A REINFORCEMENT LEARNING MODEL FOR LENDING PROBLEMS WITH LIMITED BUDGET AND INSUFFICIENT DATA

Radaporn Autravisittikul
6082959026

Advisor
Assoc. Prof. Thaisiri Watewai

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Financial Engineering
Department of Banking and Finance
Faculty of Commerce and Accountancy
Chulalongkorn University
Academic Year 2019

แบบจำลองการเรียนรู้แบบเสริมกำลังสำหรับปัญหาการให้สินเชื่อที่มีงบประมาณที่จำกัดและข้อมูลที่ไม่เพียงพอ

น.ส.รดาภรณ์ อุตราวิสิทธิกุล
6082959026

อาจารย์ที่ปรึกษา
รศ. ดร. ไทยศิริ เวทไว

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมการเงิน ภาควิชาการธนาคารและการเงิน
คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

Thesis Title A REINFORCEMENT LEARNING MODEL FOR LENDING PROBLEMS WITH LIMITED BUDGET AND INSUFFICIENT DATA
By Radaporn Autravisittikul
Field of Study Financial Engineering
Thesis Advisor Associate Professor Thaisiri Watewai, Ph.D.

Accepted by the Faculty of Commerce and Accountancy, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

.... Dean of Faculty of Commerce and Accountancy
(Associate Professor Wilert Puriwat, Ph.D.)

THESIS COMMITTEE

.... Chairman and External Examiner
(Anant Chiarawongse, Ph.D.)

.... Thesis Advisor
(Associate Professor Thaisiri Watewai, Ph.D.)

.... Examiner
(Associate Professor Sira Suchintabandid, Ph.D.)

รดาภรณ์ อุตราวิสิทธิกุล: แบบจำลองการเรียนรู้แบบเสริมกำลังสำหรับปัญหาการให้สินเชื่อที่มีงบประมาณที่จำกัดและข้อมูลที่ไม่เพียงพอ (A REINFORCEMENT LEARNING MODEL FOR LENDING PROBLEMS WITH LIMITED BUDGET AND INSUFFICIENT DATA) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ.ดร.ไทยศิริ เวทไว, 47 หน้า.

นโยบายการปล่อยกู้โดยทั่วไปจะอาศัยข้อมูลของผู้กู้ในการพิจารณาการปล่อยสินเชื่อ ดังนั้นผู้กู้ที่ไม่เคยมีประวัติการทำธุรกรรมกับสถาบันการเงินมักจะประสบปัญหาในการเข้าถึงแหล่งเงินทุน วิทยานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อสร้างแบบจำลองสำหรับการให้สินเชื่อโดยผู้ให้สินเชื่อมีงบประมาณที่จำกัดและมีข้อมูลที่ไม่เพียงพอ เนื่องจากผู้ให้สินเชื่อไม่มีความรู้เกี่ยวกับพฤติกรรมของผู้ขอสินเชื่อที่เข้ามา ทำให้ความแม่นยำในการพยากรณ์ความน่าจะเป็นที่ลูกหนี้จะผิดนัดชำระหนี้อยู่ในระดับต่ำในช่วงแรกของการทดสอบแบบจำลอง แบบจำลองสามารถเพิ่มความแม่นยำในการพยากรณ์ความน่าจะเป็นที่ลูกหนี้จะผิดนัดชำระหนี้ได้โดยการสังเกตและเรียนรู้จากผลลัพธ์หลังจากการให้สินเชื่อ โดยเมื่อแบบจำลองเลือกที่จะให้สินเชื่อแก่ผู้ขอสินเชื่อที่ชำระเต็มจำนวน งบประมาณที่ตั้งไว้ก็จะเพิ่มขึ้นตามจำนวนดอกเบี้ยที่ได้รับ และเมื่อแบบจำลองให้สินเชื่อแก่ผู้ขอสินเชื่อที่ผิดนัดชำระ งบประมาณก็จะลดลงตามปริมาณความเสียหายที่เกิดขึ้น จะเห็นว่าแบบจำลองสามารถเรียนรู้ได้มากขึ้นเมื่อแบบจำลองเลือกที่จะให้สินเชื่อมากขึ้น ในขณะเดียวกันก็ทำให้งบประมาณมีความเสี่ยงเพิ่มมากขึ้น เป้าหมายของแบบจำลองคือการทำให้งบประมาณหลังจากการให้สินเชื่อมีค่ามากที่สุด วิทยานิพนธ์ฉบับนี้ใช้วิธีการเรียนรู้แบบเสริมกำลังในการสร้างแบบจำลอง โดยใช้ข้อมูลของผู้ขอสินเชื่อร่วมกับความแม่นยำของแบบจำลองและงบประมาณคงเหลือในการพิจารณาให้สินเชื่อ เมื่อนำแบบจำลองไปทดสอบกับข้อมูลจำลอง พบว่างบประมาณหลังการปล่อยสินเชื่อจากแบบจำลองที่ใช้วิธีเรียนรู้แบบเสริมกำลังมีค่าสูงกว่าเมื่อเทียบกับแบบจำลองทั่วไป และเมื่อนำแบบจำลองที่ใช้วิธีเรียนรู้แบบเสริมกำลังไปทดสอบกับข้อมูลจริง พบว่าแบบจำลองที่ใช้วิธีเรียนรู้แบบเสริมกำลังสามารถให้ผลลัพธ์ที่ดีในสินเชื่อบางประเภท

ภาควิชา การธนาคารและการเงิน ลายมือชื่อนิสิต ....

สาขาวิชา วิศวกรรมการเงิน ลายมือชื่ออ.ที่ปรึกษาหลัก ....

ปีการศึกษา 2562

6082959026: MAJOR FINANCIAL ENGINEER
KEYWORDS : REINFORCEMENT LEARNING / BANK LENDING / LIMITED BUDGET / INSUFFICIENT DATA

RADAPORN AUTRAVISITTIKUL: A REINFORCEMENT LEARNING MODEL FOR LENDING PROBLEMS WITH LIMITED BUDGET AND INSUFFICIENT DATA. ADVISOR: ASSOC. PROF. THAISIRI WATEWAI, Ph.D., 47 pp.

Traditional lending policy requires sufficient data for making lending decisions, therefore, some small companies could not access to the fund. In this study, we propose a decision making model that can decide whether to accept or reject a sequence of unfamiliar loan applications while having a limited budget. Our model does not have any knowledge about the incoming loans, therefore, it can predict the default probability with low accuracy at the beginning. The model can learn by observing the outcomes of the accepted loans. The model's budget increases every time the model accepts a fully paid loan and decreases when the model accepts a defaulted loan. The objective of our model is to maximize the final budget. By using the reinforcement learning method, we propose a decision making model that takes the current budget and model accuracy into consideration when making decisions. Based on simulated data, the results show that our model yields a better performance compared to a traditional default prediction model. For the real data, our model performs well in some type of loans.

Department: Banking and Finance Student's signature ....

Field of Study: Financial Engineering Advisor's signature ....

Academic Year: 2019

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

E-Commerce has been growing in Thailand. According to Electronic Transactions Development Agency (ETDA) Annual report 2018, Thailand e-Commerce has approximate value of 3 trillion baht with an 8.76% growth rate from 2017 to 2018 and a 38.31% growth rate from 2014 to 2018. The high growth rate of Thailand e-Commerce indicates that Thailand has many opportunities for e-Commerce companies. Unfortunately, some small e-Commerce companies cannot access to funds from any banks and these companies have to get the funds from non-standard loans. The main reason is that some companies cannot meet the traditional credit risk requirements set by banks even if some of them can return the funds to non-standard loans.

According to the detail specified in the previous paragraph, there are many opportunities in the lending business. The problem is that banks do not lend to new or unfamiliar types of business, of which many are small e-Commerce companies. One reason that banks do not lend is insufficiency of data for achieving accurate prediction of default. It is hard for the banks to decide whom to lend to because banks cannot predict default accurately. Unfortunately, the only way to obtain more data is to lend. Lending to non-defaulted borrowers can generate profit. On the contrary, lending to defaulted borrowers generates loss. Prediction accuracy is low at the beginning, therefore, cost of exploring from making mistakes (i.e., lending to the defaulted borrowers) can be high. Furthermore, given that we have limited budget to explore this new group of borrowers, making mistakes too often at the beginning may use up all the budget and hence stop us from making more loans. However, being too conservative and lending to borrowers, who we strongly believe that are low risk borrowers, may take extremely long time to collect enough data and leads to losing competitive advantage.

The low accuracy of the prediction model and the budget constraint should be taken into account in the decision making process because each lending decision affects the amount of data that the prediction model can get and causes a change in the budget. In the lending situation, rejecting most of the loans could save the budget, but this makes the model learn slowly. On the other hand, accepting too many loans could make the budget drop even though more data could be collected. This kind of problem, in which an action affects environment, can be handled by a method called reinforcement learning (RL). Barto et al. (1981) presented RL as a model that observes environment, then the model decides an action based on the state of the environment. The objective of the model is to optimize the reinforcement signal, also known as reward, which agent observes after taking an action. RL can be viewed as an approach to find a balance between exploration and exploitation. In the lending business sense, an exploration is to gain more data for learning without caring about losing the budget, as the model believes that learning more leads to higher accuracy. On the other hand, an exploitation is to make best decision based on the current knowledge/belief. In this work, the RL model needs a forecast of probability of default to make decision.

In order to predict the probability of default, Fantazzini and Figini (2009) proposed Random Survival Forests (RSF) model for SME credit risk measurement. They used new financial ratios in the RSF model, which is a non-parametric procedure, then compared it with a logistic regression model, which is a parametric procedure. They reported that both could predict probability of default for Small and Medium Enterprises, but the logistic regression model could predict better than the RSF model in terms of forecasting performances in out-of-sample data due to less estimation bias. Credit scoring models are also used to estimate the probability of default. Harris (2015) measured the credit scoring by using clustered support vector machine (CSVM). The author used German and Barbados dataset to train the model. He created a linear regression model, linear support vector machine (SVM) model and radial basis function kernel SVM model. He compared models in terms of AUC (area under the receiver operating characteristic (ROC) curve), balanced accuracy (BAC), training accuracy, test accuracy and training time. He suggested that CSVM performs better than the nonlinear SVM based techniques in terms of AUC. However, the classification performance and mean model training time of this model still need to be improved.

Another approach is to use credit ratings to predict default probability as low credit rating indicates high probability of default. Chi and Zhang (2017) proposed a credit rating model by combining a rank sum test and rank correlation analysis using data from a Chinese Bank. They use the Mann–Whitney rank sum test to ensure that only indices that can classify default and non-default samples are included in the model. Then they avoid multicollinearity problem by using the Spearman rank correlation analysis to remove duplicate indices. Next, they used entropy weighting to weight indices and apply those indices to the credit rating model. Their model divides credit rating into nine levels and calculates loss given default (LGD) for each level. The result is reasonable as high credit rating gives lower rates of default and loss. They discovered that non-financial indices also help identifying defaults and their model could predict credit rating for new loan customer with reasonable accuracy rate. Because all of these researches aimed to create default probability prediction models, the budget constraint and decision making are not included in their analyses.

To account for limited budget, the concept of safe RL can be helpful. Safe RL is RL with some safety constraint. Leike et al. (2017) mentioned that safe RL's environment has two functions: (i) A reward function, which is observed by the agent and (ii) A performance (safety) function, which agent cannot observe, but this function can convince the agent to act the way we want the agent to do. When these two functions are not identical, they called this a specification problem. They considered safety in eight ways: (i) Safe interruptibility, (ii) Avoiding side effects, (iii) Absent supervisor, (iv) Reward gaming, (v) Self-modification, (vi) Distributional shift, (vii) Robustness to adversaries and (viii) Safe exploration. In our situation, the reward function can be viewed as gain and loss from providing loans and the performance function is the value of the budget. Leike et al. (2017) summarized Safe exploration as how to build an agent that respects the safety constraint at the beginning of learning and during the operational period. Pecka et al. (2014) introduced many approaches to make safe exploration. They also defined safety in many ways: (i) Cost safety, (ii) Ergodic safety, (iii) Safety in terms of expected variance and (iv) Safe explorations. By using Q-learning algorithm, they made model safe by labeling states and actions, thus they know safety of each states and actions by following Hans et al. (2008). State spaces were divided into safe, critical and unsafe states. To help the model at the beginning of the training process, Saunders et al. (2017) used human to intervene any unsafe actions. They mentioned that during the training process, the model did not have enough knowledge to avoid making bad actions. Therefore, they used human to overwrite the unsafe actions that the model made. They trained a supervised learner to mimic intervention decisions of the human to intervene the model's unsafe actions. They used Convolutional Neural Network (CNN) blocker as the supervised learner. They named this model Safe RL via Human Intervention (HIRL) scheme.

Some of the previous studies identified the risk and created new parameter or function to control this risk. Junges et al. (2015) used permissive schedulers based on SMT-solving to ensure that unsafe actions are avoided. Berkenkamp et al. (2017) defined safety in terms of stability guarantee. They used Lyapunov function to determine whether state-action were safe or not. Fan et al. (2019) also used Lyapunov to provide safety cost and used Gaussian Process to provide statistical guarantee.

The objective of this research is to create a decision making model that can make a profit, even if the model has limited knowledge about clients. To make sure that the model is aware of loss from lending, the model needs to concern about the prediction accuracy and the budget. Therefore, we created two extra parameters used in the prediction model: (i) Prediction accuracy and (ii) Budget. Thus, we can ensure that our model takes prediction accuracy and budget constraint into consideration. We expect that our model is more cautious when the budget and the accuracy level are low. Safe exploration, mentioned in Leike et al. (2017), could be applied in our model. We want the model to make an action that concerns about budget at the beginning of the lending process. From Saunders et al. (2017)'s HIRL scheme, we create an intervention module. We want this module to adjust the probability of default according to the current budget and the model accuracy. We aim that this module can help the prediction model learn how to make an action, with respect to budget and the model accuracy, in the same way as an expert would do.

2. OVERVIEW OF RELATED TECHNIQUES

In this section we will review the techniques that are used in this thesis. First, we explain the basic of the reinforcement learning method. Then we will talk about the neural network model. The neural network will update its parameters using the reinforcement learning algorithm. Next, we will talk about the safe reinforcement learning and finally we will talk about the one-hot encoding method used for preparing the input data.

2.1. Reinforcement learning

In RL, an agent takes actions within an Environment, then the agent observes changes in the environment and receives rewards. The agent uses an algorithm, called policy, to determine an action for a given state of the environment. The objective of RL is to make an action that maximizes the expected long-term rewards.

In a simple lending problem, the Environment is represented by a group of loan borrowers or clients. The current state of the environment is the current information of the current client. The client's information is observed by the agent. Then the agent takes an action based on its policy and receives a reward as a result of its action. There are two actions that can be performed by the agent: (i) To accept and (ii) To reject the loan of the current client. For example, if the agent decides to accept a loan, but this loan is defaulted, then the agent receives a signal, called reward, which has a negative value in this case. The agent uses this reward to learn that this kind of borrower is not worth to lend because the current client defaulted. The process is illustrated in Figure 1.

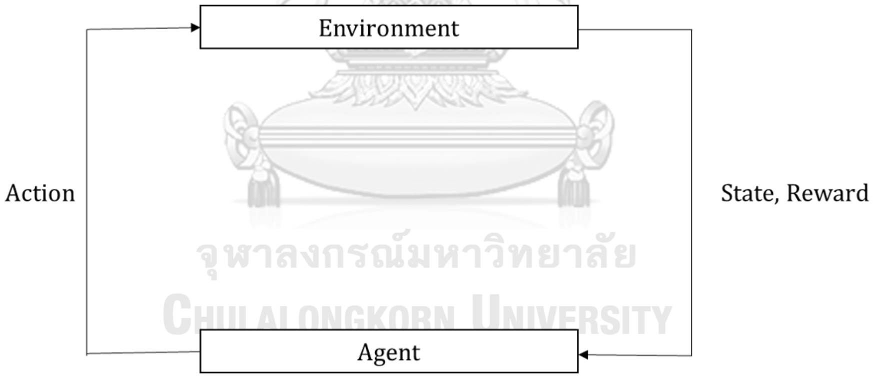


Figure 1: Reinforcement Learning

The policy is an algorithm that is used by the agent to determine the agent's actions. Policy Gradients and Q-learning are two commonly used policies in RL. For Q-learning, the Environment is mapped into Markov decision process (MDP). This algorithm creates reference table, called q-table, which is a state-action matrix. The q-table stores q-values, one for each state-action pair. A high q-value indicates good action given that state. The agent has two ways to determine the action: (i) By using the q-table and (ii) Randomly takes an action. The agent's chance to explore is some fixed probability $\boldsymbol{\varepsilon}$. By using the q-table, the agent finds best action based on q-table given current state by exploiting its belief. On the other hand, when the agent randomly picks an action, the agent can explore. For Policy Gradients (PG) algorithm, the algorithm optimizes its parameters based on the gradients of the loss between the chosen action and the suggested action from the model, and the observed reward. In each iteration, the agent picks an action based on the model's prediction and computes the gradients at each step. Positive rewards indicate that the chosen actions are good, and the model should increase the likelihood of choosing the same actions when similar clients come. The gradient descent step is used to update the model parameters.

