Rational Delegation Learning Outsourcing Scheme Based on Game Theory

Kang Xiang[1,2], Youliang Tian[1,2(✉)], Sheng Gao[3], Changgen Peng[1,2], Weijie Tan[1]
1. State Key Laboratory of Public Big Data, College of Computer Science and Technology
Guizhou University, Guiyang 550025, P. R. China
2. Institute of Cryptography and Data Security, Guizhou University, Guiyang 550025, P. R. China
3. School of Information, Central University of Finance and Economics, Beijing 100081, P. R. China
Corresponding author E-mail: youliangtian@163.com

***Abstract*—At present, many enterprises or individuals with a lot of data are limited by their own computing power, so can't mine useful information from the data. To address this problem, this paper proposes a rational delegation machine learning pattern. In the proposed pattern, we firstly construct a reliable game model, and then derive a formal model of rational delegation machine learning according to delegation computation thought. Finally, we design an outsourcing scheme for decision tree model. We analyze the security and performance of the proposed scheme, the analysis results show that the proposed scheme can not disclose any useful information. Last but not least, the experimental result demonstrates that the proposed scheme can obtain a decision tree model with high accuracy.**

***Index Terms*—rational delegation learning; game theory; decision tree; machine learning;**

I. Introduction

Due to the complexity of machine learning technology [1] and the high demand for data sets, many enterprises or individuals with big data are limited by their own computing power and can't mine useful information from the data, so they can only rely on the service provider which has the calculate ability to mine data. However, in the absence of feasible schemes and technical support, it is unrealistic for an enterprise to directly entrust the data set to the service provider for processing. Therefore, it is of great theoretical significance and practical application value to design a new outsourcing service pattern based on the traditional delegation computation [2] and machine learning. We propose a delegation learning pattern for machine learning outsourcing training, as shown in the Fig. 1.

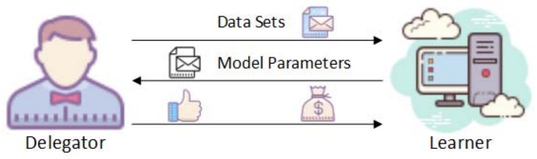


Fig. 1: The delegation learning pattern

Because there is a lot of sensitive information in the data send by the delegator to the service party, some malicious attackers may target data to get the privacy of delegator. Consequently, the design of the delegation learning outsourcing scheme needs to consider the following factors:

a. The service party cannot access the real data. In other words, the service party should use the encrypted data to train the machine learning model, and no sensitive information can be obtained from the mined results.

b. The method for removing sensitive information must be as simple as possible. And the encrypted data must meet the requirements of model training.

c. During the model training process, the service party should be prevented to forge model parameters to deceive the delegator. And the final model obtained by the delegator must be correct and available. This is the basic purpose and meaning of delegation learning.

In brief, even if the data set sent by the delegator is encrypted, the service party must ensure that the model trained on the encrypted data set is correct and feasible. In other words, if the design of the delegation learning outsourcing scheme meets the requirements of a, b and c, the client will be able to outsource the task of machine learning model training to the service providers.

A. Related Works

The research related to this work mainly includes two aspects: delegation computation and machine learning.

In terms of research on delegation computation, it has become current hotspot that use game theory to study the rational delegation computation and the relationship between participants. Originally, Azar et al. [3] proposed a rational proof system according to the appropriate scoring rules, in which the participants are neither honest nor malicious, but rational. Subsequently, Azar et al. [4] construct anultra-efficient rational proof system by using the idea of Utility Gaps. In [5,6,7], the researchers also studied from a rational perspective. In addition, other works [8,9] studied the rational secret sharing and rational delegation computing technology based on game theory. Inspired by [10], we also establish incentive mechanism in the proposed scheme.

The related research of privacy protection technology in machine learning can be divided into data perturbation-based methods and secure multi-party computing-based methods. In terms of data perturbation, Agrawal et al. [11] proposed a privacy protection method which adds random noise to implement decision tree mining. However, this method of randomly adding noise is too simple. Kargupta et al. [12, 13] based on the random matrix theory, proposed a real data estimating method from perturbed data. Subsequently, Bu et al. [14] presented a function-based perturbation method, which uses inverse function transformation to restore the decision tree on the perturbed data to the decision tree on the real data. Based on the reference [14], This paper replaces the real data, and further improves the security of the data without affecting the data mining. In terms of secure multi-party computing, Mohassel et al. [15] proposed a new and efficient confidential machine learning protocol, in which the researchers assume that the two servers are not collusive, but in reality, the service provider is rational. Compared with [16,17], our delegation learning scheme contains only one server, so the scheme does not need to aggregate the data sets first and then distribute them, which not only avoids collusion between servers, but also reduces risk of privacy disclosure and extra cost caused by communication between servers.

