Privacy Preserving Fuzzy Modeling for Secure Multiparty Computation

Hirofumi Miyajima[†1], Noritaka Shigei[†2], Hiromi Miyajima[†3], Yohtaro Miyanishi[†4] Shinji Kitagami[†5] and Norio Shiratori[†6]

***Abstract*—Many studies on privacy preserving of machine learning and data mining have been done in various methods by use of randomization techniques, cryptographic algorithms, anonymization methods, etc. Data encryption is one of typical approaches. However, its system requires both encryption and decryption for requests of client or user, so its complexity of computation is very high. Therefore, studies on secure computation using shared data are made to avoid secure risks being abused or leaked and to reduce computing cost. The secure multiparty computation (SMC) is one of these methods. So far, some studies have been done with SMC, but complex calculation processing such as machine learning has never proposed yet. In the previous paper, we proposed BP learning method for SMC on cloud computing system. In this paper, we propose learning method(Fuzzy modeling) of fuzzy inference system for SMC and prove the validity of it. Further, the performance of the proposed method is shown in numerical simulations.**

***Index Terms*—cloud computing, secure multiparty computation, fuzzy modeling, control problem.**

I. Introduction

PRIVACY preserving machine learning and data mining can be achieved in various methods by use of randomization techniques, cryptographic algorithms, anonymization methods, etc [1]–[6]. However, data encryption system requires both encryption and decryption for requests of client or user, so its complexity is very high. The problem is the trade-off between the security and the complexity. As one of these studies, secure multiparty computation (SMC) has been introduced [7]–[9]. The purpose of SMC is to allow servers(parties) to carry out distributed computing tasks in secure way. As one of them, SMC systems sharing data itself to each party attract attention, and some studies with them have been done. A simple method to share data was proposed and they were applied to some problems [10], [11]. In the previous paper, we proposed BP learning for neural networks and clustering methods for SMC [12], [13]. In this paper, we propose learning method(fuzzy modeling) of fuzzy inference systems for SMC and show the effectiveness of them in numerical simulations. The difference between BP learning and fuzzy modeling is as follows [14] : In BP learning input and output relation for learning data is represented as weights of neurons of neural network, but it is difficult to know the interpretability of input and output learning data. In fuzzy modeling, input and output relation for learning data is represented as fuzzy rules for fuzzy inference system and it is easy to understand the interpretability of input and output learning data. In this paper, we will show this relation using a control problem. In Section 2, we describe the idea for sharing data securely. Further, we also introduce the conventional learning algorithm for fuzzy inference systems. In Section 3, we describe our proposed learning algorithm for SMC. In Section 4, some simulation results are presented to demonstrate the effectiveness of the proposed method.

Affiliation: Graduate School of Science and Engineering, Kagoshima University, 1-21-40 Korimoto, Kagoshima 890-0065, Japan
corresponding auther to provide email: miya@eee.kagoshima-u-ac.jp
†1 email: k3768085@kadai.jp
†2 email: shigei@eee.kagoshima-u-ac.jp
†3 email: miya@eee.kagoshima-u-ac.jp
Affiliation: Information Systems Engineering and Management, Tokyo, Japan
†4 email: miyanisi@jade.dti.ne.jp
Affiliation: Waseda University Graduate School of Global Information and Telecommunication Studies(GITS), Tokyo, Japan
†5 email: kitagami.shinji@meltec.co.jp
Affiliation: Waseda University GITS, Tokyo, Japan
†6 email: norio@shiratori.riec.tohoku.ac.jp

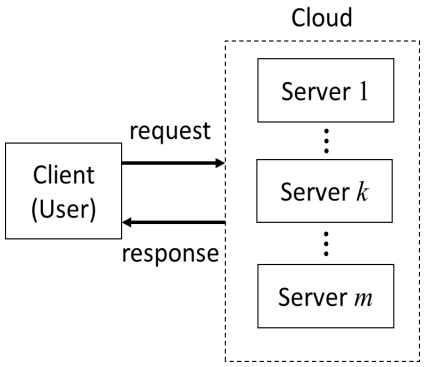


Fig. 1. A configuration of cloud computing system.