In this research, we choose to use the Policy Gradients algorithm as a policy because the Environment states are client's information, not the state of MDP in our problem.

2.2. Neural network

In our research, the neural network model is used to calculate the probability of default. The neural network model consists of nodes, layers, weights, activation functions and loss functions. An example of a neural network is illustrated in Figure 2. The neural network is divided into input layer, hidden layers and output layer. Each layer consists of a set of nodes. Nodes in the input layer take the input data and pass them onto nodes in the next layers, which is the first hidden layer, through the weights between the nodes and the activation functions of the nodes in the next layer. The information in the first hidden layer is passed onto the second hidden layer and so on in similar way until it reaches the output layer, which generates the outcome of the model.

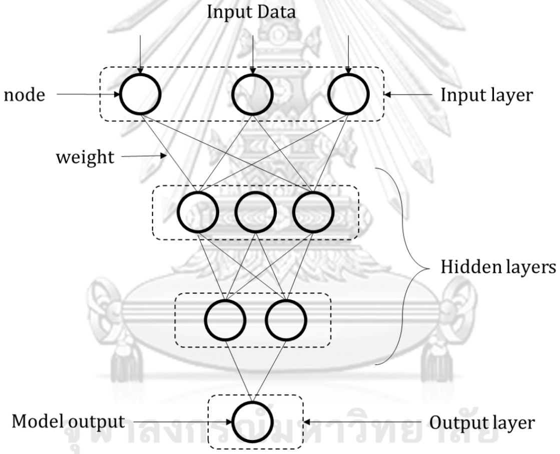


Figure 2: Neural network's components

The loss function in our case calculates the loss based on the suggested action and the chosen action. The neural network model is used in the reinforcement learning model. Figure 3 shows how a neural network can be used in the RL model under our lending context.

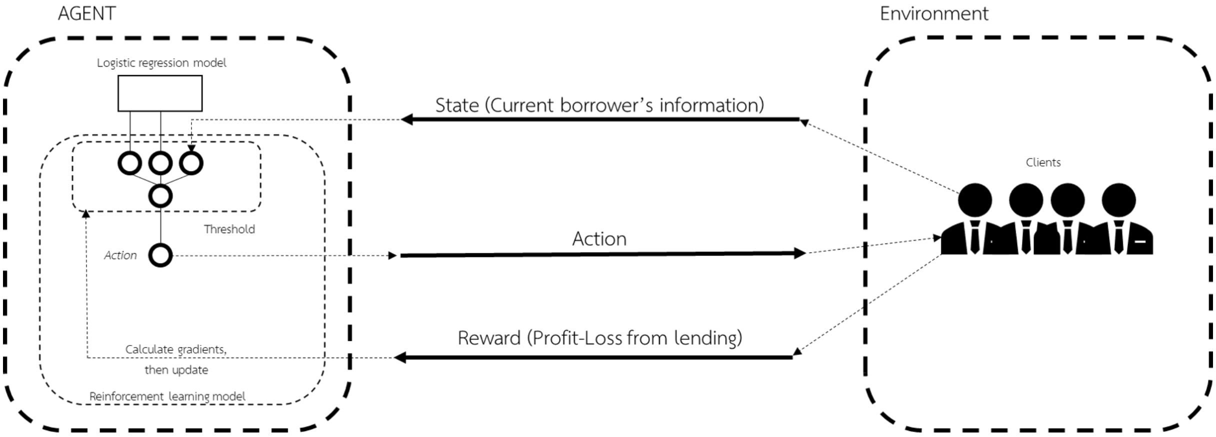

AGENT
Environment
Logistic regression model
State (Current borrower's information)
Clients
Threshold
Action
Action
Reward (Profit-Loss from lending)
Calculate gradients,
then update
Reinforcement learning model

Figure 3: RL's component

The state of environment is used as input nodes in the first layer of the neural network model. The first layer is fully connected to the second layer by corresponding weights. Each node in the second layer is calculated by applying the multiplication of the first layer nodes and the weights to an activation function. There are many activation functions that could be used in the neural network model, such as ReLU, Leaky ReLU, eLU, sigmoid and tanh. Table 1 provides the definitions of these activation functions.

Table 1: Definition of activation functions

Function	$f(x)$
ReLu	$\max(0,x)$.
Leaky ReLU	$\max(0.1x,x)$.
eLU	$\begin{cases} e^x - 1 & \text{, when } x < 0 \\ x & \text{, otherwise.} \end{cases}$
sigmoid	$\frac{1}{1+e^{-x}}$.
tanh	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$.

The activation functions are illustrated in Figure 4.

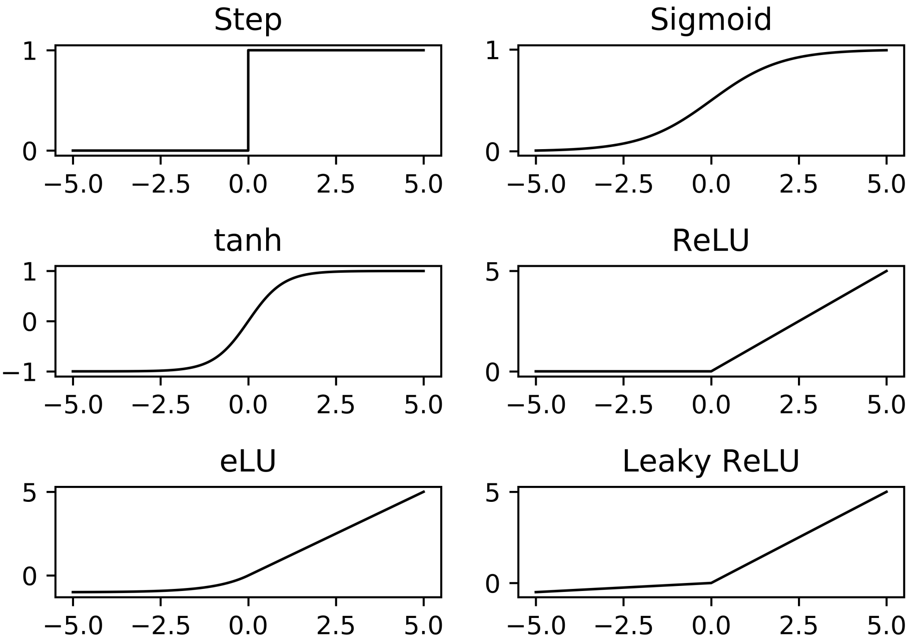


Figure 4: Activation function's graphs

To predict the probability of default, the sigmoid function can be used in the last layer of the prediction model as its value is between 0 to 1, and can be interpreted as a probability. The neural network model updates its parameters by using the backpropagation method. The model calculates the loss of prediction and computes the gradients for parameter optimization. There are many loss functions that can be used.

Let y_i be the actual action for observation i, and $\hat{y}_i$ the predicted outcome. Here are some commonly used loss function:

1. Mean absolute error (MAE), also known as L1, loss function

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \ ,$$

   where n is number of observations.

2. Mean square error (MSE), also known as L2, loss function

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

3. Binary cross entropy loss function

$$Binary\ cross\ entropy = -\sum_{i=1}^{n} [y_i log(\hat{p}_i) + (1-y_i) log(1-\hat{p}_i)] \ ,$$

   where $\hat{p}_i$ is predicted probability of outcome y_i.

The objective of the backpropagation method is to minimize the loss from prediction, which in the form of

$$\min_{\theta} J(\theta) \ ,$$

where $J(\theta)$ is the loss function, and $\theta = [\theta_1, \dots, \theta_N]$ is a vector of trainable parameters, which are the weights and biases of units in a neural network. The backpropagation method calculates gradients to adjust the weights. Many optimizing algorithms can be implemented.

1. Gradient descent
   The update rule of this method from iteration t to $t+1$ can be written as

   $$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J(\theta_t) \ ,$$

   where η is the learning rate. Gradient descent calculates gradients of loss with respect to all trainable variables, which normally all the weights in the prediction model. When the prediction model contains many nodes and layers, this method is computationally inefficient.

2. Stochastic gradient descent

   $$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J(\theta_t, \tilde{x}, \tilde{y}) \ ,$$

   where $(\tilde{x}, \tilde{y})$ are a randomly chosen subset of the whole dataset. While the gradient descent method processes the whole dataset, and it may take a long time to calculate, this method randomly picks some batch of samples. Unfortunately, we need to find the learning rate η to be able to train the prediction model efficiently.

3. AdaGrad (Adaptive Gradient)

   $$g_{t,i} = \nabla_{\theta} J(\theta_{t,i}) \ ,$$
   $$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{\sum_{s=1}^{t} g_{s,i}^2 + e}} g_{t,i} \ ,$$

   where $g_{t,i}$ is the gradient of node i in iteration t, and e is a small positive number (e.g. 10^{-8}). The denominator is an approximation of the second derivative. If the second derivative is large, the learning rate becomes small and the model may stop learning.

4. Root Mean Square Propagation (RMSProp)

   $$E[g^2]_{t+1} = \beta E[g^2]_t + (1-\beta) g_t^2 \ ,$$
   $$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t}} g_t \ ,$$

   where g_t is the gradient of the loss function with respect to θ at the iteration t, $E[g^2]_t$ is the moving average of squared gradients and β is the moving average parameter, normally equal to 0.9. This method is introduced by Geoffrey Hinton in lecture 6 of the coursera's online course. He used exponential moving averages. RMSProp keeps the moving average of the squared gradients for each weight. When updating each weight, the square root of the mean square is used as the denominator of the gradients.

5. Adam (Adaptive Moment Estimation)
mean of momentum: $m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t$.
uncentered variance: $v_t = \beta_1 v_{t-1} + (1-\beta_2)g_t^2$.

$$\hat{m_t} = \frac{m_t}{1-\beta_1^t} \ ,$$
$$\hat{v_t} = \frac{v_t}{1-\beta_2^t} \ ,$$
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v_t}+e}}\hat{m_t} \ ,$$

where $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $e = 10^{-8}$. This method is introduced by Kingma and Ba (2014). They extended the stochastic gradient method by using adaptive estimation of first and second order derivative. They claimed that Adam can deal with sparse gradients problem like AdaGrad and non-stationary objective problem like RMSProp.

2.3. Safe reinforcement learning

Safe reinforcement learning tries to maximize reinforcement signals, which are rewards or desirable outcomes, with respect to safety constraints. Garcia et al. (2015) stated that this safe RL can be approached in two ways: modifying the optimization criteria and modifying the exploration process. They categorized the optimization criteria into the four groups: (i) The worst-case criterion, (ii) The risk-sensitive criterion, (iii) The constrained criterion and (iv) Other optimization criteria. The worst-case criterion is optimal when the worst-case return is maximized. The risk-sensitive criterion adds a risk measure in the optimization criterion; for example, it may use the exponential utility function or linear combination of return and risk. The constrained criterion is to optimize the expected return that is subjected to one or more constraints. The other approach is to modify the exploration process. They considered two ways in this approach: (i) Incorporate external knowledge and (ii) Use a risk-directed exploration. For the use of a risk-directed exploration way, optimization criterion can be the same while the risk is used to adjust the probabilities of action during exploration. Saunders et al. (2017)'s HIRL scheme used external knowledge to avoid dangerous situation during the exploration process. The external knowledge is in the form of a human overseer.

2.4. One-Hot Encoding

Categorical features, which are in the form of text, cannot be directly used in the prediction model. The categorical features need to be changed into numerical type. One-Hot encoding method changes categories to binary variables. Each unique category has its representative binary variable. An example is illustrated in Table 2. Note that there are three categories in this example and we need only two columns to describe them.

Table 2: One-Hot Encoding

Before Encoded		After Encoded		
Index	Animal Type	Index	Animal type Cat	Animal type Dog
1	Cat	1	1	0
2	Dog	2	0	1
3	Bird	3	0	0
4	Cat	4	1	0

3. METHODOLOGY

In this section, we provide the methodology of this thesis. The first subsection will be about the problem setup. This subsection explains how we formulate the lending problem, how we apply the client's data to the decision making model and how the model interacts with the loan's outcome. The next subsection is about the decision making model. There are two kinds of models: (i) the benchmark model and (ii) the reinforcement based model. There are two sub-kinds of reinforcement learning based models which are the linear terms and interaction terms. The last subsection is about how to simulate the dataset for simulation data, how to prepare the real data before applying to the model and how to setup the experiments.

3.1. Problem setup

Consider a dynamic lending decision problem of an agent. At each time $\tau \in \{1, 2, \ldots, T\}$, a client comes to borrow S_τ dollars from the agent. The information about the client is represented by a vector x, which is known to the agent. Let B_τ denote the budget of the agent at time τ, and a_τ the model accuracy at time τ. Before the agent makes a decision, the agent uses the information x_τ, the budget B_τ and the accuracy a_τ to make a lending decision d_τ. If the agent decides to lend ($d_\tau = 1$), and the loan does not default, the budget becomes

$$B_{\tau+1} = B_\tau + r_\tau \ ,$$

where r_τ is the interest payment of client τ. However, if the loan defaults, the budget becomes

$$B_{\tau+1} = B_\tau - L_\tau \ ,$$

where L_τ is the loss of the defaulted loan. If the agent decides not to lend, the budget remains the same

$$B_{\tau+1} = B_\tau.$$

Then the agent uses the information observed in period τ to update the decision making policy and the next period begins. Before arrival of the first customer, the agent has initial budget amount B_0. When a customer comes, the model makes decisions and the budget changes according to the decision that the model made. If the budget becomes zero or negative, agent stops making decisions and the process is terminated.

We assume that the agent has enough loan data of one loan type. The agent can use the data of this loan type to learn how to adjust the reinforce signal (reward), how much the model learns in one period (learning rate) and the agent can find the default probability cut-off value (threshold) using this loan data. After the agent finds the hyperparameters, the agent is then given a small set of loan data for another loan type, the unfamiliar one. The agent builds a new probability prediction model and a decision making model for the unfamiliar loan type. The summary diagram is shown in Figure 5.

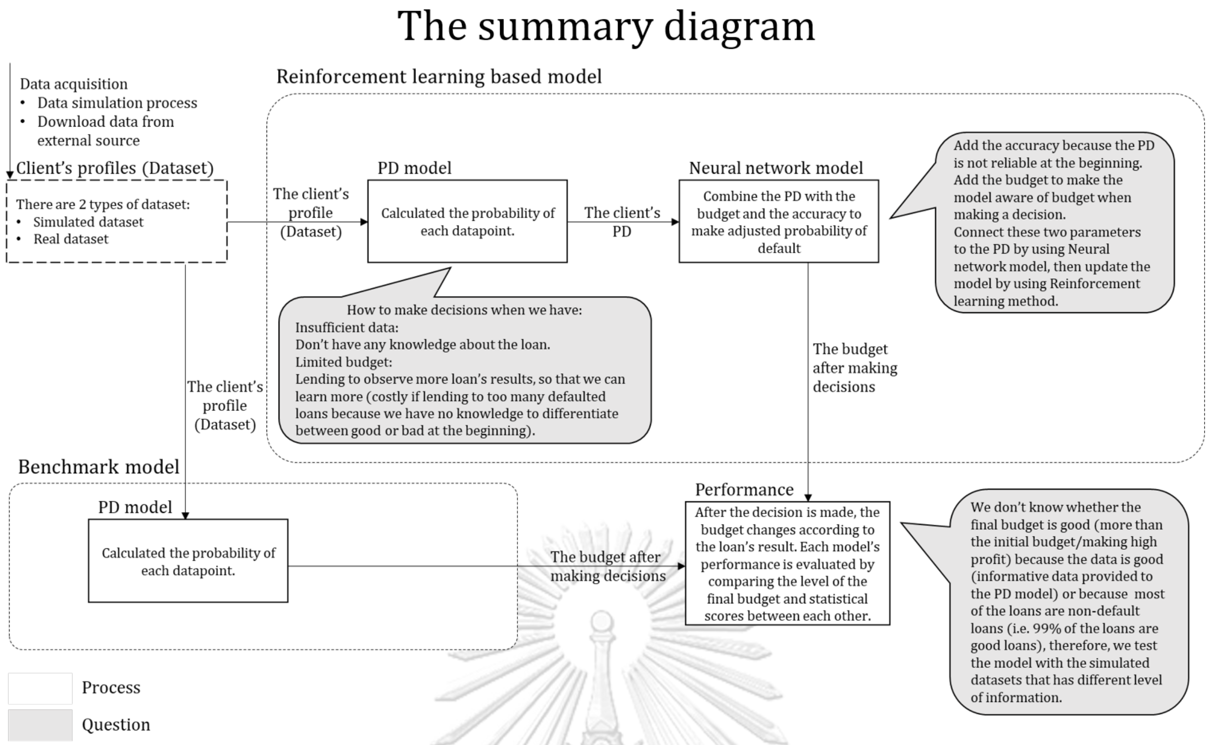

The summary diagram
Data acquisition
• Data simulation process
• Download data from external source
Client's profiles (Dataset)
There are 2 types of dataset:
• Simulated dataset
• Real dataset
Reinforcement learning based model
The client's profile (Dataset)
PD model
Calculated the probability of each datapoint.
The client's PD
Neural network model
Combine the PD with the budget and the accuracy to make adjusted probability of default
Add the accuracy because the PD is not reliable at the beginning. Add the budget to make the model aware of budget when making a decision. Connect these two parameters to the PD by using Neural network model, then update the model by using Reinforcement learning method.
How to make decisions when we have: Insufficient data: Don't have any knowledge about the loan. Limited budget: Lending to observe more loan's results, so that we can learn more (costly if lending to too many defaulted loans because we have no knowledge to differentiate between good or bad at the beginning).
The budget after making decisions
The client's profile (Dataset)
Benchmark model
PD model
Calculated the probability of each datapoint.
The budget after making decisions
Performance
After the decision is made, the budget changes according to the loan's result. Each model's performance is evaluated by comparing the level of the final budget and statistical scores between each other.
We don't know whether the final budget is good (more than the initial budget/making high profit) because the data is good (informative data provided to the PD model) or because most of the loans are non-default loans (i.e. 99% of the loans are good loans), therefore, we test the model with the simulated datasets that has different level of information.
Process
Question

Figure 5: The summary diagram

Figure 5 summarize all the process involved in the model. The first part is the dataset. We obtain the datasets by either simulating the dataset or downloading the dataset from an external source. Then the datasets are used as an input of the decision making model which are the benchmark model and reinforcement learning based models. For the reinforcement learning based model, the decision that the model makes is based on the adjusted PD which is calculated from the customer's profile, current budget and model's accuracy. However, the benchmark model only calculates PD based only on the customer's profile. We can measure the model performance by comparing the budget after the model makes decisions for every customer in the dataset. By comparing the results from the reinforcement learning based model and the benchmark model we can estimate the impact of the current budget and the accuracy on the decisions. We test the models on different variance ratios so that we can measure how the level of information from the input data affects the improvement of the model's performance when the budget and prediction accuracy are accounted for.