B. Our Contributions

Aim to the above factors, the main contributions of this paper are summarize as follows:

1. *We introduce rational participants into the delegation learning pattern. In addition, based on the utility thought and the game theory, we construct the game model and incentive function of rational delegation learning outsourcing scheme. The scheme achieves the utility balance of both parties by inspiring and restricting the participants, and can obtain a high accuracy model finally.*
2. *We derive the formal model of rational delegation learning based on the traditional delegation computation, and design a rational delegation learning outsourcing scheme for the decision tree model. This scheme not only ensures the security of user data but also has the result concealment characteristic.*
3. *According to the way of setting dummy variables, we design a preprocessing method of output integer based on function perturbation (OIFP) for text data and discrete data. In this method, we replace the real data and adopt the piecewise function-based perturbation (FP) method to encrypt the data set.*

This paper is organized as follows: Section II introduces the preliminary knowledge required in this article. Section III analyzes the rational delegation learning game model. Section IV proposes the rational delegation learning formal model and scheme. Section V performs security analysis and performance evaluation. Section VI concludes this paper.

II. PRELIMINARIES

A. Monochromatic Value and Monochromatic Piece

Suppose there are n instances in data set D, and each instance has d attributes ($B = \{B_1, B_2, \cdots, B_d\}$) and classification label Y. We express the value set of attribute B_i as $b_i(i = 1, 2, \cdots, d)$, and put all instances in ascending or descending order according to the value in b_i. If the label of all instances with value v on attribute B_i are equal, then v is a monochromatic value [14]. Namely, $\forall c_1, c_2 \in D$, if $c_1.B_i = c_2.B_i = v$ and $c_1.Y = c_2.Y$, then v is a monochromatic value, where c is an instance. For example, in the following Table I, there are 8 instances in total. We put all instances in ascending order with the value of age attribute. All values except 25 in the table are monochromatic values. If the age attribute values of all instances in the same data piece are monochromatic and the label values are equal, then this data block is called a monochromatic piece relative to age attribute. In Table I, p_1 and p_3 are monochromatic pieces.

B. Piecewise Function Perturbation

According to Section II.A, we randomly insert (w-1) breakpoints to divide the data that has been sorted according to the value of the attribute B_i into w pieces, namely $nb_i = \delta_1(nb_i) \cup \delta_2(nb_i) \cup \cdots \cup \delta_w(nb_i)$ and $\delta_r(nb_i) \cap \delta_k(nb_i) = \emptyset, r \neq k$, where nb_i represents the value sequence of n instances on the attribute B_i. Then w data pieces are respectively disturbed by w different functions, namely $nb'_i = \{f_1(\delta_1(nb_i)), f_2(\delta_2(nb_i)), \cdots, f_w(\delta_w(nb_i))\}$. Suppose that the data is arranged in ascending order. In order to satisfy the global monotonicity, if and only if, $1 < r < k < w, \forall v \in \delta_r(nb_i), \forall u \in \delta_k(nb_i)$, then $f_r(v) < f_k(u)$. According to [14], monochromatic pieces are suitable for any permutation function, but non-monochromatic pieces are only limited to monotonic functions. Here, we can define the function family suitable for monochromatic piece as F_{mon}, and the function family suitable for non-monochromatic piece as F_{nonmon}.

III. THE GAME-ANALYSIS OF RATIONAL DELEGATION LEARNING PATTERN

Assume that the participants include the delegator U and the learner L, and all of them are rational. The game model

TABLE I: The salary sample data

$Case:$	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
$Age(B_1):$	19	22	24	25	25	32	32	35
$Sex(B_2):$	w	m	m	w	w	m	w	m
$Education(B_3):$	$junior$	$bachelor$	$bachelor$	$master$	$bachelor$	$doctor$	$bachelor$	$master$
$Salary(Y):$	Low	Low	Low	$High$	Low	$High$	$High$	$High$
$Pieces:$		p_1			p_2		p_3	

of rational delegation learning is defined as: $G = \{PS, S, u\}$, participant set $PS = \{U, L\}$. The relevant definitions are as follows:

• $V(M)$ represents the worth of the model.

