectiveness of the deep Q-network (DQN) methods in comparison with some strong baselines of traditional approach.

The remainder of this paper is structured as follows: section 2 introduces the dataset used in doing experiments. We present the proposed method using deep reinforcement learning in section 3. Then, section 4 shows the experimental setups and results. Lastly, section 5 summarizes the paper and then suggests some future research directions.

## 2. METHOD

### 2.1. Dataset

The dataset has 23 columns (22 independent variable and 1 dependent variable) and 83,732 observations. Of which, 15 input variables are categorical variables and the remaining 7 are numerical variables as shown in Table 1. The data collected from 01/01/2019 to 31/12/2022. Independent variables: 22 of them are listed in Table 1 and dependent variable shown in Table 2.

Table 1. Attributes of customers collected from the company

| No. | Attributes | Type | Explanation |
|---|---|---|---|
| 1 | AREA | Categorical | North - Central – South area of Vietnam |
| 2 | CATEGORY NAME | Categorical | Car, motorbike registration |
| 3 | PAPERTYPE | Object | Types of verification documents (Eg: Identity card, driving license, etc.) |
| 4 | PROVINCE_SHOP | Categorical | Province, city of that store (63 items) |
| 5 | DESCRIPTION | Categorical | Loan package (Eg: Consumer loan, business loan, ...) |
| 6 | KENH | Categorical | Lending channel (Eg: agent, Apps, Website, etc.) |
| 7 | LOAI_KHACH_HANG | Categorical | Customer types (individual, organization) |
| 8 | LOAI_HINH_CU_TRU | Categorical | Type of residence (Eg: temporary residence, permanent residence, …) |
| 9 | MARITAL | Categorical | Marital status (single, married, divorced no child, divorced with child, widow, unknown) |
| 10 | WORKPLACE_TYPE | Categorical | Type of workplace (Eg: 2.0: Indoors; 1.0: Outdoors) |
| 11 | INDUSTRY_NM | Categorical | Industry types (Eg: Industrial/Food/Mechanical, Transportation/Warehousing/Supply, ...) |
| 12 | JOB_NM | Categorical | Job types (Eg: Sales, workers…) |
| 13 | NUMBER_OF_CHILD | Number | Number of children |
| 14 | LOAN_PURPOSE | Categorical | Loan purpose |
| 15 | RESIDENCE_TIME | Number | Residence time in years |
| 16 | DISTANCE | Float | Distance from the center to the place of residence |
| 17 | IS_BAD_DEBT | Int | Customers have a history of bad debt |
| 18 | PACKAGE_CODE | Object | Loan package |
| 18 | IS_CUSTOMER_NEW | Boolean | Is customer new |
| 20 | INCOME | Number | Customer's income (in million VND) |
| 21 | MONEY_APPRAISAL | Number | Amount of property valuation |
| 22 | AGE | Number | Customer's age |

Table 2. Dependent variable

| Variable name | Type | Description |
|---|---|---|
| GOOD_BAD | Int | Good debt or Bad debt (>90 days overdue debt is bad debt) |

The dataset is collected from one of the largest financial institutions in Vietnam. The data describing each customer is shown in Table 1. The data was collected in nearly 4 years from Jan 2019 to Oct 2022 and split into train/development/test with the ratio 7:2:1. Numbers of observations and other statistics of the dataset are shown in Table 3.

Table 3. Some statistics about the data collected

| Data | #of Observation | #of Good Debt | #of Bad Debt | Ratio Good Debt | Ratio Bad Debt |
|---|---|---|---|---|---|
| **Train** | **60,286** | **52,084** | **8,202** | **86.39%** | **13.61%** |
| **Validation** | **15,072** | **13,384** | **1,688** | **88.80%** | **11.20%** |
| **Test** | **8,374** | **7,529** | **845** | **89.91%** | **10.09%** |
| Total | 83,732 | 72,997 | 10,735 | 88.37% | **11.63%** |

### 2.2. Proposed approach to predict non-performing loan

#### 2.2.1. Modeling the task as an RL problem

NPL prediction usually seen as a yes/no classification problem. But here we not use normal supervised learning, we make it like a RL problem. In this case, an agent learn to tell if loan is good or bad by talking with environment and try to change its policy to get more rewards. For deep Q-learning, the first step is put the start state into neural network, then it give back Q-values for all actions. The big difference between Q-learning and deep Q-learning is how they show and find Q-values. In Q-learning, Q-values is keep in a Q-table, where every state-action has its value. But in deep Q-learning, we use deep neural network (DNN) to guess the Q-values, not put them in a table. The neural net take the state as input and give out Q-values for each action in that state.