3.2. Decision making model

We have two decision making models in our study. The first one is the benchmark model that uses a logistic regression model to predict the probability of default based only on the borrower's features. The second model is our reinforcement learning based model (RL based model) that uses the current budget and the model accuracy, in addition to the borrower's features, to make decisions.

3.2.1. Benchmark model

The benchmark model uses the logistic regression model of the borrower's features to predict the default probability of each borrower at time τ. The probability of default is calculated by

$$P(y_\tau = 1) = \frac{1}{1+\exp(-[\omega_0+\omega_1 x_{1,\tau}+\ldots+\omega_N x_{N,\tau}])} \ ,$$

where y_τ is the repayment outcome of borrower at time τ, $x_{i,\tau}$ is the feature i of the borrower τ for $i = 1,\ldots,N$. If loan τ defaults, then $y_\tau = 1$ and $y_\tau = 0$ otherwise. The model compares the predicted probability $\hat{p_\tau}$ to a time varying threshold ϕ_τ. The model accepts the loan application of borrower τ, when $\hat{p_\tau} < \phi_\tau$ and the model rejects otherwise. The model finds the first threshold ϕ_0 from the small dataset F_m before the first borrower arrives where m is the number of data points of the small dataset. The model predicts default probability and finds the threshold from a set Φ such that the budget after m data points is maximized:

$$\phi_0 = \underset{\phi\in\Phi}{\operatorname{argmax}}\{B_m \,|\, \pi(\phi), F_m\} \ ,$$

where $\pi(\phi)$ denotes the decision making rule based on threshold ϕ and the set of thresholds Φ can be written as

$$\Phi = \{\ 1\%, 2\%, 3\%, 14\%, 15\%, 20\%, 25\%, , 45\%, 50\%\}.$$

The ϕ_0 is the threshold obtained from F_m. When the borrower arrives, the model uses ϕ_0 until the model makes n decisions. After the model makes n decisions, the model updates its parameters which are the coefficients of logistic regression ω and the threshold ϕ based on the larger set of data. In this research, we choose the number of borrowers $T = 9,900$, the initial set of data has $m = 100$ data points, the model is updated every $n = 100$ decisions and the initial budget $B_0 = 600$ is based on the larger set of data. The benchmark model evaluation process diagram is illustrated in Figure 6.

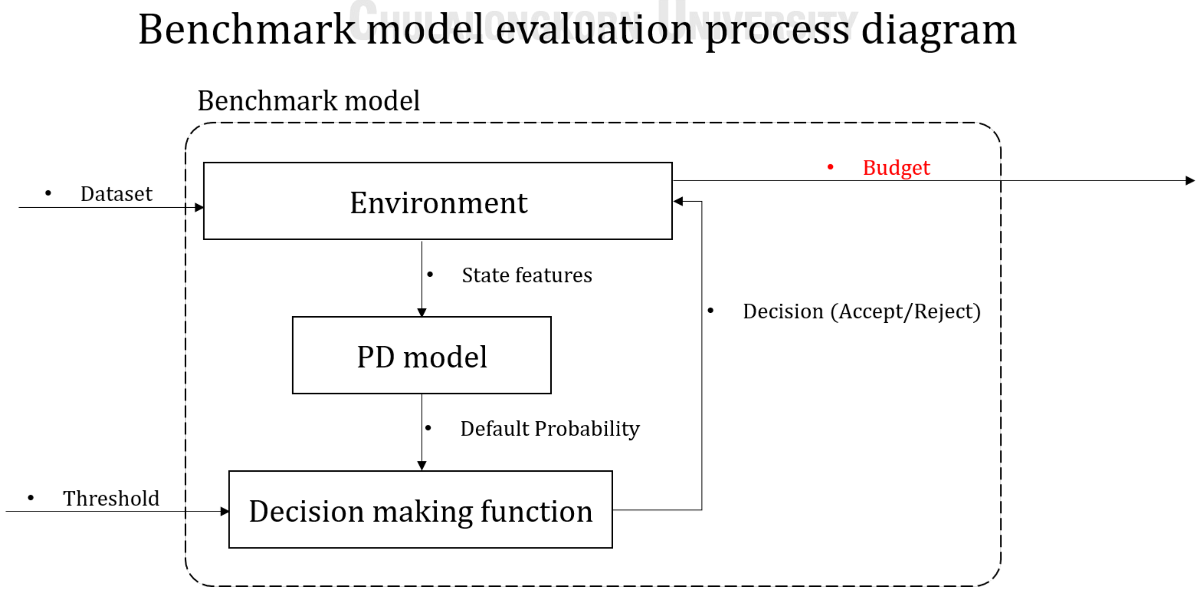


Figure 6: The process diagram for benchmark model

3.2.2. Reinforcement learning based model

We propose a decision model that accounts for the budget constraint and the accuracy of the predicted probability of default. This model uses the logistic regression model as in the benchmark model to predict the probability of default. We refer to this logistic regression model as the PD model. To make lending decisions, the model combines the predicted probability of default with the current budget and the current accuracy of the PD model to generate what we call the adjusted default probability. Then the adjusted default probability is compared with a threshold to arrive at a lending decision.

The agent finds the best hyperparameters from the loan type data that has large amount of data points. We assume that by using these hyperparameters on unfamiliar loan type, the decision making model can learn how to adjust the default probability while avoiding the budget run-out situation when model is making decisions.

The agent adjusts the reinforcement signal based on current budget B_τ to make sure that the model is aware of the budget before making any decisions. The reinforcement signal or $Reward$ adjustment parameters γ^* are:

(i) Good reward adjust

$$\gamma_g^* = 1$$

(ii) Bad reward adjust

$$\gamma_b^* = \begin{cases} \gamma_{bhigh}\left(1+\left(\tanh\left(\frac{B_\tau}{B_0}-1\right)\right)^2\right) & \text{, when } B_\tau > B_0 \\ \gamma_{blow}\left(1+\left(\tanh\left(\frac{B_\tau}{B_0}-1\right)\right)^2\right) & \text{, otherwise.} \end{cases}$$

For the symmetric case, we will set both γ_g^* and γ_b^* to 1. The adjusted reward $Reward^*$ is calculated as follows:

$$Reward^* = \begin{cases} \gamma_g^* \times Reward & \text{, when the accepted loan is a good loan} \\ \gamma_b^* \times Reward & \text{, when the accepted loan is a bad loan} \end{cases}.$$

The agent finds the good and the bad reward adjustment parameters from the set of adjusted parameters Γ which can be written as

$$\Gamma = \{\ 0.0, 0.1, 0.2, ...1.0, 1.5, 2.0\}.$$

The agent also finds the threshold ϕ from the set of thresholds Φ that maximizes the budget after making T decisions from the familiar dataset. The learning rate η controls how much weights of the network with respect to the loss gradient are adjusted. The agent finds the η that maximizes the final budget from the set of learning rates H. The set of learning rates H can be written as:

$$H = \{3\times10^{-3}, 3\times10^{-4}, 3\times10^{-5}, 3\times10^{-6}, 3\times10^{-7}\}.$$

The agent uses the hyperparameters from the familiar loan dataset in the unfamiliar loan dataset. We propose two kinds of reinforcement learning based model in this research: (i) the reinforcement learning based model without interaction terms and (ii) the reinforcement learning based model with interaction terms.

For unfamiliar loans, the agent first uses the predicted riskiness score $\hat{s_\tau}$ of each borrower τ computed from the PD model of the first m data points. The $\hat{s_\tau}$ is defined by

$$\hat{s_\tau} = \hat{\beta_0} + \hat{\beta_1} x_{1,\tau} + \ldots + \hat{\beta_N} x_{N,\tau} = ln(\frac{\hat{p_\tau}}{1-\hat{p_\tau}}) \ ,$$

where $\hat{\beta_i}$'s are the estimated parameters in the PD model. Then the agent uses the predicted riskiness score together with the budget and the model accuracy as inputs of the reinforcement learning model. The reinforcement learning model predicts the adjusted default probability. Then the model compares the adjusted default probability to the threshold obtained from the familiar loan type. If the adjusted probability is less than the threshold, then the model accepts the loan. The logistic regression model uses the state, which is the information of current client gotten from the environment, as an input and the default probability as an output.

To make sure that the default prediction model makes a budget and model's accuracy concerned policy, we add two nodes in the input layer: (i) Budget and (ii) Accuracy. The initial value of the accuracy node is the accuracy of the PD model. When the model decides to accept the loan, the accuracy goes up if the loan is a good loan and goes down otherwise. On the other hand, the accuracy stays the same if the model decides to reject the loan. The accuracy is defined by the number of accepted non-defaulted loans over the number of accepted loans, and it has the following dynamic:

$$a_{\tau+1} = \frac{a_\tau(m+\lambda_\tau) + e_{\tau+1}}{m+\lambda_\tau + |e_{\tau+1}|} \ ,$$

where a_{τ_0} is the accuracy from the PD model, a_τ is accuracy at time τ ,λ_τ is the number of the accepted loans from the first τ arrivals ($\lambda_0 = 0$) and $e_\tau = 1$ if the loan τ does not default, $e_\tau = -1$ if loan t defaults, and $e_\tau = 0$ if loan τ is rejected. So the accuracy increases each time the agent accepts a non-defaulted loan, and decreases when the agent accepts a defaulted loan.

The accuracy and the budget are transformed before they are used as inputs. The transformation functions are written as follows:

$$\tilde{a_\tau} = \begin{cases} 0 & ,\text{when } a_\tau > 0.8 \\ 1.25a_\tau - 1 & ,\text{otherwise} \end{cases} \ ,$$

$$\tilde{B_\tau} = \tanh\left(\frac{B_\tau}{B_0} - 1\right) \ ,$$

where $\tilde{a_\tau}$ is the transformed accuracy and $\tilde{B_\tau}$ is the transformed budget. The model uses the three input nodes to calculate the adjusted probability of default. Then the probability is compared to the threshold ϕ to define the action. The agent's model is illustrated in Figure 7.

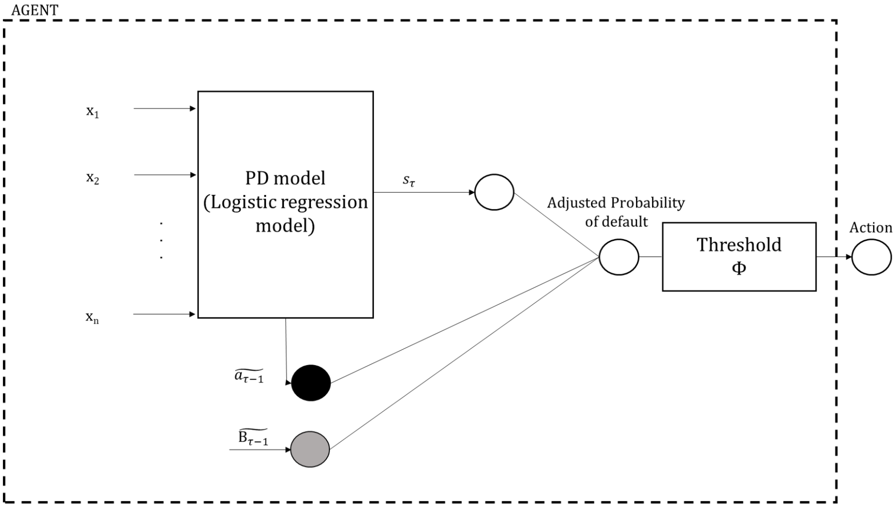


Figure 7: The agent's model

In this research, we use the policy gradient (PG) algorithm to update the model parameters. The PG algorithm finds the gradient of the loss function and optimizes the parameters based on the current and past gradients of the associated rewards. Namely, after calculating the gradients from the loss function, we compute the weighted average of the gradients using the associated rewards as the weights. Then we use the average gradient to update the parameters. The loss function used in this research is the binary cross entropy function with adam optimization technique as the optimizer.

To learn more about the effect of the budget and the past performance on the adjusted default probability, we introduce two ways of connecting the nodes: (i) linear and (ii) linear with interaction terms. The RL based model evaluation process diagram is illustrated in Figure 8.

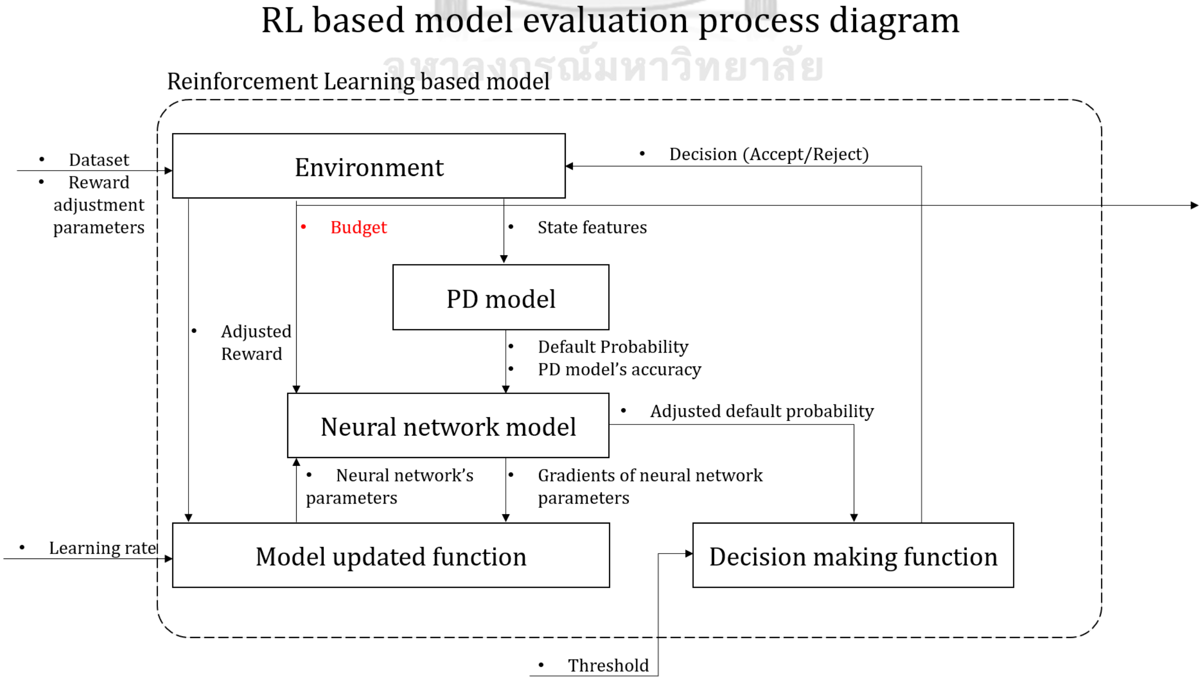


Figure 8: The process diagram for RL based model

3.2.3. Type I: Linear

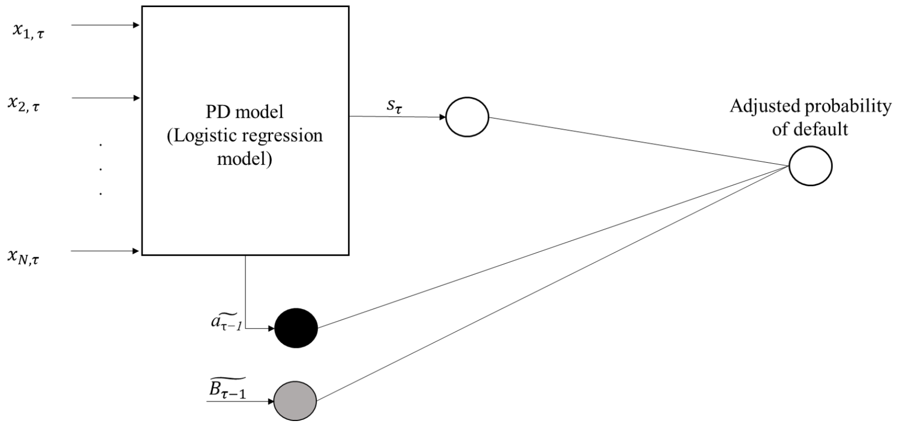


Figure 9: Type I network

In this network, the score s_τ summarizes all the client's information $x_1, ,x_N$ obtained from the PD model. This node tries to measure the riskiness of the client. We join the value of the budget node $\tilde{B}_{\tau-1}$ and the value of the accuracy node $\tilde{a}_{\tau-1}$ with s_τ using the linear relationship at the output node. The adjusted probability of default is given by

$$\tilde{p_\tau} = \frac{1}{1+e^{-(w_0+s_\tau+w_a\tilde{a}_{\tau-1}+w_b\tilde{B}_{\tau-1})}} \ ,$$

where w_0 is a constant representing an intercept term, w_b and w_a are the corresponding weights of budget and accuracy respectively. We anticipate that high values of the budget and the accuracy should adjust the probability of default downward (easier to accept loans). The starting values of w_b and w_a are equal to 0 and are updated when the model makes decisions. Figure 9 shows the network with linear connection.