• $L(M)$ represents the cost of the learner training model,

• $U(M)$ represents the cost that the delegator needs to pay to the learner for outsourcing the training model.

• $T(D)$ represents the transmission cost that the delegator sends the data set and the learning task to the learner.

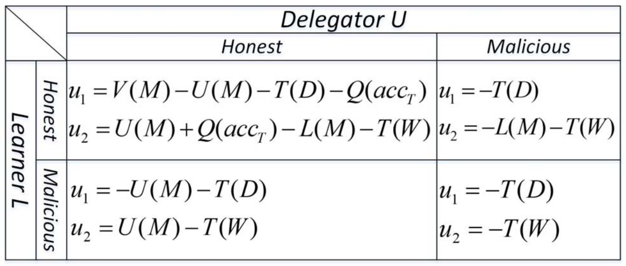
• $T(W)$ represents the transmission cost of sending the parameter set W of the model to the delegator after the learner trained the model. The above definition must meet $V(M) > U(M) > L(M)$ and $U(M) > L(M) + T(W)$, otherwise, the model is not necessary to delegate learning.

Since the participants are rational, they may behave maliciously for their own benefits. Therefore, both the delegator and the learner have the same set of behavioral strategies $S : \{honest, malicious\}$. In order to obtain a better model, the delegator can motivate the learner by set the incentive function Q and the minimum required accuracy acc_u of the actual application of the model. For simplicity, suppose the delegator's incentive budget is E and the incentive function is a quadratic function:

$$Q(acc_T) = \theta(acc_T - acc_u)^2 \times E, \qquad (1)$$

where θ and acc_T respectively represents the incentive coefficient and the actual accuracy of the model. And the incentive coefficient satisfies $\theta(1 - acc_u)^2 = 1$. If and only if $acc_T - acc_u > 0$, the delegator will pay the incentive amount of money $Q(acc_T)$. In short, when the higher the accuracy of the model trained by the learner, the more the learner will gain income. Of course, the reward, transmission cost and remuneration must meet $V(M) > Q(acc_T) + T(D) + U(M)$, otherwise, the model will lose the value of outsourcing training. The utility functions of the delegator and the learner are denoted by u_1 and u_2, as shown in the Table II.

TABLE II: The behavior utility matrix of participants



		Delegator U	
		Honest	Malicious
Learner L	Honest	$u_1 = V(M) - U(M) - T(D) - Q(acc_T)$ $u_2 = U(M) + Q(acc_T) - L(M) - T(W)$	$u_1 = -T(D)$ $u_2 = -L(M) - T(W)$
	Malicious	$u_1 = -U(M) - T(D)$ $u_2 = U(M) - T(W)$	$u_1 = -T(D)$ $u_2 = -T(W)$

According to the behavior utility matrix, it can be seen that the learner will prefer to choose a malicious strategy in order to maximize his own interests. This shows that the feasibility of delegation learning cannot be guaranteed. Therefore, we add a trusted third-party platform P to the delegation learning pattern. Suppose that the cost of introducing a third-party and the fee of third-party verification is P_R and P_V respectively. The P_R should be paid jointly by the delegator and the learner. If a party engages in malicious deception, the verification fee P_V will be paid by it. Before the start of the delegation learning process, the delegator and the learner respectively submits the deposit c and l on platform P. The deposits must meet $c \geq U(M) + P_V$ and $l > U(M) + T(D) + P_V + P_R/2$. Simultaneously, $U(M) > L(M) + T(W) + P_R/2$ and $V(M) > Q(acc_T) + T(D) + U(M) + P_R/2$. The final utility game tree of participants is shown in the Fig. 2.