II. Preliminary

A. System configuration of cloud system and related works

Fig.1 shows a system for SMC of cloud computing. The system is composed of a client and cloud with m servers(parties) [12]. The client sends data to each server and each server memorizes them. If the client requires data processing, each server performs one's computation and sends each result to client. The client computes the final result using them. If the result is not obtained by one processing, data processing between client and servers are iterated until the final result is obtained. The problem is how data are shared and the computation for each server is carried out.

Let us consider about conventional works related with them easily. Three partitioned representation of data such as horizontally, vertically and any partitioned methods for SMC are known [9], [12]. Let us explain about them using an example of Table 1. In Table 1, a and b are original data (marks) and ID is student identifier. The purpose of computation is to get the average of them.

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

First, let us explain about horizontally partitioned method(HPM) using Table 1. All the dataset are divided into two servers, Server 1 and 2 as follows:

Server 1: dataset for ID=1, 2

Server 2: dataset for ID=3, 4.

In this case, two averages with subsets A and B for Server 1 are computed as $(90+60)/2$ and $(55+82)/2$, respectively. Likewise, two averages with subsets A and B for Server 2 are $(70+40)/2$ and $(30+70)/2$, respectively. As a result, two averages for subsets A and B are 65.0 and 59.25, respectively. These are obtained as the sums of average of $a^{(1)}$ and $a^{(2)}$, and of $b^{(1)}$ and $b^{(2)}$,where $a = a^{(1)} + a^{(2)}$ and $b = b^{(1)} + b^{(2)}$. Each server cannot know half of the dataset, so privacy preserving holds.

Likewise, vertically partitioned method(VPM) is introduced. In this care, dataset for subject A and B are assigned to Server 1 and 2, respectively.

At third, let us consider about any partitioned method for SMC. All the dataset are divided into two parties, mixed horizontally and vertically partitioned data. These methods need a large number of learning data to keep security, but the proposed method using secure shared data seems to keep them by small number of data.

B. The representation of secure shared data

Let us explain data representation for the proposed method using Fig.2 [10], [11]. Let a and b be two positive integers. First, two integers a and b are shared into m real numbers. Let $a = a^{(1)} + \cdots + a^{(m)}$ and $b = b^{(1)} + \cdots + b^{(m)}$ as the addition form and $a = A^{(1)} \cdots A^{(m)}$ and $b = B^{(1)} \cdots B^{(m)}$ as the multiplication form. Then the following results hold:

1)$a + b = (a^{(1)} + b^{(1)}) + \cdots + (a^{(m)} + b^{(m)})$

2)$a - b = (a^{(1)} - b^{(1)}) + \cdots + (a^{(m)} - b^{(m)})$

3)$ab = (A^{(1)}B^{(1)}) \cdots (A^{(m)}B^{(m)})$

4)$a/b = (A^{(1)}/B^{(1)}) \cdots (A^{(m)}/B^{(m)})$

That is, four basic operations of arithmetic (addition, subtraction, multiplication, and division) hold as integration of the result computed independently by each server [11], [12]. In this paper, $a^{(k)}$ and $A^{(k)}$ for $1 \leq k < m-1$ are selected randomly in $[0, 1]$ and $[-1, 1]$, respectively. Further, $a^{(m)}$ and $A^{(m)}$ are selected as $a - \sum_{k=1}^{m-1} a^{(k)}$ and $a/\Pi_{k=1}^{m-1} A^{(k)}$, respectively.

Let us show an example of shared data using Table I as follows [11]:

$a = a^{(1)} + a^{(2)} : a^{(1)} = a(r_1/10)$ and $a^{(2)} = a(1 - r_1/10)$,

$b = b^{(1)} + b^{(2)} : b^{(1)} = b(r_1/10)$ and $b^{(2)} = b(1 - r_2/10)$,

$a = A^{(1)}A^{(2)} : A^{(1)} = \sqrt{a}(r_1/10)$ and $A^{(2)} = \sqrt{a}(10/r_1)$,

$b = B^{(1)}B^{(2)} : B^{(1)} = \sqrt{b}(r_2/10)$ and $B^{(2)} = \sqrt{b}(10/r_2)$