3.2.4. Type II: Linear with interaction terms

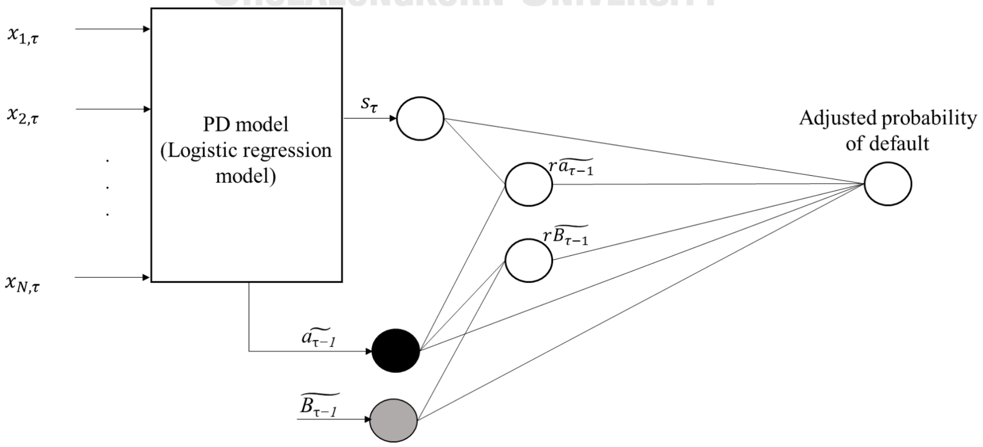

$x_{1,\tau}$
$x_{2,\tau}$
$x_{N,\tau}$
PD model (Logistic regression model)
s_τ
$r\widetilde{a_{\tau-1}}$
$r\widetilde{B_{\tau-1}}$
Adjusted probability of default
$\widetilde{a_{\tau-1}}$
$\widetilde{B_{\tau-1}}$

Figure 10: Type II network

We introduce interaction terms into the prediction model as illustrated in Figure 10. There are two additional nodes for the interaction terms. The first term is the multiplication of score and budget, and the second term is the multiplication of score and accuracy. The interaction terms are added because the effect of the riskiness on the adjusted probability of default may depend on the budget and accuracy. For example, the adjusted probability of default should depend mostly on the riskiness level when the budget and the accuracy are high, while the adjusted probability of default should be high regardless of the level of the riskiness when the budget and the accuracy are low (more difficult to accept loans when the budget is low and the model is not accurate). The adjusted probability of default can be written as

$$\tilde{p}_\tau = \frac{1}{1+e^{-(w_0+s_\tau+w_a\tilde{a}_{\tau-1}+w_b\tilde{B}_{\tau-1}+w_{sa}s_\tau\tilde{a}_{\tau-1}+w_{sb}s_\tau\tilde{B}_{\tau-1})}}.$$

3.3. Data

We first run the model with simulated data. By using simulated data, we can measure the information content available to the decision making model. We observe the model performance from the different levels of information contained in the different datasets.

3.3.1. Data simulation

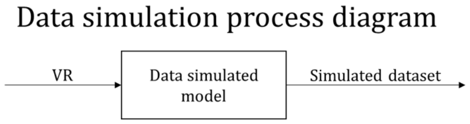
We assume that the true probability of default of each borrower τ or PD_τ is given by a logistic function of the linear combination of the set of five features

$$PD_\tau = \frac{1}{1+e^{-(c_1x_{1,\tau}+\ldots+c_5x_{5,\tau})}} ,$$

where c_i's are constant coefficients. We also assume that each feature $x_{i,\tau}$ is i.i.d. with standard normal distribution and we sample their values according to this assumption. We sample the value of each c_i randomly following the discrete uniform distribution of integers between -10 and 10. We use the computed PD_τ to sample the repayment outcome of each borrower τ. To make the data imperfect to the lender, we assume that the lending agent can observe only the first three features, namely $x_{1,\tau}, x_{2,\tau}$ and $x_{3,\tau}$. We also sample two independent uncorrelated features $x_{6,\tau}$ and $x_{7,\tau}$ following i.i.d. standard normal distribution and assume that these two features are given to the agent too. That is, the agent observes $x_{1,\tau}, x_{2,\tau}, x_{3,\tau}, x_{6,\tau}$ and $x_{7,\tau}$ but the agent does not know which features in fact have a prediction power for the probability of default and needs to learn from the data. We also make sure that the standard deviation of $\sum_{i=1}^{5} c_i x_{i,\tau}$ is equal to 1 by adjusting the constant coefficients. We provide the distribution of the PD in the Appendix (see Figures 25 - 26). We set the number of borrowers T=9,900, and the number of data points for the initial loan data m=100, and simulate the data for six independent samples. The first sample is used as the training data set (familiar loan type) for the RL model, and the other five samples are used as five testing data sets for both benchmark and RL models. To measure the information content available to the agent, we use the ratio of the variance of the linear combination of the true features observed by the agent to the variance of the linear combination of all of the true features. We call this as the variance ratio

$$VR = \frac{c_1^2+c_2^2+c_3^2}{c_1^2+c_2^2+c_3^2+c_4^2+c_5^2} .$$

Note that a high variance ratio indicates that the missing features are not significant to the probability of default prediction. The data simulation process algorithm is illustrated in Figure 11.



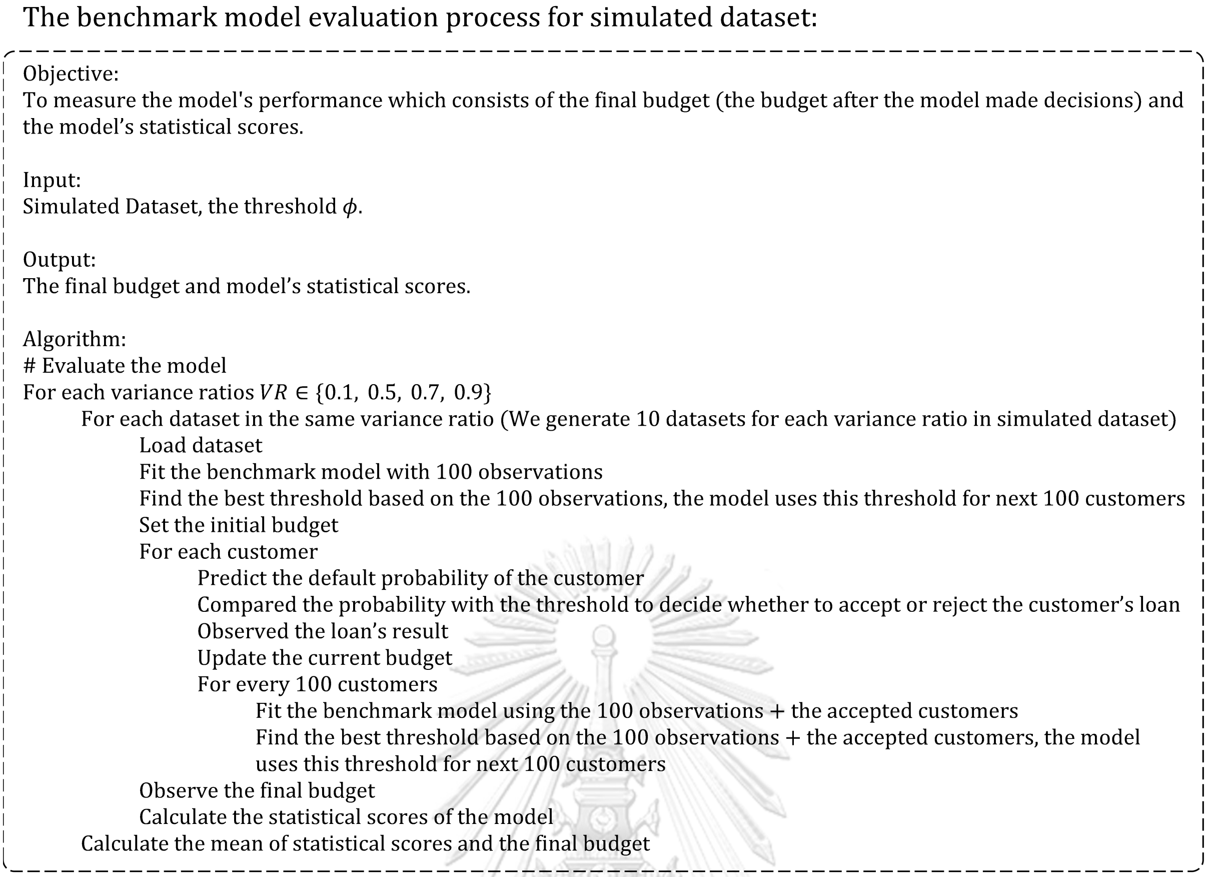
Data simulation process:

```
Objective:
To create a simulated dataset of loan data including the set of features of each customer, and his or her repayment flag (default or non-default) to match the given variance ratio.

Input:
Variance ratio ($VR$)

Output:
Simulated loan data with $N = 10{,}000$ data points, each of which includes the features $\{x_1, x_2, x_3, x_6, x_7\}$, and the repayment flag.

Algorithm :
Fix the random seed
For $VR \in \{0.1, 0.5, 0.7, 0.9\}$
    Repeat the following 10 times
        Generate 5 i.i.d. with standard normal distribution using $\mu = 0, \sigma = 1$ and $\tau = 12000$ as $x = \{x_1, \dots, x_5\}$.
        While ~$((\sum_{i=1}^{5} c_i, x_{i,\tau} = 1)$ & $(-10^{-5} < \widehat{VR} - VR < 10^{-5}))$
            Generate 5 discrete uniform distribution of integers between -10 and 10 as the coefficients $c = \{c_1, \dots, c_5\}$.
            Given the coefficients $c$, calculate the variance ratio $\widehat{VR}$.
        Given the $x$ and $c$, calculate the default probability using logistic function.
        Sampling the loan's results from the calculated default probability.
        Generate 2 i.i.d. with standard normal distribution using $\mu = 0, \sigma = 1$ and $\tau = 12000$ as $x_5, x_6$.
        Pick 5,000 datapoints for each default and non-defaulted loan.
        The simulated dataset contains $x_1, x_2, x_3, x_6, x_7$ and loan's result this dataset contains 10,000 datapoints.
```

Figure 11: The algorithm of data simulation process

3.3.2. Experiment setup for simulated data

We find the best model parameters with the training scenario. We first fit the default prediction model with m observations. Then we let the default prediction model predict default probabilities of the next n_Δ observations. The reinforcement learning model uses the PD, budget and accuracy to predict the adjusted default probabilities for the n_Δ observations. If the adjusted default probability is greater than the threshold, the model rejects the loan. The agent updates its parameters every n_Δ observations. The model makes decisions until the last borrower comes at $T = 9{,}900$ or until the agent runs out of the budget $B_\tau \leq 0$. We set the number of observations before each update $n_\Delta = 100$ and the initial budget $B_0 = 600$. The benchmark model and the RL based model evaluation process are illustrated in Figure 12 and 13, respectively.



The benchmark model evaluation process for simulated dataset:

Objective:
To measure the model's performance which consists of the final budget (the budget after the model made decisions) and the model’s statistical scores.

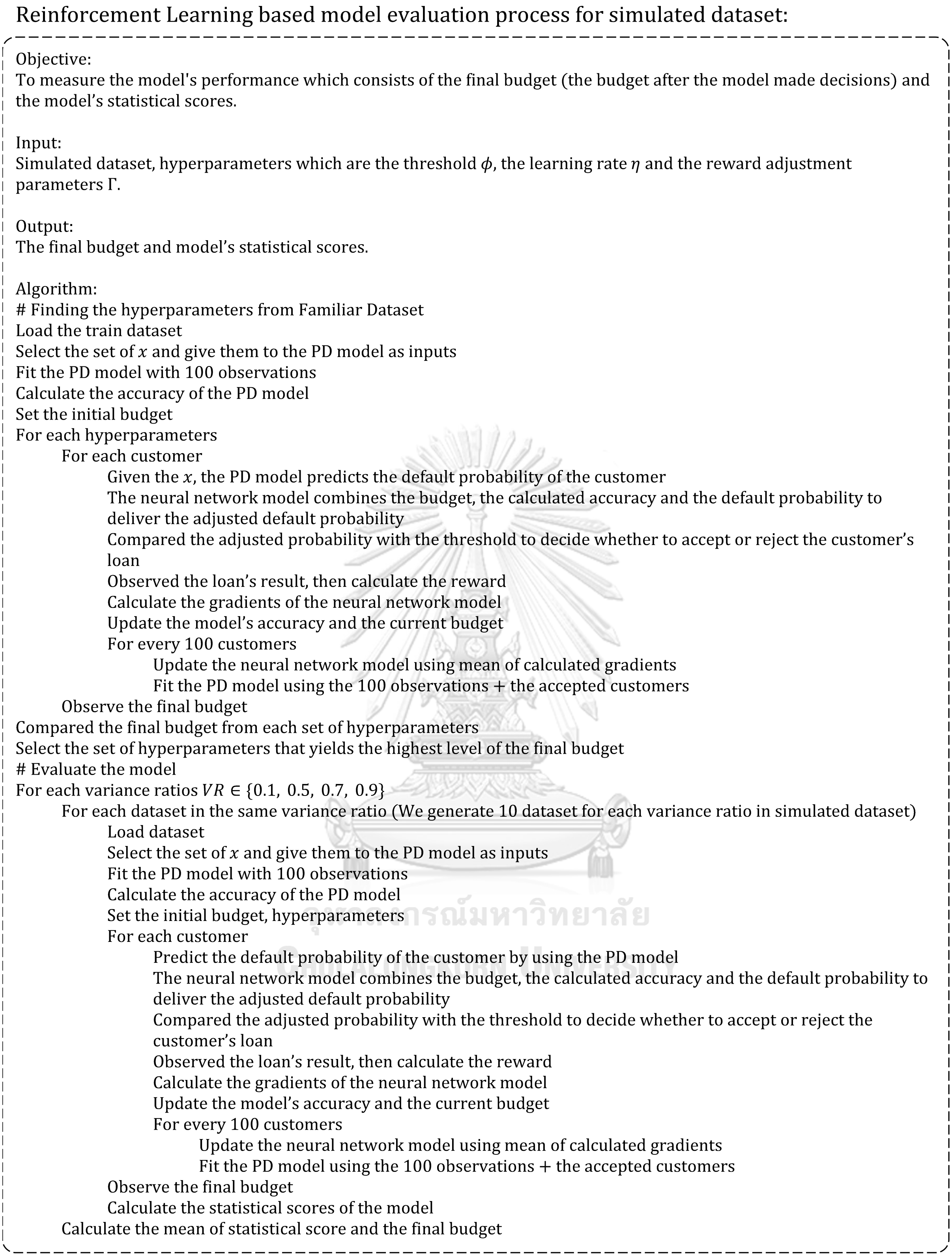
Input:
Simulated Dataset, the threshold ϕ.

Output:
The final budget and model’s statistical scores.

Algorithm:

```
# Evaluate the model
For each variance ratios $VR \in \{0.1,\ 0.5,\ 0.7,\ 0.9\}$
    For each dataset in the same variance ratio (We generate 10 datasets for each variance ratio in simulated dataset)
        Load dataset
        Fit the benchmark model with 100 observations
        Find the best threshold based on the 100 observations, the model uses this threshold for next 100 customers
        Set the initial budget
        For each customer
            Predict the default probability of the customer
            Compared the probability with the threshold to decide whether to accept or reject the customer’s loan
            Observed the loan’s result
            Update the current budget
            For every 100 customers
                Fit the benchmark model using the 100 observations + the accepted customers
                Find the best threshold based on the 100 observations + the accepted customers, the model
                uses this threshold for next 100 customers
        Observe the final budget
        Calculate the statistical scores of the model
    Calculate the mean of statistical scores and the final budget
```

Figure 12: The benchmark model evaluation process for simulated data



Reinforcement Learning based model evaluation process for simulated dataset:

Objective:
To measure the model's performance which consists of the final budget (the budget after the model made decisions) and the model's statistical scores.