The main idea of RL is try and fail, like do again and again and learn from every try. Deep Q-learning have six big things: agent, environment, state, action, reward, and policy. The meaning of these six things explain like this:

Assume that the imbalanced training data set is $D = \{(x_1, l_1), (x_2, l_2), \ldots, (x_n, l_n)\}$ where $x_i$ is the $i^{th}$ sample and $l_i$ is the label of the $i^{th}$ sample:

– Environment: the dataset containing loan applicants' features and their actual loan performance.

– State $s_t$: a 22-dimensional feature vector representing a loan applicant. When training starts, the agent get the first sample $x_1$ as the first state $s_1$. The state $s_t$ of at every time step means the sample $x_t$. When the new episode starts, the environment mix up the order of samples in training data.
– Action $a_t$: the agent chooses between two actions: classify as NPL (bad loan) or classify as PL (good loan). For yes/no classification problem, A={0, 1} where 0 represents the smaller class and 1 represents the bigger class.
– Reward $r_t$: a reward $r_t$ is like the feedback from the environment, it tells if agent's action is good or bad. To help agents learn better rule in unbalanced data, the reward value for sample in small class is bigger than sample in big class. So, if agent say right or wrong on small class sample, the environment give bigger reward or bigger punish. Small class sample is hard to find correct in unbalanced dataset. To make agent see small class better, the algorithm must be more careful with small class. So, when agent meet small class sample, it get big reward or big punish. The reward function is like this:

$$R(s_t, a_t, l_t) = \begin{cases} +1, & a_t{=}l_t \text{ and } s_t{\in}D_P \\ -1, & a_t{\neq}l_t \text{ and } s_t{\in}D_P \\ \lambda, & a_t{=}l_t \text{ and } s_t{\in}D_N \\ -\lambda, & a_t{\neq}l_t \text{ and } s_t{\in}D_N \end{cases}$$

where λ ∈ [0, 1], $D_P$ is smaller class sample set, $D_N$ is bigger class sample set. The best performance in experiment is when λ equals to the imbalanced ratio $p = \frac{|D_P|}{|D_P|}$
– Transition probability P: transition probability $p(s_t + 1|s_t, a_t)$ is deterministic. The agent goes from the current state $s_t$ to the next state $s_t$ +1 by following the order of samples in the training data. Discount factor γ: γ ∈ [0, 1] is to help balancing the current and future reward.
– Episode in RL is just the path from the first state to the last state $\{s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_t, a_t, r_t\}$. An episode stops when all samples in training data are classified or when the agent wrongly classifies the sample from smaller class.
– Policy $\pi_\theta$: the policy $\pi_\theta$ is like a rule function π: S → A where $\pi_\theta$ ($s_t$) means what action agent should do when it in state $s_t$. The policy $\pi_\theta$ can be seen like a classifier with θ.
– Q-value $Q(s, a)$: the expected cumulative reward for taking action $a$ in state $s$, which the agent learns to optimize.

With the meaning and symbols above, the unbalanced classification problem is just to find the best policy π∗: S → A, that make the cumulative rewards as big as possible. The overall structure is shown in Figure 1. One key strong point of DQN is that it can adaptively focus on minority class predictions through its reward mechanism:
– Penalty for misclassifying NPLs (false negatives) is larger → encourages correct detection of bad loans.
– Higher reward for correctly classifying NPLs → forces the agent to learn the minority class better.

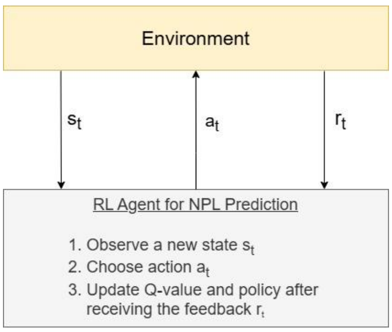


Figure 1. The architecture of modelling NPL prediction as a reinforcement learning problem

### 2.2.2. Implementation using deep Q-learning
a) State representation: each loan is represented as a 22-dimensional vector.
b) DQN: a neural network estimates Q-values for each action given the state (loan data).
c) Training process:
  – The agent observes a loan's features (state).
  – It chooses an action (classify as NPL/PL) using an ε-greedy strategy (balancing exploration and exploitation).