, where r_1 and r_2 are real random numbers for $-9 \leq r_1 \leq 9$ and $0.2 \leq r_2 \leq 9$, $r_1 \neq 1$, $r_2 \neq 1$, respectively. For example, $a^{(1)}$ and $a^{(2)}$ for ID=1 are computed as $a^{(1)} = 90 \times (2/10) = 18$ and $a^{(2)} = 90 \times (1 - 2/10) = 72$ and data $A^{(1)}$ and $A^{(2)}$ for ID=1 are computed as $A^{(1)} = \sqrt{90} \times (4/10) = 3.79$ and $A^{(2)} = \sqrt{90} \times (10/4) = 23.71$, respectively. Note that Server 1 has all the data in column-wise of $a^{(1)}$, $b^{(1)}$, $A^{(1)}$ and $B^{(1)}$ for each ID and Server 2 has all the data in column-wise of $a^{(2)}$, $b^{(2)}$, $A^{(2)}$ and $B^{(2)}$ for each ID as shown in Table.I.

Remark that each data for server is randomized and the method does not need to use encrypted data.

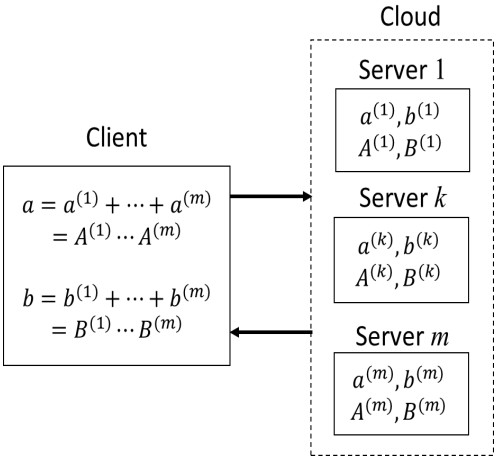


Fig. 2. The representation of secure shared data.

C. The conventional fuzzy inference system

The conventional fuzzy inference system is described [14]. Let $Z_j = \{1, \cdots, j\}$ for the positive integer j. Let R be the set of real numbers. Let $\boldsymbol{x} = (x_1, \cdots, x_N)$ and y^* be input and output data, respectively, where $x_j \in R$ for $j \in Z_N$ and $y^* \in R$. Then the rule of fuzzy inference model is expressed as

$$R_i \ : \ \text{if } x_1 \text{ is } M_{i1} \text{ and } \cdots \text{ and } x_N \text{ is } M_{iN} \text{ then } y \text{ is } c_i \qquad (1)$$

, where $i \in Z_n$ is a rule number, $j \in Z_N$ is a variable number, M_{ij} is a membership function of the antecedent part, and c_i is a real number of the consequent part.

A membership value of the antecedent part μ_j for input $\boldsymbol{x}$ is expressed as

$$\mu_i = \prod_{j=1}^{N} M_{ij}(x_j). \qquad (2)$$

If Gaussian membership function is used, then M_{ij} is expressed as follow

$$M_{ij} = \exp\left(-\frac{1}{2}\left(\frac{x_j - a_{ij}}{b_{ij}}\right)^2\right). \qquad (3)$$

, where a_{ij} and b_{ij} are the center and the width values of M_{ij}, respectively.

The output y^* of fuzzy inference is calculated by the following equation:

$$y = \frac{\sum_{i=1}^{n} \mu_i \cdot c_i}{\sum_{i=1}^{n} \mu_i}. \qquad (4)$$

D. Learning algorithm for the conventional model

In order to construct the effective model, the conventional learning method is introduced. The objective function E is defined to evaluate the inference error between the desirable output y^r and the inference output y^*. In this section, we describe the conventional learning algorithm. Let $\boldsymbol{D} = \{(x_1^p, \cdots, x_N^p, y_p^r) | p \in Z_P\}$ be the set of learning data. The objective of learning is to minimize the following mean square error(MSE):

$$E = \frac{1}{P}\sum_{p=1}^{P}(y_p^* - y_p^r)^2. \qquad (5)$$

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I
DATA ON SERVER 1 AND SERVER 2.