Input:
Simulated dataset, hyperparameters which are the threshold ϕ, the learning rate η and the reward adjustment parameters Γ.

Output:
The final budget and model's statistical scores.

Algorithm:

```
# Finding the hyperparameters from Familiar Dataset
Load the train dataset
Select the set of $x$ and give them to the PD model as inputs
Fit the PD model with 100 observations
Calculate the accuracy of the PD model
Set the initial budget
For each hyperparameters
    For each customer
        Given the $x$, the PD model predicts the default probability of the customer
        The neural network model combines the budget, the calculated accuracy and the default probability to deliver the adjusted default probability
        Compared the adjusted probability with the threshold to decide whether to accept or reject the customer's loan
        Observed the loan's result, then calculate the reward
        Calculate the gradients of the neural network model
        Update the model's accuracy and the current budget
        For every 100 customers
            Update the neural network model using mean of calculated gradients
            Fit the PD model using the 100 observations + the accepted customers
    Observe the final budget
Compared the final budget from each set of hyperparameters
Select the set of hyperparameters that yields the highest level of the final budget
# Evaluate the model
For each variance ratios $VR \in \{0.1,\ 0.5,\ 0.7,\ 0.9\}$
    For each dataset in the same variance ratio (We generate 10 dataset for each variance ratio in simulated dataset)
        Load dataset
        Select the set of $x$ and give them to the PD model as inputs
        Fit the PD model with 100 observations
        Calculate the accuracy of the PD model
        Set the initial budget, hyperparameters
        For each customer
            Predict the default probability of the customer by using the PD model
            The neural network model combines the budget, the calculated accuracy and the default probability to deliver the adjusted default probability
            Compared the adjusted probability with the threshold to decide whether to accept or reject the customer's loan
            Observed the loan's result, then calculate the reward
            Calculate the gradients of the neural network model
            Update the model's accuracy and the current budget
            For every 100 customers
                Update the neural network model using mean of calculated gradients
                Fit the PD model using the 100 observations + the accepted customers
        Observe the final budget
        Calculate the statistical scores of the model
    Calculate the mean of statistical score and the final budget
```

Figure 13: The RL based model evaluation process for simulated data

3.3.3. Real data preparation

This thesis uses the lending club loan dataset from Kaggle (www.kaggle.com/wordsforthewise/lending-club). In this dataset, each observation contains 150 features. This data set contains monthly loan status and loan features from 2007 to 2018. We select observations that correspond to fully paid, default and completed charged off loan status. Then we reduce the features used in RL model by dropping the features that have missing values more than 80 %. We handle the missing data by imputation. Namely, we use the feature's mean values for numerical data and use the most frequent values for categorical data. Only numerical data can be provided to the neural network model, thus, categorical data are encoded by One-Hot encoding method. We plot the loan's behavior by loan types to make sure that the selected features can distinguish between the good loans and the bad loans (see Figures 13-15 in the Appendix). We also drop the features that are highly correlated (correlation greater than 0.95). We provide the correlation heatmap in the Appendix (see Figures 16).

For the training set and test set, we divide the dataset based on the loan purpose because using different loan purposes between the training set and the test set can be interpreted as the situation where we use the data from familiar business to train the model and implement it in the unfamiliar business environment. For the training set, we sample the dataset to get 1:1 ratio of good and bad loans. Good loans are fully paid loans and bad loans are the default and completed charged off loans. For the test set, we sample 1,000 observations from the dataset. We divide the features into 2 groups: (i) features used to predict the probability of default and (ii) features used to calculate the reward. The process is summarized in Figure 14.

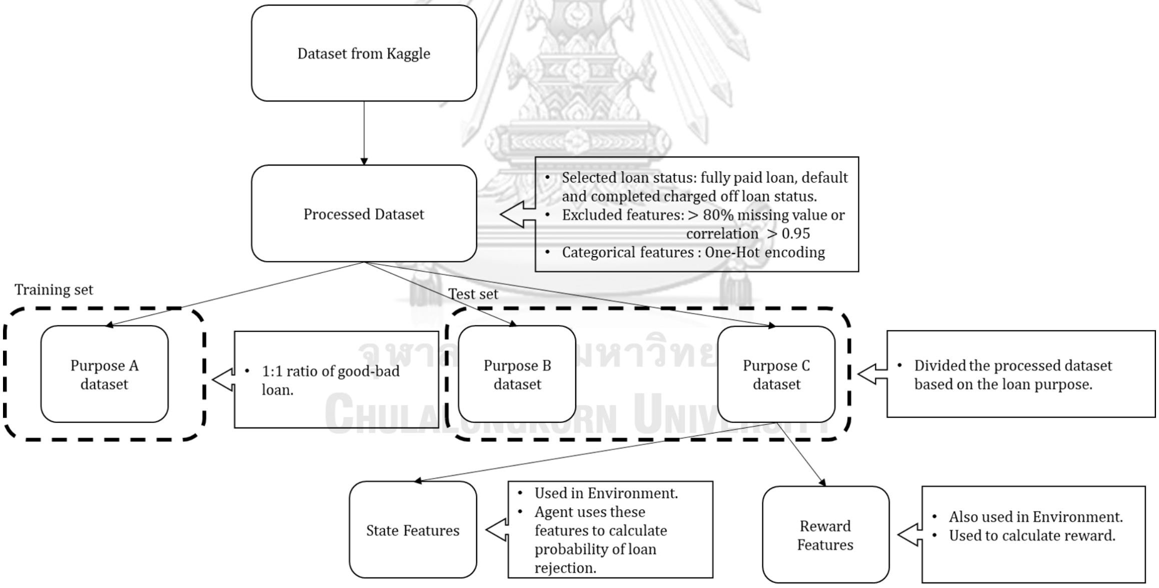
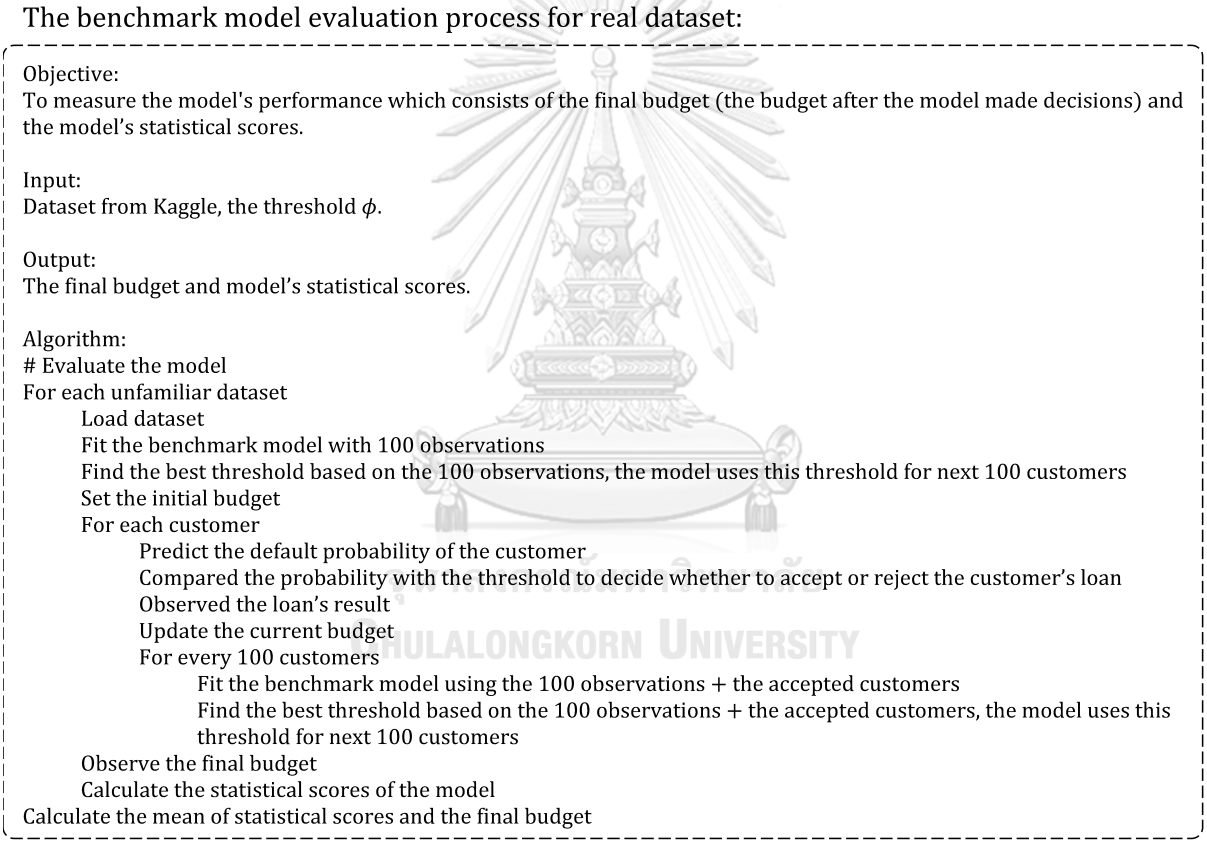

Dataset from Kaggle
Processed Dataset
• Selected loan status: fully paid loan, default and completed charged off loan status.
• Excluded features: > 80% missing value or correlation > 0.95
• Categorical features : One-Hot encoding
Training set
Test set
Purpose A dataset
• 1:1 ratio of good-bad loan.
Purpose B dataset
Purpose C dataset
• Divided the processed dataset based on the loan purpose.
State Features
• Used in Environment.
• Agent uses these features to calculate probability of loan rejection.
Reward Features
• Also used in Environment.
• Used to calculate reward.

Figure 14: Data preparation process

3.3.4. Experiment setup for real data

To simulate the situation where the agent has been familiar with one type of business but not familiar with another type of business for which the agent has to learn how to make lending decision with limited budget, we first find the best hyper parameters in the reinforcement learning model using 1,000 observations from loan purpose A. We first fit the default prediction model with 100 observations. Then we let the default prediction model predict default probabilities of the next n'_Δ observations. The reinforcement learning model uses the PD, budget and accuracy to predict the adjusted default probabilities for the n'_{Δ} observations. If the adjusted default probability is greater than the threshold, the model rejects the loan. The agent updates its parameters every n'_{Δ} observations. The model makes decisions until the last borrower comes at $T' = 900$ or until the agent runs out of the budget $B_{\tau} \leq 0$. We set the number of observations before each update $n'_{\Delta} = 100$ and the initial budget $B_0 = 300,000$. We change the values of the hyper parameters which are ϕ, γ_{bhigh}, γ_{blow} and η in the reinforcement learning model and re-do the process mentioned above, then observe the final budget value. We find the best model's hyperparameters that yield the highest final budget value and use them as in the initial parameter values for the unfamiliar loan type. To simulate the situation that we are not familiar with new environment, we further train the model using 100 observations from loan purpose B and test it with 900 observations from the same loan purpose. We repeat the same process for loan purpose C. We evaluate each model based on its out-of-sample performance. The benchmark model and the RL based model evaluation process are illustrated in Figure 15 and 16, respectively.



The benchmark model evaluation process for real dataset:

Objective:
To measure the model's performance which consists of the final budget (the budget after the model made decisions) and the model’s statistical scores.

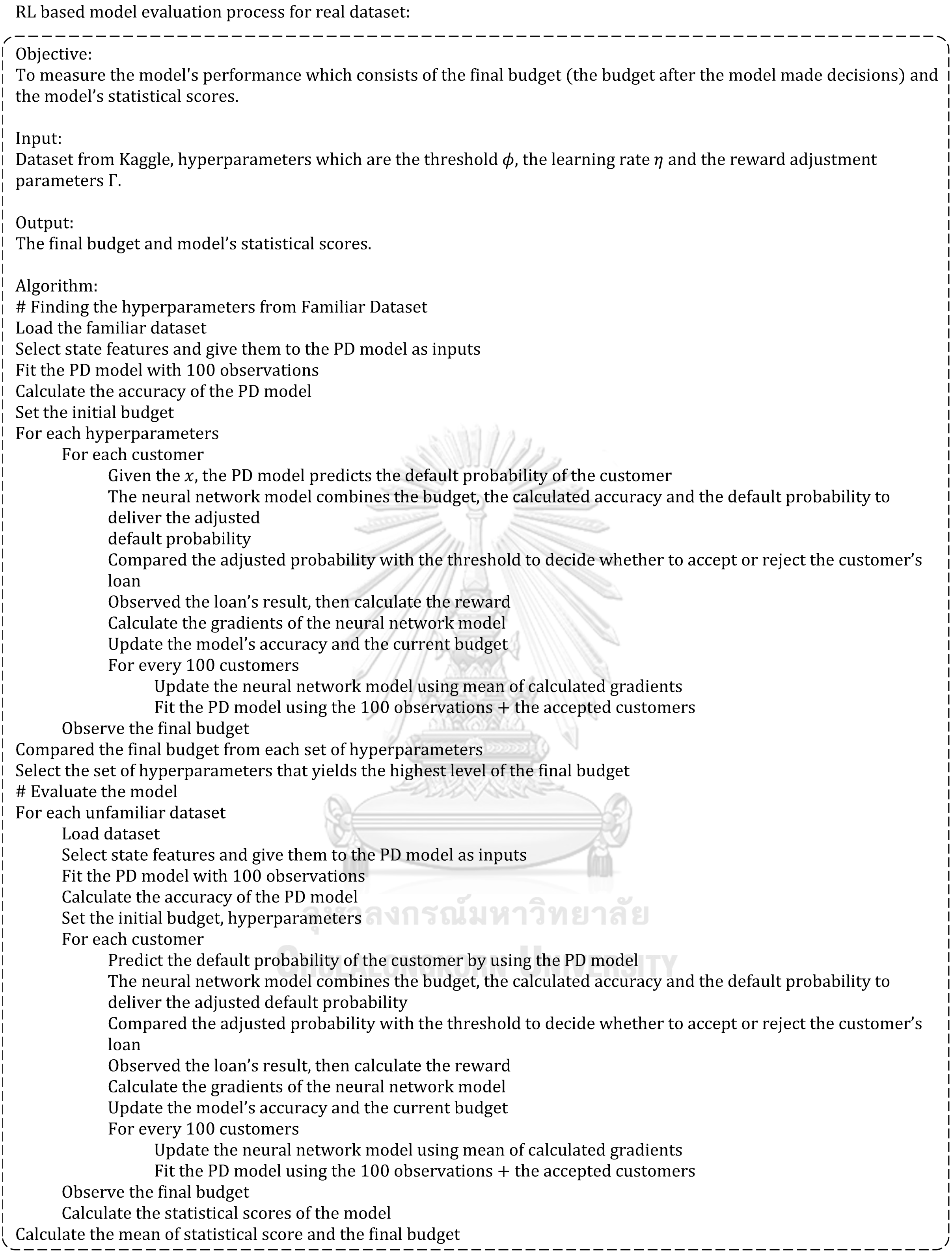
Input:
Dataset from Kaggle, the threshold ϕ.

Output:
The final budget and model’s statistical scores.

Algorithm:

```
# Evaluate the model
For each unfamiliar dataset
    Load dataset
    Fit the benchmark model with 100 observations
    Find the best threshold based on the 100 observations, the model uses this threshold for next 100 customers
    Set the initial budget
    For each customer
        Predict the default probability of the customer
        Compared the probability with the threshold to decide whether to accept or reject the customer’s loan
        Observed the loan’s result
        Update the current budget
        For every 100 customers
            Fit the benchmark model using the 100 observations + the accepted customers
            Find the best threshold based on the 100 observations + the accepted customers, the model uses this
            threshold for next 100 customers
    Observe the final budget
    Calculate the statistical scores of the model
Calculate the mean of statistical scores and the final budget
```

Figure 15: The benchmark model evaluation process for real data



```
RL based model evaluation process for real dataset:

Objective:
To measure the model's performance which consists of the final budget (the budget after the model made decisions) and
the model's statistical scores.

Input:
Dataset from Kaggle, hyperparameters which are the threshold $\phi$, the learning rate $\eta$ and the reward adjustment
parameters $\Gamma$.

Output:
The final budget and model's statistical scores.

Algorithm:
# Finding the hyperparameters from Familiar Dataset
Load the familiar dataset
Select state features and give them to the PD model as inputs
Fit the PD model with 100 observations
Calculate the accuracy of the PD model
Set the initial budget
For each hyperparameters
    For each customer
        Given the $x$, the PD model predicts the default probability of the customer
        The neural network model combines the budget, the calculated accuracy and the default probability to
        deliver the adjusted
        default probability
        Compared the adjusted probability with the threshold to decide whether to accept or reject the customer's
        loan
        Observed the loan's result, then calculate the reward
        Calculate the gradients of the neural network model
        Update the model's accuracy and the current budget
        For every 100 customers
            Update the neural network model using mean of calculated gradients
            Fit the PD model using the 100 observations + the accepted customers
    Observe the final budget
Compared the final budget from each set of hyperparameters
Select the set of hyperparameters that yields the highest level of the final budget
# Evaluate the model
For each unfamiliar dataset
    Load dataset
    Select state features and give them to the PD model as inputs
    Fit the PD model with 100 observations
    Calculate the accuracy of the PD model
    Set the initial budget, hyperparameters
    For each customer
        Predict the default probability of the customer by using the PD model
        The neural network model combines the budget, the calculated accuracy and the default probability to
        deliver the adjusted default probability
        Compared the adjusted probability with the threshold to decide whether to accept or reject the customer's
        loan
        Observed the loan's result, then calculate the reward
        Calculate the gradients of the neural network model
        Update the model's accuracy and the current budget
        For every 100 customers
            Update the neural network model using mean of calculated gradients
            Fit the PD model using the 100 observations + the accepted customers
    Observe the final budget
    Calculate the statistical scores of the model
Calculate the mean of statistical score and the final budget
```

Figure 16: The RL based model evaluation process for real data

From the setup above, we test our two models (linear, and linear with interaction terms) as summarized in Table 3. Furthermore, we use the commonly used model, which is the logistic regression model, as a benchmark. We train and test each model, then we compare each model with the logistic regression model in terms of the value and variance of the final budget value. Certain model parameters such as those in the adjusted probability of loan rejection equation can be chosen based on a validation dataset.