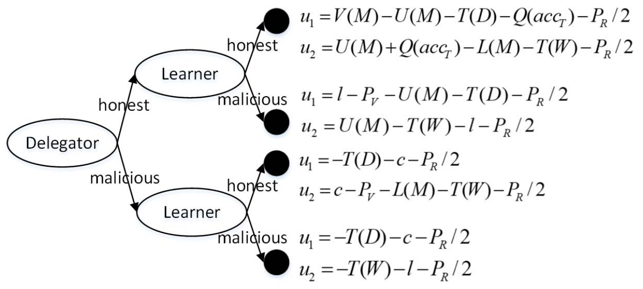

Delegator
honest
malicious
Learner
honest
malicious
Learner
honest
malicious
$u_1 = V(M) - U(M) - T(D) - Q(acc_T) - P_R/2$
$u_2 = U(M) + Q(acc_T) - L(M) - T(W) - P_R/2$
$u_1 = l - P_V - U(M) - T(D) - P_R/2$
$u_2 = U(M) - T(W) - l - P_R/2$
$u_1 = -T(D) - c - P_R/2$
$u_2 = c - P_V - L(M) - T(W) - P_R/2$
$u_1 = -T(D) - c - P_R/2$
$u_2 = -T(W) - l - P_R/2$

Fig. 2: The final utility game tree of participants

From the above analysis, when both the delegator and the learner choose the honest strategy, the interests of both parties can reach the optimal, and this strategy combination is Nash equilibrium of this game model.

IV. The Proposed Rational Delegation Learning Formal Model and Scheme

A. The Proposed Formal Model

Assume that the delegator has a discrete or text data set D, which contains n instances. Each instance has attribute set $B = \{B_1, B_2, ..., B_d\}$ and a label Y. The label value set and attribute B_i value set is represented by y and b_i respectively, i.e. $y = \{y_1, y_2, \cdots, y_s\}$, $b_i = \{b_{i,1}, b_{i,2}, \cdots, b_{i,s}\}$. Of course, the number of each attribute value is not necessarily equal. It is temporarily set as s for the convenience of description. In addition $d, n \in N^*$; $i = 1, 2, \cdots, d$; $j = 1, 2, \cdots, s$.

The formal model of the rational delegation learning (RDL) scheme for decision tree consists of the following three parts, namely $RDL = \{OIFP, Learning, Veri\}$.

1. The $OIFP$ method of data encryption based on piecewise function perturbation includes three steps:

(a). $One\text{-}hotGen(tab \xrightarrow{F_{(0,1)}} tab_b)$: First of all, the delegator needs to extract the attribute set, the attribute value set, label and label value set from data set D, and represents them with B, b_i, Y and y respectively. Then, the values in b_i and y are arranged in disorder to generate the basic information table tab, in which the data are replaced by $b_{i,j}$ and y_j. Finally, tab is transformed by mapping function $F_{(0,1)}$ to obtain the basic information table tab_b after boolean transformation.

(b). $IntegerGen(tab_b \xrightarrow{O_N} D_N)$: The data in tab_b is converted to integers and sorted in descending or ascending order. Then the actual value is replaced by the virtual value to obtain the complete virtual data set D_N, namely $D_N \leftarrow tab_N \leftarrow O_N(tab_b)$, where O_N is the virtual conversion function.

TABLE III: The basic information bijection table tab_N

$B:$	B_1						B_2		B_3				Y	
$Actual\ Value:$	24	35	25	22	19	32	w	m	ju	ma	ba	do	L	H
$Substitute\ Value:$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	$b_{1,4}$	$b_{1,5}$	$b_{1,6}$	$b_{2,1}$	$b_{2,2}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$	$b_{3,4}$	y_1	y_1
$One\text{-}hotGen:$	10^5	$0^1 10^4$	$0^2 10^3$	$0^3 10^2$	$0^4 10^1$	$0^5 1$	10^1	$0^1 1$	10^3	$0^1 10^2$	$0^2 10^1$	$0^3 1$	10^1	$0^1 1$
$IntegerGen:$	5^k	4^k	3^k	2^k	1	0	1	0	3^k	2^k	1	0	1	0

(c). $FP(D_N \xrightarrow{\vec{f}_{B_i}} D')$: Firstly, the delegator randomly inserts (w-1) breakpoints to divide the sequence nb_i of attribute B_i value into w pieces, and secretly sets the function families F_{mon} and F_{nonmon}. Next, the delegator randomly select the conversion functions from the function families F_{mon} and F_{nonmon}, and combine them into conversion function vector $\vec{f}_{B_i} = [f_1, f_2, ..., f_w]$, $\vec{f}_{B_i} \in \{F_{mon}, F_{nonmon}\}$. Finally, the delegator can obtain the converted sequence nb_i' of attribute B_i value, namely $nb_i' = \vec{f}_{B_i}(nb_i)$. After the d-round conversion, the disturbed data set D' can be obtained. Note that in the transformation, the delegator needs to secret preserve the breakpoint position and inverse function vector $\vec{f}_{B_i}^{-1} = [f_1^{-1}, f_2^{-1}, ..., f_w^{-1}]$ of each attribute sequence, so as to restore the real decision tree T from the decision tree T' based on D'.