ID	subject A	subject B	Additional form					Multiplication form				
				a		b			A		B	
	a	b	r_1	$a^{(1)}$	$a^{(2)}$	$b^{(1)}$	$b^{(2)}$	r_2	$A^{(1)}$	$A^{(2)}$	$B^{(1)}$	$B^{(2)}$
1	90	55	2	18	72	11	44	4	3.79	23.71	2.97	18.54
2	60	82	-3	-18	78	-24.6	106.6	5	3.87	15.49	4.53	18.11
3	70	30	5	35	35	15	15	0.4	0.33	209.17	0.22	136.93
4	40	70	-6	-24	64	-42	112	3	1.90	21.08	2.51	27.89
average	65	59.25		2.75	62.25	-10.15	69.4					

, where y_p^* is the inference output for the p-th input $\boldsymbol{x}^p$. In order to minimize the objective function E, each parameter $\alpha \in \{a_{ij}, b_{ij}, c_i\}$ is updated based on the descent method as follows [14]:

$$\alpha(t+1) = \alpha(t) - K_\alpha \frac{\partial E}{\partial \alpha} \tag{6}$$

where t is iteration time and K_α is a constant. When Gaussian membership function for $i \in Z_n$ and $j \in Z_m$ are used, the following relation holds [14].

$$\frac{\partial E}{\partial a_{ij}} = \frac{\mu_j}{\sum_{i=1}^n \mu_i} \cdot (y - y^*) \cdot (c_i - y) \cdot \frac{x_j - a_{ij}}{b_{ij}^2} \tag{7}$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\mu_i}{\sum_{i=1}^n \mu_i} \cdot (y - y^*) \cdot (c_i - y) \cdot \frac{(x_j - a_{ij})^2}{b_{ij}^3} \tag{8}$$

$$\frac{\partial E}{\partial c_i} = \frac{\mu_i}{\sum_{i=1}^n \mu_i} \cdot (y - y^*) \tag{9}$$

Then, the conventional learning algorithm is shown as follows [14]:

Learning Algorithm A

Step A1 : The threshold θ of inference error and the maximum number of learning time T_{max} are given. The initial assignment of fuzzy rules is set to equally intervals. Let n be the number of rules and $n = d^m$ for an integer d. Let $t = 1$.

Step A2 : The parameters a_{ij}, b_{ij} and c_i are set to the initial values.

Step A3 : Let $p = 1$.

Step A4 : A data $(x_1^p, \cdots, x_m^p, y_p^r) \in \boldsymbol{D}$ is given.

Step A5 : From Eqs.(2) and (4), μ_i and y^* are computed.

Step A6 : Parameters a_{ij}, b_{ij} and c_i are updated by Eqs.(7), (8) and (9).

Step A7: If $p = P$ then go to Step A8 and if $p < P$ then go to Step A4 with $p \leftarrow p + 1$.

Step A8: Let $E(t)$ be inference error at step t calculated by Eq.(5). If $E(t) > \theta$ and $t < T_{max}$ then go to Step A3 with $t \leftarrow t+1$ else if $E(t) \leq \theta$ and $t \leq T_{max}$ then the algorithm terminates.

Step A9: If $t > T_{max}$ and $E(t) > \theta$ then go to Step A2 with $n = d^m$ as $d \leftarrow d + 1$ and $t = 1$.

III. SECURE MULTIPARTY COMPUTATION

Let us consider a system composed of client and m servers(See Fig.1). In learning on cloud system, learning data and parameters are shared to each server in additional or multiplication forms. Each server updates shared parameters and sends the computation result to the client. The client can get new parameters by additional or multiplying the results of m servers. The process is iterated until the error (difference) between the output of the system and the desired output becomes sufficiently small. The problem is how parameters on the user are updated using the set of learning data shared on each server. The shared representation of learning data $\{(\boldsymbol{x}^l, d(\boldsymbol{x}^l)) | l \in Z_P\}$ and parameters are given as follows:

$$\boldsymbol{x}^l = \left(x_1^l, \cdots, x_j^l, \cdots, x_N^l\right) \tag{10}$$

for $l \in Z_P$ and

$$x_j^l = \sum_{k=1}^m (x_j^l)^k \tag{11}$$

for $i \in Z_N$,

$$d(\boldsymbol{x}^l) = \sum_{k=1}^m (d(\boldsymbol{x}^l))^k \tag{12}$$

, where $d(\mathrm{x}^l)$ is the desirable output for the input x^l. Note that Eqs.(11) and (12) are in additional form for shared data.