– It receives a reward based on classification accuracy.
– The experience is memorized in a replay buffer.
– The Q-network is updated using Bellman's equation:

$$Q(s,a) = r + \gamma \max Q(s', a')$$

where $\gamma$ is the discount factor of future rewards.

d) Evaluation: the model is tested on unseen loan applications to measure prediction accuracy and recall (especially for NPLs).

## 3. RESULTS AND DISCUSSION

In this section, we conduct experiments to validate the proposed model and at the same time we give some comprehensive discussion.

### 3.1. Experiment setup

#### 3.1.1. Hyper-parameter tuning

To implement DQN for NLP prediction, we used ε-greedy policy. We varied the hyper-params and chose the best value using the development set. The exploration rate ε decreases linearly from 1.0 to 0.001 during the process. The replay memory size is 1,070,000 and the interactions between agent and environment are about 1,000,000 steps. $\gamma$ - the discount factor is set at 0.2. The Q-network is optimized with the Adam algorithm with its learning rate at 0.0001, the batch size at 32. The amount of data collected for replay buffer each episode is 3,000. The step interval to collect data during training is 2,000. Update the target Q-network every 2,000 episodes. The number of imbalance ration is 0.13. For other ML methods and techniques to handle class imbalance, we exploited sklearn libraries.

#### 3.1.2. Evaluation metrics

To measure the effectiveness of the NPL prediction model, we focus on detecting class 1 (high-risk customers likely to default) rather than evaluating both classes equally. Identifying these customers is crucial for financial institutions to mitigate risk and reduce NPL growth. For this, we measure the precision, recall and F1 scores on this class. In addition, on the best-performing model, we also report the area under the curve (AUC) score, Matthews correlation coefficient (MCC) and G-mean which are also standard metrics for evaluating imbalanced classification.

### 3.2. Experiment results

Three types of experiments:

1) The first experiment tests the performance of traditional ML models without applying any data-balancing techniques. We trained the following models on the dataset using logistic regression [7]-[9]; decision tree [10]-[13]; random forest [14]-[17]; SVM [18]-[20]; LightGBM [21]-[23] and XGBoost [24]-[26]. The purpose is to observe how class imbalance affects model performance, especially in detecting high-risk (NPL) customers.
2) In the next experiment, we apply resampling techniques to improve class balance such as over-sampling and under-sampling techniques. This is to assess whether resampling techniques improve class 1 recall and overall performance of ML models.
3) In the third experiment, we introduce DQN, an RL approach that dynamically adjusts decision boundaries based on reward signals. This is to determine if DQN outperforms traditional models in handling imbalanced data without the need for resampling techniques.

#### 3.2.1. Performance of traditional ML models without applying any data-balancing techniques

Table 4 is the evaluation matrix for only class 1 (bad debt) between 6 ML algorithms in testing set. In overview, XGBoost has the best performance, followed by LightGBM and Random Forest. The worse performance is the decision tree, followed by logistic regression.

Table 4. Evaluation matrix of class 1 for multiple ML algorithms in testing set

| Metric | Logistic regression | Decision tree | Random forest | SVM | LightGBM | XGBoost |
|---|---|---|---|---|---|---|
| Precision | 57.02 | 49.78 | 59.98 | 56.76 | 59.66 | **60.21** |
| Recall | 71.60 | 67.57 | 72.90 | 70.53 | 74.20 | **75.38** |
| F1-score | 63.48 | 57.33 | 65.81 | 62.90 | 66.14 | **66.95** |

The top 3 algorithms with the highest performance in both precision, recall, and F1 score are XGBoost, random forest and LightGBM, respectively. In precision score, the highest is XGBoost, followed by random forest, the lower is XGBoost. In recall score, the highest is XGBoost, followed by LightGBM, the lower is random forest. Finally, F1-score will give an overall rating with a combination of precision and recall, F1-score of XGBoost is the highest, followed by LightGBM, lower is random forest. In summary, it can be concluded that the boosting algorithm has the best performance and it is superior to bagging algorithms.