2. $Learning((D', acc_u) \xrightarrow{C_i, A_i} (T', acc_l))$: According to the data set D' given by the delegator and the minimum accuracy requirement, the learner selects the appropriate criterion C_i and algorithm A_i for mining, and returns the final decision tree T' and accuracy acc_l.

3. $Veri(acc_T \overset{?}{\to} acc_l)$: The delegator verifies whether the accuracy of the decision tree T approaches to acc_l. Because there is a deviation when the test set verifies the accuracy of the model, the result $acc_T = acc_l$ is usually not obtained. Of course, the degree of "approaching" can be determined through negotiation between both parties. For example, when the condition $acc_u < acc_l - \lambda < acc_T < acc_l + \lambda$ is established, the delegator considers that the result returned by the learner is acceptable, otherwise the delegator can initiate a verification request to a third-party, where λ represents the average deviation of the model accuracy test.

B. The Proposed Scheme

i. *Initialization Phase*

1. Data processing

The delegator disturbs the data set according to the $OIFP$ method. Regarding the setting of functions $F_{(0,1)}$ and O_N, as long as the function satisfies the data conversion relationship, the delegator can set it arbitrarily. For the convenience of explanation, let the data in Table I as an example and design the following conversion function:

$$\begin{aligned}&F_{(0,1)}(b_{i,j}) = 0^{(j-1)}10^{(|b_i|-j)},\\&F_{(0,1)}(y_i) = 0^{(i-1)}10^{(|y|-i)},\\&O_N(b_{i,j}) = (|b_i| - j)^k,\ O_N(y_i) = |y| - i,\end{aligned} \tag{2}$$

here, the symbol $(*)$ represents that the quantity of 0 is $*$, the $|*|$ expresses the quantity of elements in the set $*$. The k expresses the index, $k \in N^*$, $k > 1$. The value of k is determined by the delegator. After the function conversion, the basic information bijection table tab_N can be obtained, as shown in the Table III. It can be seen that all the data in the original data, including text data, are converted into non-negative integers, which not only avoids the leakage of real data, but also makes it easier to perform the next step function perturbation operation.

TABLE IV: The monochrome pieces of attribute B_1

$Case:$	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
$B_1:$	5^k	4^k	3^k	3^k	2^k	1	0	0
$Y:$	1	0	0	1	1	1	0	0
$Pieces:$	p_1				p_2		p_3	

According to the descriptions in Section II.A and Section II.B, we might as well arrange the attribute values of B_1 in descending order, and randomly select two breakpoints to divide the data into three pieces. As shown in the Table IV. It is can be observed that the p_2 and p_3 data pieces are monochromatic piece, and p_1 is a non-monochromatic piece. Firstly, we need to randomly select the conversion function vector $\vec{f}_{B_1} = [f_1, f_2, f_3]$, $f_2, f_3 \in F_{mon}$, $f_1 \in F_{nonmon}$. Then, three data pieces are respectively perturbed by three different functions, i.e. $nb_1' = \{f_1(\delta_1(nb_1)), f_2(\delta_2(nb_1)), f_3(\delta_3(nb_1))\}$. It is worth noting that the global monotonic invariant characteristic must be satisfied when performing piecewise function perturbation, namely $\forall u \in \{f_1(5^k), f_1(4^k), f_1(3^k)\}$, $\forall v \in \{f_2(2^k), f_2(1)\}$, must be $u > v > f_3(0)$. The rest may be deduced by analogy, the final perturbed data set D' can be obtained after the function vectors $\vec{f}_{B_2}$ and $\vec{f}_{B_3}$ respectively convert the values of B_2 and B_3.