Table 3: List of models

Model	Network type
1	I: Linear
2	II: Linear with interaction terms
3	Logistic Regression Model (Benchmark)

4. RESULTS

In this section, we discuss the results obtained from the simulated dataset and the real dataset. For the simulated dataset, we simulate ten datasets per one variance ratio and average the results. There are three datasets with different loan purposes for the real data.

4.1. Simulated data

We implement our reinforcement learning-based model (RL model), reinforcement learning-based model with interaction terms (RL w itct model) and the benchmark model based on the ten simulated samples for each variance ratios. We run models on four scenarios that have variance ratio $VR = [0.1, 0.3, 0.7, 0.9]$. We assume that the initial budget is $B_0 = 600$, the reward from a non-defaulted loan is $r = 12$, and the loss from a defaulted loan is $L = 50$. Figure 17 reports final budget results of each scenario corresponding to each simulated sample.

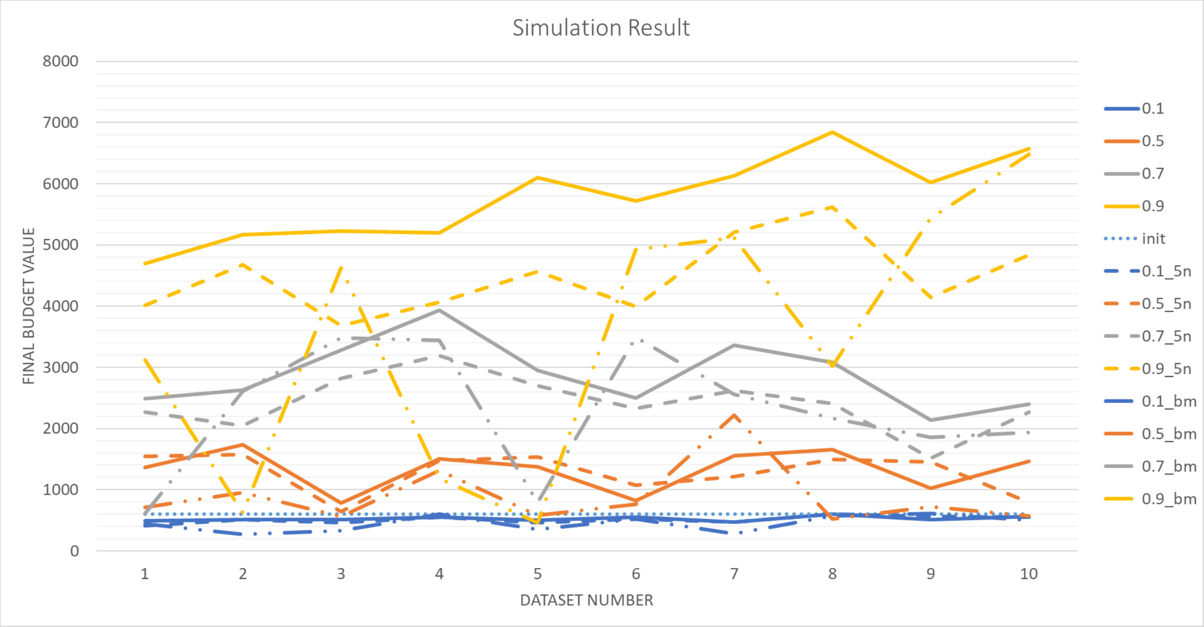


Figure 17: Models simulation result

From the graph, the solid lines, dashed lines and dash-dotted lines are the RL, RL w itct and benchmark model results respectively. The dotted line is the initial budget value, thus we want the final budget of our models to be above this line. The line's color indicates the level of variance ratio. The blue, orange, gray and yellow lines refer to 0.1, 0.5, 0.7 and 0.9 variance ratios, respectively. This graph shows the final budget for each dataset in each variance ratio (there are 10 datasets for each variance ratio). We evaluate the RL based models and the benchmark model using the simulated dataset and plot the level of final budget for each dataset. If the line is quite flat, it indicates that the final budgets do not vary much across the datasets, which implies that the result is quite robust. However, the results can very much depend on the sequence of the customer arrivals, which leads to final budgets being quite different for different datasets. More specifically, the model might underestimate the PD when the model observes few defaulted loans in the early stage of the evaluation process. The model uses parameters that underestimate the PD for the next 100 customers (until the model can update its parameters). Using these parameters may make the model accepts too many defaulted loans. Thus, the sequence of customer may affect the level of the final budget.

The final budget results seem to be better when the variance ratio is high. It can be concluded that higher information leads to higher chances of making profit. The reinforcement learning based model yields the best result. On the other hand, the benchmark model performs worst. Given that the information content measured by the variance ratio is very low (0.1), all of the models manage to survive until the last borrower.

Table 4: Average model results of each scenario

Variance ratio	Model	Number of Accepted Good Loans	Number of Accepted Bad Loans	Number of Accepted Loans	Number of Rejected Loans	Accept to Reject ratio	Final Budget
0.1	RL	2.2	2.0	4.2	9,895.8	0.0004	526.4
	RL w itct	3.3	2.6	5.9	9,894.1	0.0006	509.6
	BM	8.3	5.0	13.3	9,886.7	0.0013	449.6
0.5	RL	285.7	54.0	339.7	9,560.3	0.0355	1,328.4
	RL w itct	175.8	28.6	204.4	9,695.6	0.0211	1,279.6
	BM	207.2	43.9	251.1	9,648.9	0.0260	891.4
0.7	RL	587.4	95.5	682.9	9,217.1	0.0741	2,873.8
	RL w itct	321.7	40.9	362.6	9,537.4	0.0380	2,415.4
	BM	473.2	79.7	552.9	9,347.1	0.0592	2,293.4
0.9	RL	992.3	134.8	1,127.1	8,772.9	0.1285	5,767.6
	RL w itct	570.2	59.2	629.4	9,270.6	0.0679	4,482.4
	BM	650.0	98.0	748.0	9,152.0	0.0817	3,500.0

Table 4 reports the averages of the number of accepted good loans, the number of accepted bad loans, the number of accepted loans, the number of rejected loans, the accept to reject ratio and final budget for each value of the variance ratio and for each model. There are 4 differences level of variance ratios reported in the Table 4 which are 0.1, 0.5, 0.7 and 0.9. There are ten datasets for each variance ratio, therefore, we provide the averages in Table 4. The final budget should be better as the variance ratio increases because the model should make better decisions when the customer's profiles are more informative. The accept to reject ratio may increases when the variance ratio increases because the model should have more confident when making decision while having more informative data.

We can see that as the variance ratio increases, the average of the number of accepted loans divided by the number of rejected loans (the average of accept to reject ratio) of both RL models tend to increase, as well as the final budget. The average accept to reject ratio of the benchmark model is high compared to the reinforcement learning with interaction terms model. As a result, the benchmark model accepts bad loans more than the reinforcement with interaction terms model which make the final budget lower. In most cases, the interaction terms make the RL model more conservative. The total amount of benefit from accepting good loans is higher than the total amount of loss from accepting bad loans in the RL model without interaction terms ($n_{\text{accepted good loans}} \times r > n_{\text{accepted bad loans}} \times L$), therefore, the RL without interaction terms's final budget is higher than the interaction terms model. Unlike the benchmark model case, the difference between the amount of accepting good loans and the amount of accepting bad loans in RL without interaction terms is large enough to make the final budget high, even though the accept to reject ratio is high.

In summary, our models achieve what they are designed for. They avoid the negative budget in the low-information content environments, and they make significant long-run profits from learning in the high-information content given that the initial model accuracy is not so reliable. We provide the values of the final budget for each simulation run in the Appendix (see Figures 18 - 20).

Table 5: Average model scores of each scenario

Variance ratio	Model	Accuracy	Recall	Precision	ROC score
0.1	RL	0.50	0.00	0.29	0.50
	RL w itct	0.50	0.00	0.34	0.50
	BM	0.50	0.00	0.65	0.50
0.5	RL	0.52	0.06	0.84	0.52
	RL w itct	0.51	0.03	0.85	0.51
	BM	0.52	0.04	0.79	0.52
0.7	RL	0.55	0.12	0.86	0.55
	RL w itct	0.53	0.07	0.89	0.53
	BM	0.54	0.10	0.85	0.54
0.9	RL	0.59	0.20	0.88	0.59
	RL w itct	0.55	0.12	0.91	0.55
	BM	0.56	0.13	0.87	0.56

The accuracy, recall, precision and ROC scores for each model are reported in Table 5. The interaction terms could make the precision score higher. We can see that the benchmark model and both RL models have accuracy around 0.5, which means these models only predict correctly 50% of the time. The model's accuracy increases when the variance ratio increases. The recall for both RL and benchmark models is quite low for all scenarios. It could be interpreted that only a small portion of the non-defaulted loans are accepted. However, as the loss is much larger than the profit, a low recall is sensible. The precision and ROC scores also tend to increase as the variance ratio increases. This suggests that both RL and benchmark models can somewhat differentiate non-defaulted and defaulted loans in the right direction. The RL model with interaction terms accepts less loan than the RL based model without interaction terms. Moreover, the average of accept to rejected ratio is also lower. On the contrary, the precision score is higher. We could interpret that the interaction terms make the adjusted probability of default to be higher, therefore, it is harder for the model to accept the loan.

Table 6: Reinforcement learning's parameters

Variance ratio	Model	w_b	w_a	w_0	w_{sb}	w_{sa}	ϕ	γ_{bhigh}	γ_{blow}
0.1	RL	-0.029	-0.029	-0.028	-	-	0.84	0	1
	RL w itct	-0.029	-0.030	-0.029	-0.029	-0.028	0.84	0.6	1
0.5	RL	-0.015	0.004	-0.016	-	-	0.84	0	1
	RL w itct	-0.022	-0.028	-0.029	-0.023	-0.013	0.84	0.6	1
0.7	RL	-0.013	0.010	-0.013	-	-	0.84	0	1
	RL w itct	-0.014	-0.027	-0.030	-0.022	-0.008	0.84	0.6	1
0.9	RL	-0.011	0.009	-0.012	-	-	0.84	0	1
	RL w itct	-0.007	-0.026	-0.030	-0.020	-0.002	0.84	0.6	1

Table 6 shows the weight coefficients of both RL models at the end of each scenario. From the table, the negative w_b values indicates that the agent should be less aggressive in lending when the budget and/or the accuracy is low. The hyperparameters of both RL models (ϕ, γ_{bhigh} and γ_{blow}) are the same except γ_{bhigh} that is the penalty when the budget is high.

Our RL based models seem to act conservatively rather than aggressively because penalty parameters γ_{blow} and γ_{bhigh} are large. The loss from each defaulted loan is high compared to profit from each loan's interest, which is reflected in high penalty. As a result, our RL based models seem to reject most of the loans and have low accuracy. By adding the interaction terms, we can make the model more conservative. It is harder for the model to accept the loan because the more negative value of w_{sb} and w_{sa} make the adjusted probability of default higher. In summary, the RL with linear terms performs the best. The model accepts more good loans and less bad loans compared to the other models. The model's parameters make the model obtain the profit from accepting good loans bigger than the loss from accepting bad loans, thus leading to a higher final budget. This model can balance between the exploitation of model's current parameters and exploration of new kind of customers better than the other models.

4.2. Real data

In this subsection, we discuss the results from applying the models with the real dataset. One dataset is used to find the hyperparameters. The other two are used for evaluating the decision making model.

Table 7: Model results of each scenario

Dataset	Model	Number of Accepted Good Loans	Number of Accepted Bad Loans	Number of Accepted Loans	Number of Rejected Loans	Accept to Reject ratio	Final Budget
Credit card (train)	RL	55	12	67	833	0.08	339,159.85
	RL w itct	55	12	67	833	0.08	339,159.85
Debt consolidation	RL	52	17	69	831	0.08	272,190.21
	RL w itct	52	17	69	831	0.08	272,190.21
Home improvement	RL	96	16	112	788	0.14	355,294.92
	RL w itct	96	16	112	788	0.14	355,294.92

Table 7 reports the number of accepted good loans, the number of accepted bad loans, the number of accepted loans, the number of rejected loans, the accept to reject ratio and final budget for each value of the variance ratio and for each model. Both RL models get the same results. In the training set which is the credit card dataset, both RL models can make the final budget higher than the initial budget ($B_0 = 300,000$). Both RL models also make profit in the home improvement dataset. Unfortunately, these models cannot make profit in the debt consolidation dataset. Compared to the accept to reject ratios from the simulated data's results, the accept to reject ratios of both RL models are higher than 0.1 variance ratio case but lower than 0.5 variance ratio case. We may interpret that the quality of the real data is low and the variance ratio of the real data may be around 0.4.

Table 8: Model scores of each scenario

Data	Model	Accuracy	Recall	Precision	ROC score
Credit card (train)	RL	0.56	0.12	0.82	0.55
	RL w itct	0.56	0.12	0.82	0.55
Debt consolidation	RL	0.54	0.12	0.75	0.54
	RL w itct	0.54	0.12	0.75	0.54
Home improvement	RL	0.59	0.21	0.86	0.59
	RL w itct	0.59	0.21	0.86	0.59

The accuracy, recall, precision and ROC scores for each model are shown in Table 8. From Table 7, both models's final budgets are less than the initial budget in debt consolidation dataset, therefore, the precision score is lower than the other dataset. Compared to the results from simulated data, the recall scores from the debt consolidation dataset of both models are higher than the recall scores from the 0.1 variance ratio case but lower than the recall scores from the 0.5 case. The recall scores from the others dataset seem to be higher than the recall scores from 0.5 variance ratio case. Hence, the quality of the features in the debt consolidation dataset may be lower than others dataset.

Table 9: Reinforcement learning's parameters

Data	Model	w_b	w_a	w_0	w_{sb}	w_{sa}	ϕ	γ_{bhigh}	γ_{blow}
Credit card (train)	RL	0.0004	0.0013	-0.0013	-	-	0.25	1.4	0.6
	RL w itct	0.0004	0.0010	-0.0005	-0.0008	-0.0010	0.25	1.4	0.6
Debt consolidation	RL	0.0015	0.0016	-0.0016	-	-	0.25	1.4	0.6
	RL w itct	0.0014	0.0013	-0.0023	-0.0005	-0.0012	0.25	1.4	0.6
Home improvement	RL	0.0001	0.0010	-0.0007	-	-	0.25	1.4	0.6
	RL w itct	0.0001	0.0001	-0.0007	-0.0014	-0.0007	0.25	1.4	0.6

We believe that the hyperparameters obtained from the training set are the best parameters that can explain the features. Also, the model can make good decisions when the features have enough information by using the best parameters. Therefore, the features do not have enough information in the debt consolidation dataset. Both models yield the same result because each models's weights are similar and the interaction terms's weights, w_{sb} and w_{sa}, are low, hence the small interaction terms do not affect the adjusted probability of default.

In the real data scenarios, using the features from familiar situation in unfamiliar situations can lead to low performance. The set of features that can explain the characteristic of some kind of loans may not hold enough information about the other kind of loans. Using the same set of features cannot guarantee performance of the decision making model. Thus, both RL based models accept only small amounts of loans. Moreover, few test data points can make RL model without interaction terms model and RL with interaction terms model yield similar result because both RL models have few chances to update its parameters. We may conclude that the customer profile of the credit card loan type is different from the debt consolidation loan type because the debt consolidation process is occurred after the customer cannot pay the debt according to the original plan. Thus, the customer credit profile of the customer in debt consolidation loan type dataset should be worse than the customer in the credit card loan type dataset. The model may need more information (features) to improve the lending policy.