### 3.2.2. Conventional methods to handle class imbalance

Table 5 shows the F1-score of several handling imbalanced methods in 6 ML algorithms on testing set. However, the special thing is that random forest has become the algorithm with the highest F1-score even though in the validation set is lower than boosting algorithms. This has shown that with the method of handling imbalanced data, specifically the over-sampling methods, these methods have increased the accuracy of the model. However, the level of performance improvement is there but not high when using traditional imbalanced data processing methods. The research direction in the next section will show a new method, which is widely known in the AI community but has not been applied much in the problem of imbalanced data. It's deep Q-learning algorithm, we will apply and optimize this algorithm to show that it will be amazingly great in handling the imbalanced-data problem compared to the above methods.

Table 5. Comparison F1-score of handling imbalanced data methods in testing set

| Sampling method | Logistic regression | Decision tree | Random forest | SVM | LightGBM | XGBoost |
|---|---|---|---|---|---|---|
| Base | 63.4 | 60.05 | 66.98 | 63.72 | 66.14 | 66.95 |
| SMOTE | 62.83 | 60.05 | 66.53 | 62.50 | 66.18 | 66.28 |
| OVERSAMPLING | 61.80 | 58.90 | 66.98 | 63.72 | 64.72 | 66.21 |
| UNDERSAMPLING | 61.69 | 55.36 | 66.15 | 62.99 | 63.68 | 64.02 |

### 3.2.3. Effectiveness of the proposed approach using deep Q-learning to handle imbalanced data

We tried with different architectures, varied from 2, 3, 4, 5, and 6 layers to find the best configuration. Finally, the best one is the one using 4 hidden layers with 526 nodes in each layer. For each layer, we use rectified linear unit (ReLU) as the activation function. And the output layer includes two nodes which are good and bad. In total, we have 864,746 parameters.

Figure 2 shows the precision recall (PR) curve and receiver operating characteristic (ROC) curve. As we can see, in PR curve, the average precision score is very good in training set but quite low in testing set. In ROC curve, AUC score is quite good for the test set, that means the model has classified quite well between 2 classes good and bad. In the testing set, the precision score of deep Q-learning has become the highest, almost 2% higher than LightGBM and 1% higher than XGBoost. Deep Q-learning's recall is only reduced by nearly 1% compared to the validation set and still has the highest score.

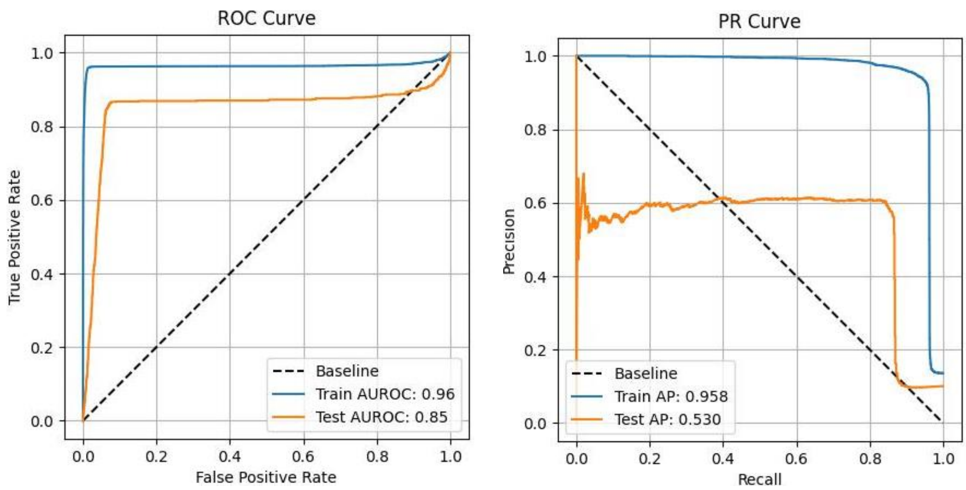


Figure 2. ROC curve and PR curve

Finally, in Figure 3, deep Q-learning has shown great performance compared to boosting or bagging algorithms in F1-score. Deep Q-learning's F1-score is 8% higher than LightGBM and 7% higher than XGBoost and random forest.