In this case, Eqs.(7), (8) and (9) are renewed as follows:

$$\triangle a_{ij}^k = \frac{\mu_j}{\sum_{i=1}^n \mu_i} \cdot (y - y^*) \cdot (c_i - y) \cdot \frac{x_j - a_{ij}}{b_{ij}^2} \tag{13}$$

$$(a_{ij}^k)(t+1) = (a_{ij}^k)(t) + K \triangle a_{ij}^k(t) \tag{14}$$

$$\triangle b_{ij}^k = \frac{\mu_i}{\sum_{i=1}^n \mu_i} \cdot (y - y^*) \cdot (c_i - y) \cdot \frac{(x_j - a_{ij})^2}{b_{ij}^3} / b_{ij}^k \tag{15}$$

$$(b_{ij}^k)(t+1) = (b_{ij}^k)(t) + K \triangle b_{ij}^k(t) \tag{16}$$

$$\triangle c_i^k = \frac{\mu_i}{\sum_{i=1}^n \mu_i} \cdot (y - y^*) \tag{17}$$

$$(c_i^k)(t+1) = (c_i^k)(t) + K \triangle c_i^k(t) \tag{18}$$

where each of parameters $\{a_{ij}, b_{ij}, c_i | i \in Z_n, j \in Z_N\}$ is represented as $a_{ij} = \sum_{k=1}^m a_{ij}^k$, $b_{ij} = \Pi_{k=1}^m b_{ij}^k$ and $c_i = \sum_{k=1}^m c_i^k$, respectively.

Eq. (16) means that each server can update the parameter by dividing by b_{ij}^k for the conventional method.

From these results, learning of the fizzy inference system is shown in TableII. The validity of the algorithm is proved from Eqs.(13) to (18). Let us explain the relation between Algorithm A and Table.II.

In Step1, learning data is selected randomly. In Steps 2 and 3, $x_j - a_{ij}$ is computed using the result of each server. In Steps 4 and 5, $(x_j - a_{ij})/b_{ij}$ and Eq.(3) are computed. In Step 6, the denominator of Eq.(4) and the numerator of Eq.(4) for each server are computed. In Step 7, Eq.(4) is computed and each shared result of it is sent to each server. In Steps (8) and (9), $\triangle a_{ij}^k$, $\triangle b_{ij}^k$ and $\triangle c_i^k$ for Eqs.(13), (15), and (17) are computed, respectively, and sent them to each

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

server. In Step10, a_{ij}, b_{ij}, and c_i are updated. In Step 11, if the error E is sufficient small, or the maximum learning time is attained then the algorithm terminates else go to Step 1.

IV. Numerical Simulations

In this section, numerical simulations of function approximation and control problem for conventional and proposed methods are performed. The conventional method means the method without sharing data and the proposed method is ones with $m = 3$ and $m = 10$.

A. Function Approximation

This simulation uses four systems specified by the following functions with 4-dimensional input space $[0, 1]^4$ (for Eqs.(19) and (20)) and $[-1, 1]^4$ (for Eqs.(21) and (22)). The simulation condition is shown in Table III. The numbers of learning and test data selected randomly are 512 and 6400, respectively.

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{37.21} \times \frac{(4\sin(\pi x_3) + 2\cos(\pi x_4) + 6}{12} \tag{19}$$

$$y = \frac{\sin(2\pi x_1) \times \cos(x_2) \times \sin(\pi x_3) \times x_4 + 1.0}{2.0} \tag{20}$$

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{74.42} + \frac{4\sin(\pi x_3) + 2\cos(\pi x_4) + 6}{446.52} \tag{21}$$

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{74.42} + \frac{(3e^{3x_3} + 2e^{-4x_4})^{-0.5} - 0.077}{4.68} \tag{22}$$

Table IV shows the results of comparison between the conventional and the proposed methods. In each box of Table IV, three numbers from the top to the bottom show MSE of learning ($\times 10^{-4}$), MSE of test ($\times 10^{-4}$) and the number of parameters, respectively. The result of simulation is the average value from twenty trials.

The result shows that the accuracy of the conventional and the proposed method is almost the same. Further, it is shown that the results for $m = 3$ and $m = 10$ is also about the same accuracy.