By using the model's weights in Table 9, the models achieve the statistical scores and the final budget shown in Table 7 and Table 8. The negative values of w_{sb} and w_{sa} make the PD higher, thus making the RL model with interaction terms rejects the loan easier than the RL model with only the linear terms. Because of the effect of the interaction terms, the RL with interaction terms accepts less loans that have PD around the threshold making the model's precision score higher than the precision score of the RL with linear terms. Unfortunately, rejecting more loans make the RL with interaction terms has fewer chance to make profit. Thus, leading to lower a final budget and a lower accuracy score.

5. CONCLUSION

We study the lending decision making problem for unfamiliar loan types. In this problem, the lending agent initially has a low-accuracy model for predicting the default probability and has to learn about the relationship between the borrower's features and the probability of default though lending. One difficulty is that the agent has a limited budget. We propose a reinforcement learning based decision making model that accounts for the model accuracy and the budget constraint. We test the model out-of-sample based on simulated data and real data. For the simulated data, we compare the results with the traditional logistic regression model. The result shows that our RL based models outperform the logistic regression model. Furthermore, our models can avoid losing all the budget when the information content available to the lender is low and hence learning is limited, and our models can generate significant profits from learning through lending when there is sufficient information associated with each borrower. Moreover, we also find that by adding the interaction terms, we can improve the model's precision score and make the model more conservative. For the real data, our model performs well in some type of loans. The reason that the model cannot perform well may be because the features used in the PD model cannot predict the risk of the borrowers well (as in the results from the simulated data with variance ratio of 0.1). Results of this study suggest that the current budget and the model accuracy level are important factors that lenders should account for before they make a lending decision. The lenders should act less aggressively when the current budget is low to avoid losing the money and more aggressively otherwise to learn more about the customers. The accuracy also prevents the model from making unsafe actions when its level is low. Therefore, when the lenders have low budget and low prediction accuracy, they should be very confident about the customers before they decide to lend out the money. The performance of this lending strategy is improved when the data about the customers contain more predictive power.

REFERENCES

1 Barto, A. G., Sutton, R. S., and Brouwer, P. S. Associative search network: a reinforcement learning associative memory. Biological cybernetics, 40(3):201–211, 1981.

2 Berkenkamp, F., Turchetta, M., Schoellig, A., and Krause, A. Safe model-based reinforcement learning with stability guarantees. In Advances in neural information processing systems, 908–918, 2017.

3 Chi, G. and Zhang, Z. Multi criteria credit rating model for small enterprise using a nonparametric method. Sustainability, 9:1834, 2017.

4 Fan, J. and Li, W. Safety-guided deep reinforcement learning via online gaussian process estimation. arXiv preprint arXiv:1903.02526, 2019.

5 Fantazzini, D. and Figini, S. Random survival forests models for sme credit risk measurement. Methodology and computing in applied probability, 11(1):29–45, 2009.

6 Garcıa, J. and Fernández, F. A comprehensive survey on safe reinforcement learning. Journal of machine learning research, 16(1):1437–1480, 2015.

7 Hans, A., Schneegaß, D., Schäfer, A. M., and Udluft, S. Safe exploration for reinforcement learning. In ESANN, 143–148, 2008.

8 Harris, T. Credit scoring using the clustered support vector machine. Expert systems with applications, 42(2):741–750, 2015.

9 Junges, S., Jansen, N., Dehnert, C., Topcu, U., and Katoen, J.-P. Safety-constrained reinforcement learning for mdps. In International conference on tools and algorithms for the construction and analysis of systems, 130–146. Springer, 2016.

10 Leike, J., Martic, M., Krakovna, V., Ortega, P. A., Everitt, T., Lefrancq, A., Orseau, L., and Legg, S. Ai safety gridworlds. arXiv preprint arXiv:1711.09883, 2017.

11 Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. Safe and effcient off-policy reinforcement learning. In Advances in neural information processing systems, 1054–1062, 2016.

12 Saunders, W., Sastry, G., Stuhlmueller, A., and Evans, O. Trial without error: towards safe reinforcement learning via human intervention. In Proceedings of the 17th international conference on autonomous agents and multiAgent systems, 2067–2069. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

13 Serrano-Cuevas, J., Morales, E. F., Hernandez-Leal, P., Bloembergen, D., and Kaisers, M. Learning on a budget using distributional rl. In ALA workshop at FAIM, volume 6, 2018.

14 Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3-4):229–256, 1992.

APPENDIX

Table 10: List of features used in the analysis and their description

Features name	Description
annual inc	The self-reported annual income provided by the borrower during registration.
application type	Indicates whether the loan is an individual application or a joint application with two co-borrowers.
delinq 2yrs	The Number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years.
empLength	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
fico range high	The upper boundary range the borrower's FICO at loan origination belongs to.
installment	The monthly payment owed by the borrower if the loan originates.
int rate	Interest Rate on the loan.
loan status	The status of the loan.
mort acc	Number of mortgage accounts.
open acc	The number of open credit lines in the borrower's credit file.
pub rec	Number of derogatory public records.
pub rec bankruptcies	Number of public record bankruptcies.
revol bal	Total credit revolving balance.
sub grade	LC assigned loan subgrade.
tax liens	Number of tax liens.
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
total acc	The total number of credit lines currently in the borrower's credit file.
total pymnt	Payments received to date for total amount funded.
total rec int	Interest received to date.
total rec prncp	Principal received to date.

Table 11: List of available features in the lending club loan dataset and their description

Features name	**Description**
acc open past 24mths	Number of trades opened in past 24 months.
acceptD	The date which the borrower accepted the offer
accNowDelinq	The number of accounts on which the borrower is now delinquent.
accOpenPast24Mths	Number of trades opened in past 24 months.
addrState	The state provided by the borrower in the loan application
all util	Balance to credit limit on all trades
annual inc joint	The combined self-reported annual income provided by the co-borrowers during registration
annualInc	The self-reported annual income provided by the borrower during registration.
application type	Indicates whether the loan is an individual application or a joint application with two co-borrowers.
avg cur bal	Average current balance of all accounts
bc open to buy	Total open to buy on revolving bankcards.
bcOpenToBuy	Total open to buy on revolving bankcards.
bcUtil	Ratio of total current balance to high credit/credit limit for all bankcard accounts.
chargeoff within 12 mths	Number of charge-offs within 12 months
collection recovery fee	post charge off collection fee
collections 12 mths ex med	Number of collections in 12 months excluding medical collections
creditPullD	The date LC pulled credit for this loan
debt settlement flag date	The most recent date that the Debt Settlement Flag has been set
deferral term	Amount of months that the borrower is expected to pay less than the contractual monthly payment amount due to a hardship plan
delinq2Yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
delinqAmnt	The past-due amount owed for the accounts on which the borrower is now delinquent.
desc	Loan description provided by the borrower
earliestCrLine	The date the borrower's earliest reported credit line was opened

Table 11 (cont.): List of available features in the lending club loan dataset and their description

Features name	Description
effective int rate	The effective interest rate is equal to the interest rate on a Note reduced by Lending Club's estimate of the impact of uncollected interest prior to charge off.
emp title	The job title supplied by the Borrower when applying for the loan.*
empLength	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
expD	The date the listing will expire
expDefaultRate	The expected default rate of the loan.
ficoRangeHigh	The upper boundary range the borrower's FICO at loan origination belongs to.
ficoRangeLow	The lower boundary range the borrower's FICO at loan origination belongs to.
funded amnt	The total amount committed to that loan at that point in time.
funded amnt inv	The total amount committed by investors for that loan at that point in time.
fundedAmnt	The total amount committed to that loan at that point in time.
grade	LC assigned loan grade
hardship amount	The interest payment that the borrower has committed to make each month while they are on a hardship plan
hardship dpd	Account days past due as of the hardship plan start date
hardship end date	The end date of the hardship plan period
hardship flag	Flags whether or not the borrower is on a hardship plan
hardship last payment amount	The last payment amount as of the hardship plan start date
hardship length	The number of months the borrower will make smaller payments than normally obligated due to a hardship plan
hardship loan status	Loan Status as of the hardship plan start date
hardship payoff balance amount	The payoff balance amount as of the hardship plan start date

Table 11 (cont.): List of available features in the lending club loan dataset and their description

Features name	**Description**
hardship reason	Describes the reason the hardship plan was offered
hardship start date	The start date of the hardship plan period
hardship type	Describes the hardship plan offering
id	A unique LC assigned ID for the loan listing.
il util	Ratio of total current balance to high credit/credit limit on all install acct
ils exp d	wholeloan platform expiration date
inq fi	Number of personal finance inquiries
inq last 12m	Number of credit inquiries in past 12 months
inqLast6Mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
installment	The monthly payment owed by the borrower if the loan originates.
int rate	Interest Rate on the loan.
issue d	The month which the loan was funded
last credit pull d	The most recent month LC pulled credit for this loan
last fico range high	The upper boundary range the borrower's last FICO pulled belongs to.
last fico range low	The lower boundary range the borrower's last FICO pulled belongs to.
last pymnt amnt	Last total payment amount received
last pymnt d	Last month payment was received
listD	The date which the borrower's application was listed on the platform.
loan status	The status of the loan.
loan status	Current status of the loan
max bal bc	Maximum current balance owed on all revolving accounts
member id	A unique LC assigned Id for the borrower member.
memberId	A unique LC assigned Id for the borrower member.
mo sin old il acct	Months since oldest bank installment account opened
mo sin old rev tl op	Months since oldest revolving account opened
mo sin rcnt rev tl op	Months since most recent revolving account opened
mo sin rcnt tl	Months since most recent account opened
mort acc	Number of mortgage accounts.
msa	Metropolitan Statistical Area of the borrower.

Table 11 (cont.): List of available features in the lending club loan dataset and their description

Features name	Description
mths since last delinq	The number of months since the borrower's last delinquency.
mths since last major derog	Months since most recent 90-day or worse rating
mths since last record	The number of months since the last public record.
mths since oldest il open	Months since oldest bank installment account opened
mths since rcnt il	Months since most recent installment accounts opened
mths since recent bc	Months since most recent bankcard account opened.
mths since recent bc dlq	Months since most recent bankcard delinquency
mths since recent inq	Months since most recent inquiry.
mths since recent revol delinq	Months since most recent revolving delinquency.
mthsSinceLastDelinq	The number of months since the borrower's last delinquency.
mthsSinceLastRecord	The number of months since the last public record.
mthsSinceMostRecentInq	Months since most recent inquiry.
mthsSinceRecentBc	Months since most recent bankcard account opened.
mthsSinceRecentLoanDelinq	Months since most recent personal finance delinquency.
mthsSinceRecentRevolDelinq	Months since most recent revolving delinquency.
next pymnt d	Next scheduled payment date
num accts ever 120 pd	Number of accounts ever 120 or more days past due
num actv bc tl	Number of currently active bankcard accounts
num actv rev tl	Number of currently active revolving trades
num bc sats	Number of satisfactory bankcard accounts
num bc tl	Number of bankcard accounts
num il tl	Number of installment accounts
num op rev tl	Number of open revolving accounts
num rev accts	Number of revolving accounts
num rev tl bal gt 0	Number of revolving trades with balance greater than 0
num sats	Number of satisfactory accounts
num tl 120dpd 2m	Number of accounts currently 120 days past due (updated in past 2 months)
num tl 30dpd	Number of accounts currently 30 days past due (updated in past 2 months)

Table 11 (cont.): List of available features in the lending club loan dataset and their description

Features name	**Description**
num tl 90g dpd 24m	Number of accounts 90 or more days past due in last 24 months
num tl op past 12m	Number of accounts opened in past 12 months
open acc	The number of open credit lines in the borrower's credit file.
open acc 6m	Number of open trades in last 6 months
open act il	Number of currently active installment trades
open il 12m	Number of installment accounts opened in past 12 months
open il 24m	Number of installment accounts opened in past 24 months
open rv 12m	Number of revolving trades opened in past 12 months
open rv 24m	Number of revolving trades opened in past 24 months
openAcc	The number of open credit lines in the borrower's credit file.
orig projected additional accrued interest	The original projected additional interest amount that will accrue for the given hardship payment plan as of the Hardship Start Date. This field will be null if the borrower has broken their hardship payment plan.
out prncp	Remaining outstanding principal for total amount funded
out prncp inv	Remaining outstanding principal for portion of total amount funded by investors
pct tl nvr dlq	Percent of trades never delinquent
percentBcGt75	Percentage of all bankcard accounts greater than 75 percent of limit.
policy code	”publicly available policy code=1
pub rec	Number of derogatory public records.
pub rec bankruptcies	Number of public record bankruptcies.
purpose	A category provided by the borrower for the loan request.
pymnt plan	Indicates if a payment plan has been put in place for the loan
recoveries	post charge off gross recovery
reviewStatusD	The date the loan application was reviewed by LC
revol bal	Total credit revolving balance.

Table 11 (cont.): List of available features in the lending club loan dataset and their description

Features name	**Description**
revolBal	Total credit revolving balance
sec app chargeoff within 12 mths	Number of charge-offs within last 12 months at time of application for the secondary applicant
sec app collections 12 mths ex med	Number of collections within last 12 months excluding medical collections at time of application for the secondary applicant
sec app earliest cr line	Earliest credit line at time of application for the secondary applicant
sec app fico range high	FICO range (low) for the secondary applicant
sec app fico range low	FICO range (high) for the secondary applicant
sec app inq last 6mths	Credit inquiries in the last 6 months at time of application for the secondary applicant
sec app mort acc	Number of mortgage accounts at time of application for the secondary applicant
sec app mths since last major derog	Months since most recent 90-day or worse rating at time of application for the secondary applicant
sec app num rev accts	Number of revolving accounts at time of application for the secondary applicant
sec app open acc	Number of open trades at time of application for the secondary applicant
sec app open act il	Number of currently active installment trades at time of application for the secondary applicant
sec app revol util	Ratio of total current balance to high credit/credit limit for all revolving accounts
serviceFeeRate	Service fee rate paid by the investor for this loan.
settlement amount	The loan amount that the borrower has agreed to settle for
settlement date	The date that the borrower agrees to the settlement plan
settlement percentage	The settlement amount as a percentage of the payoff balance amount on the loan
settlement term	The number of months that the borrower will be on the settlement plan
sub grade	LC assigned loan subgrade.
tax liens	Number of tax liens.
term	The number of payments on the loan. Values are in months and can be either 36 or 60.

Table 11 (cont.): List of available features in the lending club loan dataset and their description

Features name	**Description**
title	The loan title provided by the borrower
tot coll amt	Total collection amounts ever owed
tot cur bal	Total current balance of all accounts
tot hi cred lim	Total high credit/credit limit
total acc	The total number of credit lines currently in the borrower's credit file.
total bal ex mort	Total credit balance excluding mortgage
total bal il	Total current balance of all installment accounts
total bc limit	Total bankcard high credit/credit limit
total cu tl	Number of finance trades
total il high credit limit	Total installment high credit/credit limit
total pymnt	Payments received to date for total amount funded.
total pymnt inv	Payments received to date for portion of total amount funded by investors
total rec int	Interest received to date.
total rec late fee	Late fees received to date
total rec prncp	Principal received to date.
total rev hi lim	Total revolving high credit/credit limit
totalAcc	The total number of credit lines currently in the borrower's credit file
totalBalExMort	Total credit balance excluding mortgage
totalBcLimit	Total bankcard high credit/credit limit
url	URL for the LC page with listing data.
zip code	The first 3 numbers of the zip code provided by the borrower in the loan application.