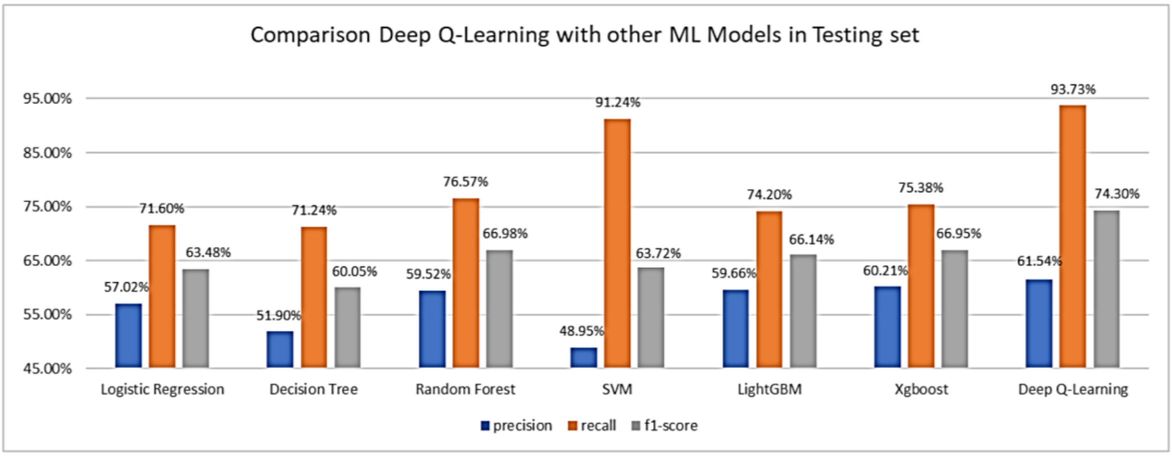


Figure 3. Comparison deep Q-learning with other ML models in testing set

For the best model, we also calculated the final results as shown in Table 6. The model performs quie well on the majority class, achieving an F1-score of 96.25% and precision of 99.25%, while still identifying minority NPL cases effectively with a high recall of 93.73%. Although precision for class 1 is lower (61.54%) due to class imbalance, the overall metrics: AUC (0.856), G-mean (93.48%), and MCC (0.48) indicate that the model maintains good discriminative ability and handles the imbalance effectively. Overall, the model yielded balanced and reliable performance for NPL prediction.

Table 6. Detailed metrics for measuring performance of deep Q-learning on imbalanced dataset

| | Label 0 | Label 1 | Macro avg | Weighted avg |
|---|---|---|---|---|
| Precision | 99.25 | 61.54 | 80.40 | 95.45 |
| Recall | 93.43 | 93.73 | 93.58 | 93.46 |
| F1 | 96.25 | 74.3 | 85.27 | 94.04 |
| AUC score | 0.856 | | | |
| MCC | 0.48 | | | |
| G-mean | 93.48 | | | |

### 3.2.4. Model discussion

Deep Q-learning is a black-box model. However, we can still explain the model by providing intrinsic feature importance scores using a surrogate-model explainability approach widely used in the literature. We conducted an explainability analysis using SHAP. We trained an XGBoost surrogate model to imitate the DQN's predictions and computed SHAP values on the surrogate. The results show: high-impract indicators are history of past bad debt, income level, customer's age, job types, loan package types, and years at current residence; moderate indicators are asset valuation, amount of property valuation, industry sector, marital status, industry types, workplace type; and the remaining features are at lower-impact indicators.

The experimental results indicate that the proposed model is feasible for real-world deployment. Training remains stable even with a large replay buffer, as the average Q increases steadily and the loss decreases over time. The model is trained on a local machine with limited resources (CPU only with 8 GB RAM), while the testing phase is very fast, making it suitable for practical applications requiring quick responses.

## 4. CONCLUSION

This paper proposed a novel approach to NPL prediction by leveraging deep Q-learning, a reinforcement learning technique, instead of traditional ML methods. Unlike conventional approaches that treat NPL prediction as a static classification problem, our method models the problem as a sequential decision-making task, where the agent learns optimal loan classification strategies through interaction with the environment. Our research addressed key challenges in NPL prediction, including class imbalance, which often leads to poor recall for NPL cases.