B. Obstacle avoidance and arriving at designated point

In order to show the interpretability of the proposed model, let us perform simulation of control problem. The problem is how the object avoids the obstacle and arrives at the designated place. As shown in Fig.3, the distance r_1 and the angle θ_1 between object and obstacle and the distance r_2 and the angle θ_2 between object and the designated place are selected as input variables, where θ_1 and θ_2 are normalized.

Fuzzy inference rules for conventional and proposed method are constructed from learning data of $P = 400$ points shown in Fig. 4. An obstacle is placed at $(0.5, 0.5)$ and a designated place is placed at $(1.0, 0.5)$. The number of rules for each method is 81 and the number of attributes is 3. The object moves with the vector $\boldsymbol{A}$ at each step, where A_x of $\boldsymbol{A}$ is constant and A_y of $\boldsymbol{A}$ is output variable. Learning for

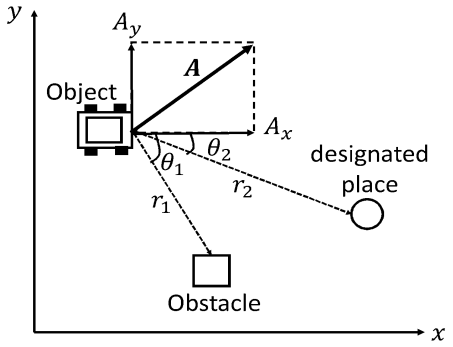


Fig. 3. Simulation on obstacle avoidance and arriving at the designated place, where A_x is constant and A_y is adjusted.

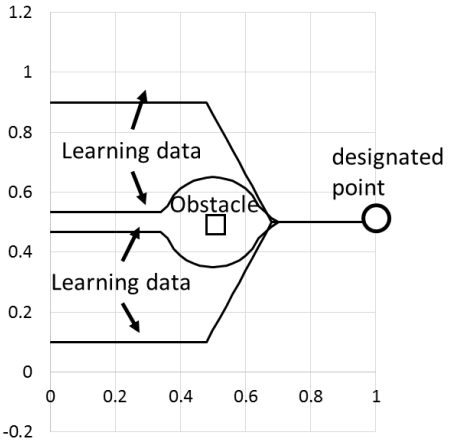


Fig. 4. Learning data to avoid obstacle and arrive at the designated place (1.0, 0.5).

two methods is successful and the following scenarios are performed.

(1)Scenario 1 is simulation for obstacle avoidance and arriving at the designated place when the mobile object stars from various places (See Fig.5). Fig.5 shows the results of moves of object for starting places at $(0.1, 0), (0.2, 0), \cdots, (0.8, 0), (0.9, 0)$ after learning. Simulations are successful for all cases. Fig.5 shows the result of conventional and proposed methods for $m = 10$.

(2)Scenario 2 is simulation for the case where the mobile object avoids obstacle placed at different place and arrives at the different designated place. Simulations with obstacle placed at the place $(0.4, 0.4)$ and arriving at the designated place $(1, 0.6)$ are performed for all cases. The results are successful as shown. Fig.6 shows the result of conventional and proposed methods for $m = 10$.

(3)Scenario 3 is simulation for the case where obstacle moves with the fixed speed. Simulations with obstacle moving with the speed $(0.01, 0.02)$ from the place $(0.3, 1.0)$ to the place $(0.8, 0.0)$ and object arriving at the place $(1, 0.6)$ are performed. Simulations are successful for all cases. Fig.7 shows the results of simulations for conventional and proposed methods for $m = 10$. Since the object does not collide in the obstacle at $t = 30$, thereafter it does not collide in the obstacle.

Lastly, let us consider interpretability of fuzzy rules obtained for the proposed method. Let us consider fuzzy rules constructed for the proposed method of $m = 10$ by learning. Assume that three attributes are short, middle and long for r_1 and r_2, minus, central and plus for θ_1 and θ_2 and left, center and right for the direction of A_y, respectively. Then,

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE II
LEARNING PROCESS OF FUZZY INFERENCE LEARNING FOR SMC.