2. Creating test set, validation set and training set

Firstly, the delegator sends the data set D' to the third-party P. Then, the third-party randomly selects 20% and 10% of the data from the D' as the test set TeS' and the validation set VS', and all the remaining data as the training set TrS', namely $D' = TeS' + VS' + TrS'$. It is worth noting that the reason why the training set and the test set are extracted by the third-party from the data set D' is to ensure that the data when building the model and testing the model have the same distribution. We assume that there are

m instances in the training set and validation set, namely $TrS' + VS' = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_m, Y_m)\}$.
3. Setting the time node and submitting deposit
The delegator publishes the minimum requirement for the accuracy acc_u of the model, and negotiates with the learner to decide that the learner must return to the decision tree T' and its promised accuracy acc_l within a limited time t. If the delegator verifies that the result is correct, the $U(M)$ and $Q(acc_T)$ fee must be paid to the learner within a limited time t'. After the delegator and the learner reach the delegation agreement, they respectively submit the deposit c and l to the trusted third-party.

ii. *Learning Phase*

Firstly, the learner selects the algorithm A_i from the algorithm space A and learns the data features in the TrS' to obtain the model, namely $TrS' \xrightarrow{A_i} T''$. Secondly, iteratively optimize the model using different criterion C_i to make the model optimal, namely $T'' \xrightarrow{C_i, VS'} (T', acc_l)$. Finally, the learner returns the optimal model T' and acc_l to the delegator.

iii. *Verification and Payment Phase*

This phase is discussed into two situations:
(1) The learner does not return the result within the agreed time t.
(2) The learner returns the result within the agreed time t.

In the first case, it shows that the learner does not honestly implement. At this time, the delegator can contact P to recover his deposit c and confiscate the deposit l of the learner.

In the second case, the delegator decrypts T' after receiving T' and acc_l by using its own secretly saved inverse function vector $\vec{f}_{B_i}^{-1}$, bijection table tab_N and breakpoint location of each attribute to obtain the real decision tree model T. Then, the delegator verifies accuracy of T. The specific steps are as follows:
1. The delegator uses $\vec{f}_{B_i}^{-1}$, the breakpoint locations of each attribute and the bijection table tab_N to restore the test set TeS;
2. The delegator tests T by the TeS to get the accuracy acc of the model.

$$acc(T, TeS) = \frac{1}{n-m} \sum_{i=1}^{n-m} \mathrm{I}(T(X_i) = Y_i), \tag{3}$$

where $\mathrm{I}(*)$ represents the indication function and takes a value of 1 or 0 when $\star$ is true or false.
3. Of course, the delegator can evenly divide the TeS into h parts ($TeS = \{TeS_1, TeS_2, \cdots, TeS_h\}$) to test the model and use the final average value $\overline{acc}$ as the accuracy acc_T of the model. The value of h can generally define a minimum value. For example, $h \geq 5$,

$$\begin{aligned} acc_T = \overline{acc} = \frac{1}{h} \sum_{j=1}^{h} acc_j(T, TeS_j), \\ \lambda = \frac{1}{h} \sum_{j=1}^{h} |acc_j - \overline{acc}|. \end{aligned} \tag{4}$$

4. The delegator announces the result of comparing acc_T with acc_l and discusses the following situations:
a) When $acc_u \leq acc_l - \lambda \leq acc_T$, the delegator must pay remuneration and reward within the time of t', otherwise the learner can make a request to the third-party P to confiscate the delegator's deposit.
b) When $acc_u \leq acc_T < acc_l - \lambda$ or $acc_T < acc_u$, if the learner questions the test result announced by the delegator, it can initiate a verification request to a third-party P.
The third-party verification process is as following:
1. The third-party obtains T' and acc_l from the learner.
2. The third-party uses the same method as the delegator to test the model with the encrypted TeS'.
The verification results include the following three situations:
a) When the result shows $acc_u \leq acc_T < acc_l - \lambda$, it means that the delegator does not deceive the learner during the test process, so the verification fee of the third-party should be paid by the learner. But the delegator must pay the remuneration and reward to the learner within a limited time according to the actual accuracy acc_T of the model, otherwise the third-party helps the learner to confiscate the delegator's deposit.
b) However, when the result shows $acc_u \leq acc_l - \lambda \leq acc_T$, it means that the delegator deceives the learner during the test process, so the verification fee of the third-party should be paid by the delegator. Not only that, the learner can request the third-party to confiscate the delegator's deposit.
c) When the result shows $acc_T < acc_u$, it means that the learner deceives the delegator during the learning process, so the verification fee of the third-party should be paid by the learner. At the same time, the delegator can request the third-party to help him confiscate the deposit of the learner.