By designing a reward function that penalizes misclassifications more heavily for the minority class (NPL), our model learns to focus on correctly identifying high-risk loans, improving decision-making accuracy. We also experimented with data balancing techniques, such as over-sampling (SMOTE) and under-sampling (NearMiss), and compared deep Q-learning with traditional ML models (logistic regression, decision trees, SVM, random forest, LightGBM, and XGBoost).

Our results, based on 83,732 customer records from a leading financial institution in Vietnam (2019-2022), demonstrate that deep Q-learning outperforms traditional models, particularly in handling imbalanced datasets. The proposed approach offers a more adaptive and automated NPL prediction framework, reducing manual risk assessment efforts and improving loan approval efficiency. For the future work, we will explore more advanced reinforcement learning techniques like policy gradient methods to enhance stability or apply explainability techniques to increase model transparency for financial institutions.

## FUNDING INFORMATION

This research is funded by International School, Vietnam National University, Hanoi (VNU-IS) under project number CS.2024-05.

## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Luyen Anh Do | | | ✓ | ✓ | | | | ✓ | ✓ | ✓ | ✓ | | | |
| Huong Thi Viet Pham | | ✓ | | | | ✓ | | ✓ | ✓ | | | ✓ | | ✓ |
| Thinh Duc Le | ✓ | | | ✓ | | | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ |
| Oanh Thi Tran | ✓ | ✓ | ✓ | | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | | |

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

## CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

## DATA AVAILABILITY

Derived data supporting the findings of this study are available from the corresponding author on request.

## REFERENCES


[1] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[2] D. Elreedy and A. F. Atiya, "A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance," *Information Sciences*, vol. 505, pp. 32–64, Dec. 2019, doi: 10.1016/j.ins.2019.07.070.

[3] I. Mani and I. Zhang, "kNN approach to unbalanced data distributions: A case study involving information extraction," 2003.

[4] I. M. Alkhawaldeh, I. Albalkhi, and A. J. Naswhan, "Challenges and limitations of synthetic minority oversampling techniques in machine learning," *World Journal of Methodology*, vol. 13, no. 5, pp. 373–378, Dec. 2023, doi: 10.5662/wjm.v13.i5.373.

[5] M. Carvalho, A. J. Pinho, and S. Brás, "Resampling approaches to handle class imbalance: a review from a data perspective," *Journal of Big Data*, vol. 12, no. 1, p. 71, Mar. 2025, doi: 10.1186/s40537-025-01119-4.

[6] E. Lin, Q. Chen, and X. Qi, "Deep reinforcement learning for imbalanced classification," *Applied Intelligence*, vol. 50, no. 8, pp. 2488–2502, Aug. 2020, doi: 10.1007/s10489-020-01637-z.

[7] "Logistic regression," in *Encyclopedia of Machine Learning*, Boston, MA: Springer US, 2011, pp. 631–631.

[8] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *The Journal of Educational Research*, vol. 96, no. 1, pp. 3–14, Sep. 2002, doi: 10.1080/00220670209598786.

[9] S. Sperandei, "Understanding logistic regression analysis," *Biochemia Medica*, pp. 12–18, 2014, doi: 10.11613/BM.2014.003.

[10] Y. Y. Song and Y. Lu, "Decision tree methods: applications for classification and prediction," *Shanghai Archives of Psychiatry*, vol. 27, no. 2, pp. 130–135, 2015, doi: 10.11919/j.issn.1002-0829.215044.

[11] I. D. Mienye and N. Jere, "A survey of decision trees: concepts, algorithms, and applications," *IEEE Access*, vol. 12, pp. 86716–86727, 2024, doi: 10.1109/ACCESS.2024.3416838.

[12] H. Blockeel, L. Devos, B. Frénay, G. Nanfack, and S. Nijssen, "Decision trees: from efficient prediction to responsible AI," *Frontiers in Artificial Intelligence*, vol. 6, Jul. 2023, doi: 10.3389/frai.2023.1124553.

[13] J. . QUINLAN, "Simplifying decision trees," *International Journal of Human-Computer Studies*, vol. 51, no. 2, pp. 497–510, Aug. 1999, doi: 10.1006/ijhc.1987.0321.

[14] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *The Stata Journal: Promoting communications on statistics and Stata*, vol. 20, no. 1, pp. 3–29, Mar. 2020, doi: 10.1177/1536867X20909688.