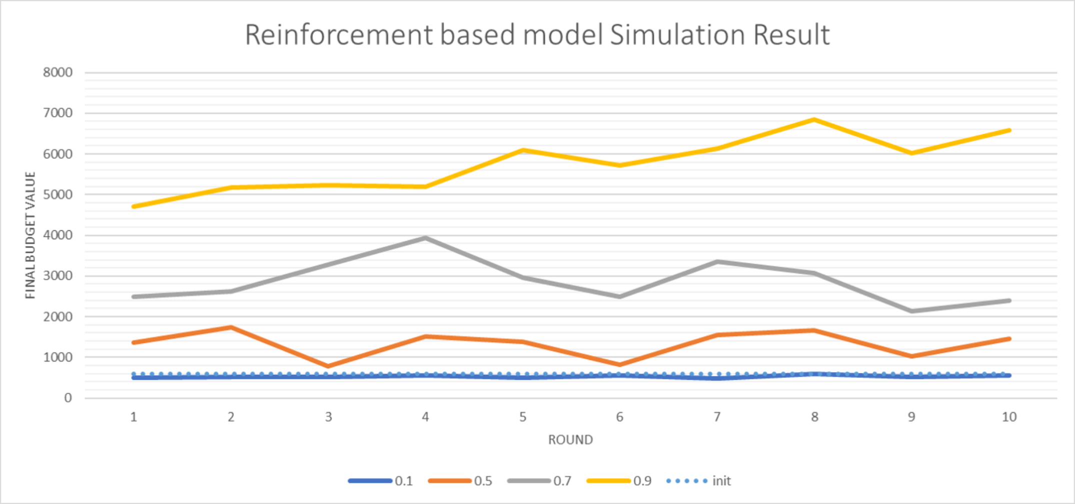


Figure 18: Reinforcement based model simulation result

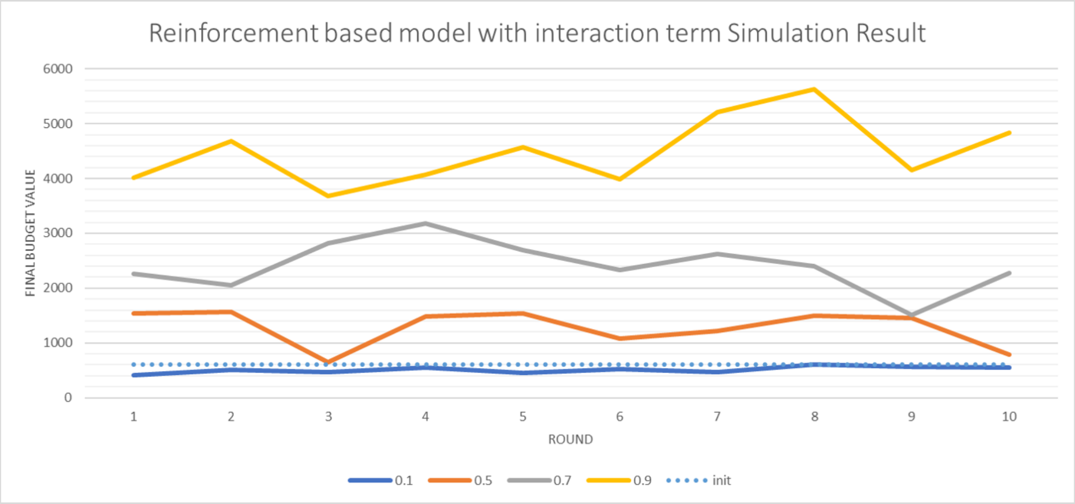


Figure 19: Reinforcement based model with interaction term simulation result

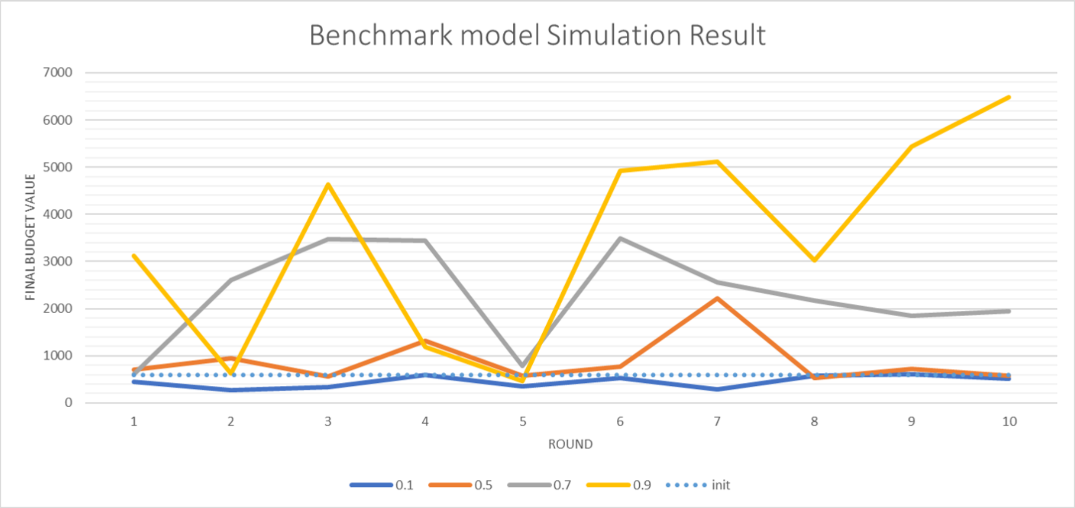


Figure 20: Benchmark model simulation result

Reinforcement based model simulation result, Reinforcement based model with interaction term simulation result and Benchmark model simulation result are illustrated in Figure 18, 19 and 20 respectively. The line's color indicates the level of variance ratio. The blue, orange, gray and yellow lines refer to 0.1, 0.5, 0.7 and 0.9 variance ratios, respectively. This graph shows the final budget for each dataset in each variance ratio (there are 10 datasets for each variance ratio).

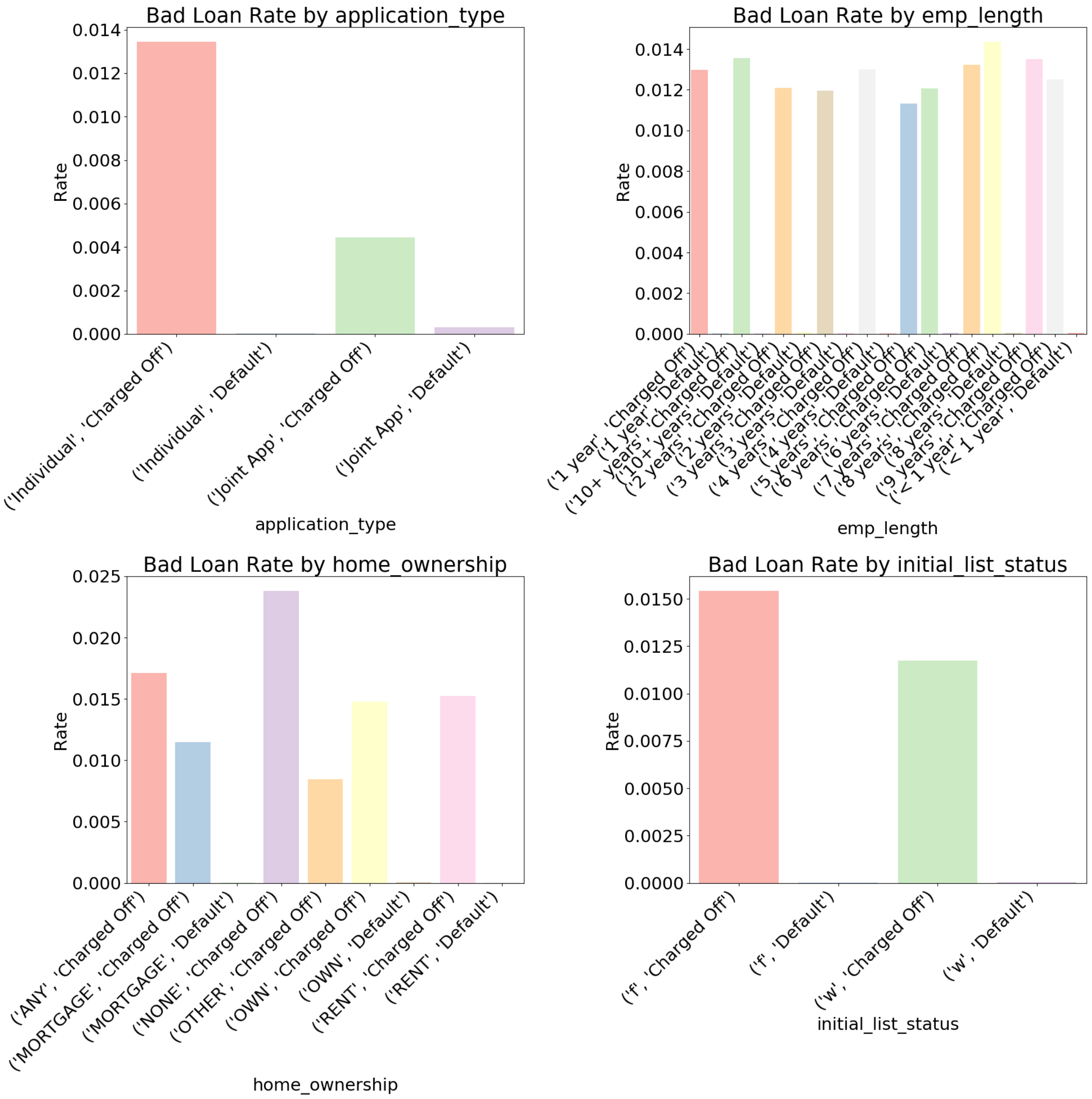


Figure 21: Bad loan rates by features

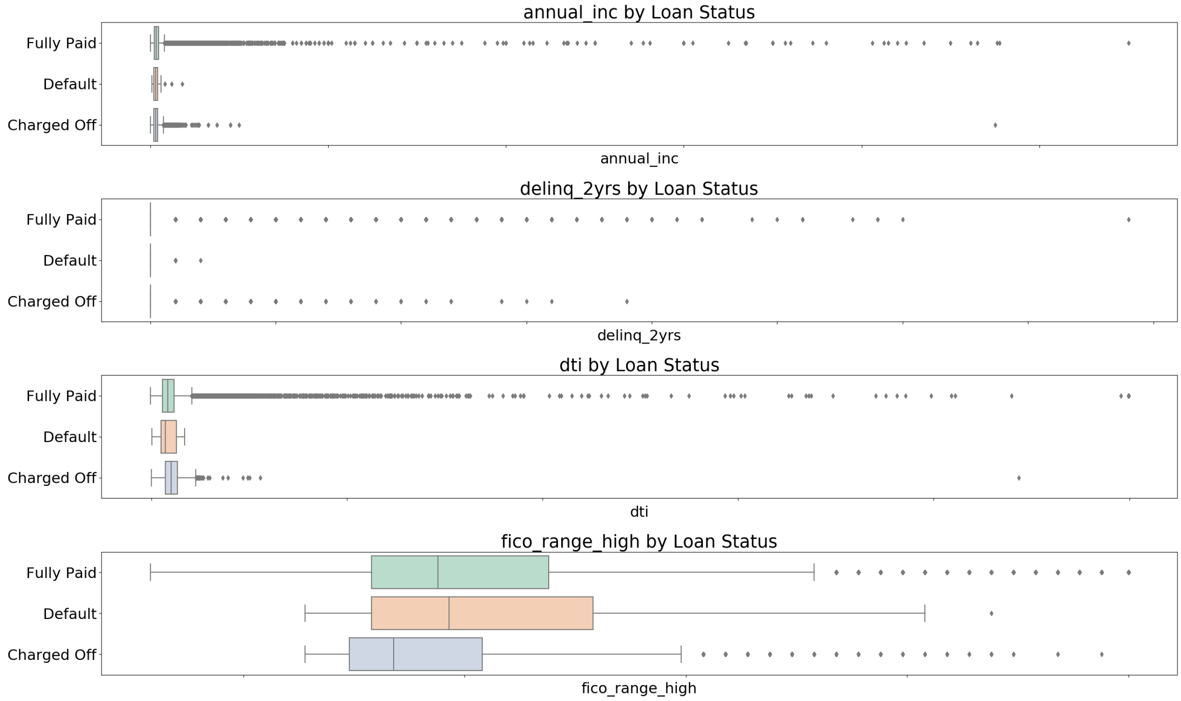


Figure 22: Loan's behavior by loan types

Figure 21 and 22 show the effect of some features on the loan's outcomes. To make sure that the selected parameters can distinguish between good loans and bad loans, we divide the value of these parameters into four bins and plot against the loan rates. We can see from the figures that the value of some parameters affects the loan rates. For example, the account that has high number of open credit line tends to default.

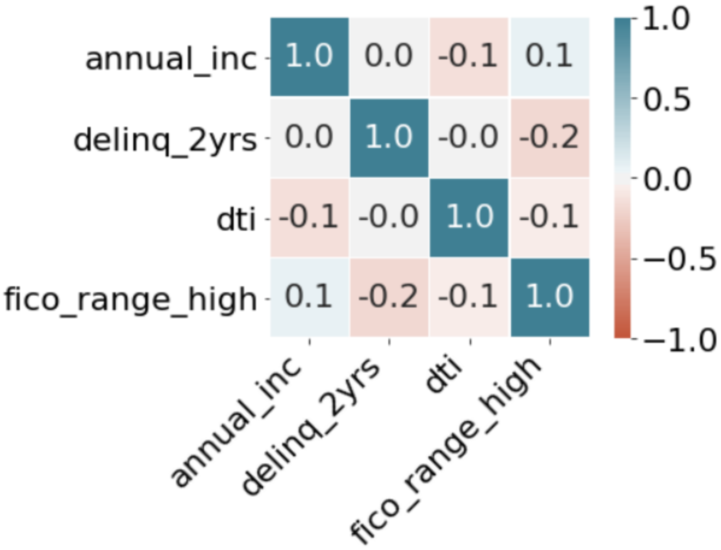


Figure 23: Features correlation heatmap

Highly correlated features can lead to multicolinearity and make the learning algorithm slower, therefore, we plot the heatmap to find the correlated features. The correlation of some features used in the model is shown in Figure 23.

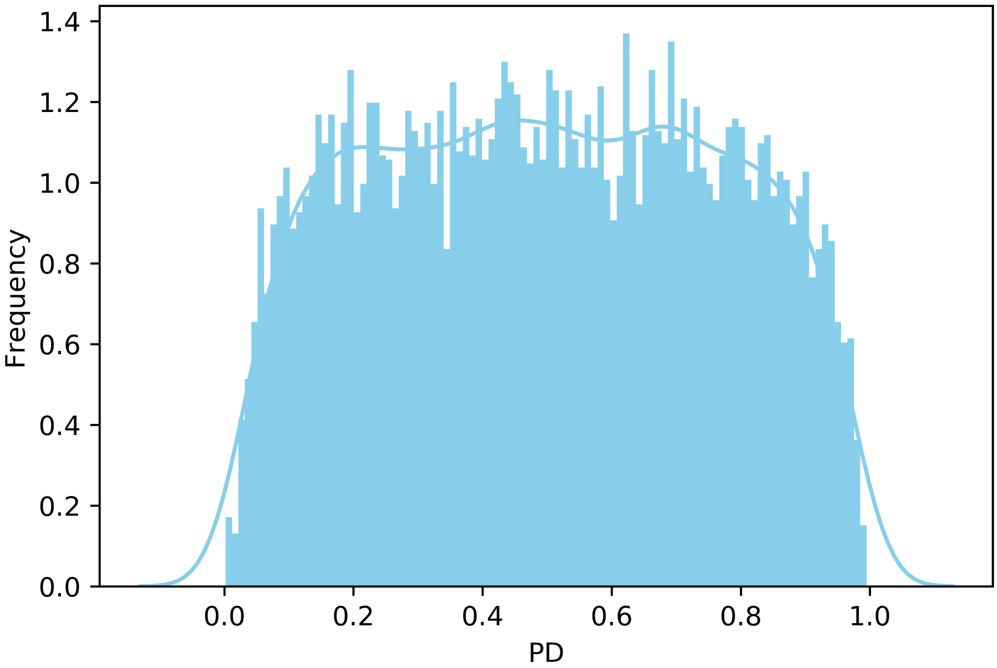


Figure 24: The distribution of simulated dataset's PD

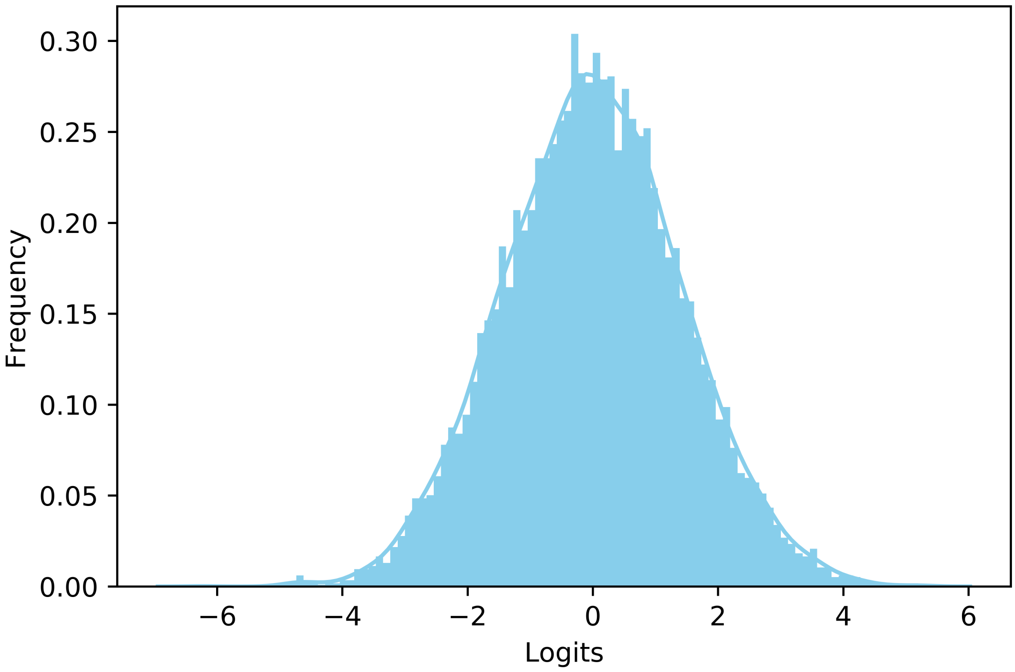


Figure 25: The distribution of simulated dataset's $\sum_{i=1}^{5} c_i x_{i,\tau}$

The distribution of the simulated dataset's PD is shown in Figure 24. We do not want the simulated PD to concentrate on some level, therefore, we make sure that the standard deviation of $\sum_{i=1}^{5} c_i x_{i,\tau}$ is equal to 1 by adjusting the constant coefficients. The distribution of simulated dataset's $\sum_{i=1}^{5} c_i x_{i,\tau}$ is shown in Figure 25.