V. Security Analysis and Performance Evaluation

A. Security Analysis

Theorem 1: If the value width of the data piece $\delta_r(nb_i), r \in [1, w]$ of each attribute is large enough, the data set D' of the delegator is hidden. That is to say, the rational delegation learning scheme is safe.

Proof. In [14], the author has demonstrated data safety of the monochromatic pieces and decision tree T'. However, concerning the defense sorting attack of non-monochrome pieces, we make the width of the data piece controllable, which means that the leakage risk of information is further reduced. For a better explanation, we first give some simple definitions:
- $g_r : \delta'_r(nb_i) \to \delta_r(nb_i)$ represent the attacker's cracking function.
- $[\min^k, \max^k] \in \delta_r(nb_i)$, the $\max^k - \min^k$ represents the value width of the r-th piece of the attribute B_i.
- If $|g_r(v') - f_r^{-1}(v')| \leq \rho$, then we think the attacker cracked a numerical point $R_\rho = [v - \rho, v + \rho]$, where $v = f_r^{-1}(v')$, ρ represents the distance between the attacker's guess value and the real value.

As a matter of fact, when $\delta'_r(nb_i)$ contains some discontinuous values, the attacker can only crack the value $v' \in \delta'_r(nb_i)$ to a limited width $R_g = [v_1, v_2] \in \delta_r(nb_i)$. Assume that there are respectively m values and n values in front of and behind the value v' in piece p'_r, the width that the attacker can attack is $R_g = [\min^k + m, \max^k - n]$. In the sorting attacks, the probability that the value v' is cracked can be defined as

$$P_{v'}(|g_r(v') - f_r^{-1}(v')| \leq \rho) = \frac{|R_g \cap R_\rho|}{|R_g|}. \quad (5)$$

According to the above analysis, it is clear that when k is larger, the range of R_g is wider and the probability $P_{v'}$ is smaller. If the value of k is large enough, the probability $P_{v'}$ is negligible, so the data set D_N of the delegator is safe.

B. Performance Evaluation

We set up two participants (the delegator U and the learner L) and a trusted third-party P according to the rational delegation learning scheme. The performance test of proposed scheme are carried out using the disease prediction data set with a size of 0.8MB. Here, assume that the minimum accuracy requirement by the delegator is 90%, the case of setting the incentive function (IF) and without the incentive function (NIF) is compared through experimental simulation, as shown in the Fig. 3. It is can be observed that when the incentive function is not set, the learner improves the accuracy of the model to acc_u and no longer trains, and directly returns the model T' to the delegator. However, when the incentive function is set, the learner tries his best to improve the accuracy of the model in order to obtain more benefits. Therefore, it is shown that the proposed scheme can finally obtain a reliable model.

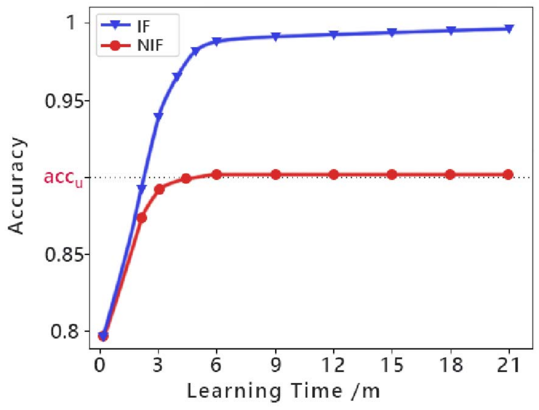


Fig. 3: The model accuracy comparison figure

VI. Conclusions

This paper proposes a rational delegation learning pattern and designs an outsourcing scheme for decision tree model, which reduces the leakage risk of delegator's data in model outsourcing training. We establish the incentive function in learning process, which makes the rational learner to increase the model accuracy. Therefore, the feasibility of the rational delegation learning scheme is improved. Finally, we evaluate the proposed scheme, the evaluation results show that the delegator can get a high-precision model.