[15] Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, "An improved random forest based on the classification accuracy and correlation measurement of decision trees," *Expert Systems with Applications*, vol. 237, p. 121549, Mar. 2024, doi: 10.1016/j.eswa.2023.121549.

[16] H. A. Salman, A. Kalakech, and A. Steiti, "Random forest algorithm overview," *Babylonian Journal of Machine Learning*, vol. 2024, pp. 69–79, Jun. 2024, doi: 10.58496/BJML/2024/007.

[17] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[18] T. Evgeniou and M. Pontil, "Support vector machines: theory and applications," in *Machine Learning and Its Applications*, 2001, pp. 249–257.

[19] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, Sep. 2020, doi: 10.1016/j.neucom.2019.10.118.

[20] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1023/A:1022627411411.

[21] G Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[22] M. R. Machado, S. Karray, and I. T. de Sousa, "LightGBM: an effective decision tree gradient boosting method to predict customer loyalty in the finance industry," in *2019 14th International Conference on Computer Science & Education (ICCSE)*, Aug. 2019, pp. 1111–1116, doi: 10.1109/ICCSE.2019.8845529.

[23] L.-H. Li, A. K. Sharma, and S.-T. Cheng, "Explainable AI based LightGBM prediction model to predict default borrower in social lending platform," *Intelligent Systems with Applications*, vol. 26, p. 200514, Jun. 2025, doi: 10.1016/j.iswa.2025.200514.

[24] T. Chen *et al.*, "XGBoost: extreme gradient boosting," *CRAN: Contributed Packages*. Sep. 01, 2014, doi: 10.32614/CRAN.package.xgboost.

[25] Z. Arif Ali, Z. H. Abduljabbar, H. A. Tahir, A. Bibo Sallow, and S. M. Almufti, "eXtreme gradient boosting algorithm with machine learning: a review," *Academic Journal of Nawroz University*, vol. 12, no. 2, pp. 320–334, May 2023, doi: 10.25007/ajnu.v12n2a1612.

[26] D. Tarwidi, S. R. Pudjaprasetya, D. Adytia, and M. Apri, "An optimized XGBoost-based machine learning method for predicting wave run-up on a sloping beach," *MethodsX*, vol. 10, p. 102119, 2023, doi: 10.1016/j.mex.2023.102119.


## BIOGRAPHIES OF AUTHORS

**Luyen Anh Do** got his bachelor degree on business data analytics in 2023. Now, he is working as a scientist at Big Data Department of VietinBank in Vietnam. His main research interests include data science, AI, and ML towards the applications in business fields. He can be contacted at email: luyen31122001@gmail.com.

**Huong Thi Viet Pham** obtained her B.Sc. in electrical engineering from Hanoi University of Science and Technology in 2007. She got her M.Sc. and Ph.D., both in electrical engineering, from University of Massachusetts Lowell in the United States, in 2010 and 2012. From 2012 to 2015, she was a researcher in the Manning School of Business, Lowell, Massachusetts. From 2017-2020, she was the faculty of VNU University of Engineering and Technology, Vietnam (VNU-UET). Since 2020, she works in International School – VNU. She is interested in data mining and analytics, machine learning methodologies, with applications in biomedical engineering. She can be contacted at email: huongpv@vnu.edu.vn.

**Thinh Duc Le** is a lecturer at International School, Vietnam National University, Ha Noi, Vietnam. He teaches mathematics in English for business majors. He obtained his Bachelor degree in 2001 and Master degree in 2004, both in mathematics at Ha Noi National University of Education, Vietnam. He obtained his Ph.D. degree in 2012 at the Pennsylvania State University, USA. His research interests include machine learning, mathematical finance, and economic statistics. He can be contacted at email: thinhld@vnu.edu.vn.

**Oanh Thi Tran** got the bachelor and master degrees in computer science at the University of Engineering and Technology, Vietnam National University, Hanoi in 2006 and 2009, respectively. She was awarded a Japanese Government Scholarship to pursue Ph.D. in Computer Science at Japan Advanced Institute of Science and Technology (JAIST) from 2011 to 2014. Currently, she is a lecturer at the International School of Vietnam National University, Hanoi (VNU-IS). Her main research interests are AI and ML. Her contributions to the field include 50 publications in esteemed journals and conferences. She can be contacted at email: oanhtt@gmail.com or tranthioanh@vnu.edu.vn.