Acknowledgment

This work was supported by National Natural Science Foundation of China (Grant Nos. 61662009 and 61772008), Science and Technology Major Support Program of Guizhou Province (Grant No. 20183001), Ministry of Education-China Mobile Research Fund Project (Grant No. MCM20170401), Key Program of the National Natural Science Union Foundation of China (Grant No. U1836205), Science and Technology Program of Guizhou Province (Grant No. [2019]1098), Project of High-level Innovative Talents of Guizhou Province (Grant No. [2020]6008).

References

[1] M. Somvanshi, P. Chavan, S. Tambade and S. V. Shinde, "A review of machine learning techniques using decision tree and support vector machine," *In: International Conference on Computing Communication Control and Automation*, pp. 1-7, 2016.

[2] S. Goldwasser, Y. T. Kalai and G. N. Rothblum, "Delegating computation: interactive proofs for muggles," *Journal of The ACM (JACM)*, vol. 62, no. 4, pp. 27-45, 2015.

[3] P. D. Azar, S. Micali, "Rational proofs," *In: Proceedings of The Forty-fourth Annual ACM Symposium on Theory of Computing*, pp. 1017-1028, 2012.

[4] P. D. Azar, S. Micali, "Super-efficient rational proofs," *In: Proceedings of The Fourteenth ACM Conference on Electronic Commerce*, pp. 29-30, 2013.

[5] Y. L. Tian, J. Guo, Y. L. Wu and H. Lin, "Towards attack and defense views of rational delegation of computation," *IEEE Access*, vol. 7, pp. 44037-44049, 2019.

[6] S. Guo, A. Rosen, H. Pavel and V. Margarita, "Rational arguments: single round delegation with sublinear verification," *In: Proceedings of The 5th Conference on Innovations in Theoretical Computer Science*, pp. 523-540, 2014.

[7] Y. L. Tian, C. G. Peng, D. D. Lin, J. F. Ma, Q. Jiang and W. J. Ji, "Bayesian mechanism for rational secret sharing scheme," *Science China Information Sciences*, vol. 58, no. 5, pp. 1-13, 2015.

[8] Y. L. Tian, J. F. Ma, C. G. Peng and W. J. Ji, "Game-heoretic analysis for the secret sharing scheme," *Acta Electronica Sinica*, vol. 39, no. 12, pp. 2790-2795, 2011.

[9] Q. X. Li, Y. L. Tian and Z. Wang, "Rational delegation computation protocol based on full homomorphic encryption," *Acta Electronica Sinica*, vol. 47, no. 2, pp. 470-474, 2019.

[10] C. Dong, Y. Wang, A. Aldweesh, P. McCoeey and A. P. A. V. Moorsel, "Betrayal, distrust, and rationality: smart counter-collusion contracts for verifiable cloud computing," *In: Proceedings of The 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 211-227, 2017.

[11] R. Agrawal, R. Srikant, "Privacy-preserving data mining," *In: Proceedings of The ACM SIGMOD International Conference on Management of Data*, pp. 439-450, 2000.

[12] H. Kargupta, S. Datta, Q. Wang and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," *In: Third IEEE International Conference on Data Mining*, pp. 99-106, 2003.

[13] H. Kargupta, S. Datta, Q. Wang and K. Sivakumar, "Random-data perturbation techniques and privacy-preserving data mining," *Knowledge and Information Systems*, vol. 7, no. 4, pp. 387-414, 2005.

[14] S. Bu, L. V. S. Lakshmanan, R. T. Ng and G. Ramesh, "Preservation of patterns and input-output privacy," *In: IEEE 23rd International Conference on Data Engineering*, pp. 696-705, 2007.

[15] P. Mohassel, Y. Zhang, "SecureML: a system for scalable privacy-preserving machine learning," *In: IEEE Symposium on Security and Privacy (SP)*, pp. 19-38, 2017.

[16] P. Mohassel, P. Rindal, "ABY 3: a mixed protocol framework for machine learning," *In: Proceedings of The ACM SIGSAC Conference on Computer and Communications Security*, pp. 35-52, 2018.

[17] X. Ma, F. Zhang, X. Chen and J. Shen, "Privacy preserving multi-party computation delegation for deep learning in cloud computing," *Information Sciences*, vol. 459, pp. 103-116, 2018.