	Client	k-th Server
Initial condition	The parameters $\{a_{ij}, b_{ij}, c_i\}$ are set randomly, and send $a_{ij}^k (a_{ij} = \sum_{k=1}^m a_{ij}^k)$, $b_{ij}^k (b_{ij} = \Pi_{k=1}^m b_{ij}^k)$ and $c_i^k (c_i = \sum_{k=1}^m c_i^k)$ to each server for $i \in Z_n$, $j \in Z_N$ and $k \in Z_m$. Set $t = 1$.	$\{(\boldsymbol{x}^l)^k, \left(d(\boldsymbol{x}^l)\right)^k \mid l \in Z_L\}$
Step 1	A number l is selected randomly	
Step 2		Compute $(x_j^l)^k - a_{ij}^k$ for $i \in Z_n$ and $j \in Z_N$ and send it to Client.
Step 3	Compute $dist_{ij} = \sum_{k=1}^m \left((x_j^l)^k - a_{ij}^k\right)$ and send $dist_{ij}^k (dist_{ij} = \Pi_{k=1}^m dist_{ij}^k)$ to each server	
Step 4		Compute $p_{ij}^k = dist_{ij}^k / b_{ij}^k$ and $q_{ij}^k = dist_{ij}^k / (b_{ij}^k)^2$ and send them to Client.
Step 5	Compute $\mu_i = \exp\left(-(\Pi_{k=1}^m p_{ij}^k)^2\right)$ and send it to each server	
Step 6	Compute $S = \sum_{i=1}^r \mu_i$	Compute $\sum_{i=1}^n \mu_i c_i^k$ and send it to Client
Step 7	Compute output $y = \sum_{k=1}^m (\sum_{i=1}^n \mu_i c_i^k)/S$ and send $y^k (y = \sum_{k=1}^m y^k)$ to each server	
Step 8		Compute $s^k = c_i^k - y^k$ and $\triangle^k = (d(\boldsymbol{x}^l))^k - y^k$ and send them to Client
Step 9	Compute $\alpha_{ij} = (\sum_{k=1}^m \triangle^k)(\sum_{k=1}^m s^k)(\Pi_{k=1}^m q_{ij}^k)\mu_i/S$, $\beta_{ij} = (\sum_{k=1}^m \triangle^k)(\sum_{k=1}^m s^k)(\Pi_{k=1}^k (p_{ij}^k)^2)\mu_i/S$ and $\gamma_i = (\sum_{k=1}^m \triangle^k)\mu_i/S$ and send them to each server	
Step 10		Update $a_{ij}^k \leftarrow a_{ij}^k + K_a \alpha_{ij}$, $b_{ij}^k \leftarrow b_{ij}^k + K_b \beta_{ij}/b_{ij}^k$ and $c_i^k \leftarrow c_i^k + K_c \gamma_i$
Step 11	If $t \neq T_{max}$ or $E > \theta$ then go to Step 1 with $t \leftarrow t+1$ else the algorithm terminates	

TABLE III
THE INITIAL CONDITIONS FOR SIMULATIONS OF FUNCTION APPROXIMATION

	conventional method	proposed method
K_a	0.01	0.01
K_b	0.01	0.01
K_c	0.1	0.1
θ	1.0×10^{-4}	1.0×10^{-4}
T_{max}	50000	50000
Initial a_{ij}^k	equal intervals	
Initial b_{ij}^k	$\frac{1}{2(d-1)}\times$(the domain of input)	
Initial c_i^k	random on [0,1]	

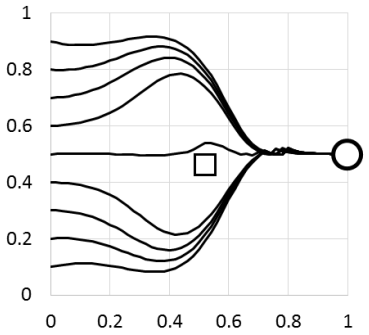
TABLE IV
RESULT OF SIMULATION OF FUNCTION APPROXIMATION

		Eq.(19)	Eq.(20)	Eq.(21)	Eq.(22)
conventional method	Learn	0.10	0.10	0.10	0.10
	Test	2.35	2.09	1.73	2.99
	#Para	729	729	729	729
proposed method ($m = 3$)	Learn	0.10	0.10	0.10	0.10
	Test	3.74	3.64	3.68	4.39
	#Para	2187	2187	2187	2187
proposed method ($m = 10$)	Learn	0.10	0.10	0.10	0.10
	Test	3.62	3.77	3.49	4.14
	#Para	7290	7290	7290	7290

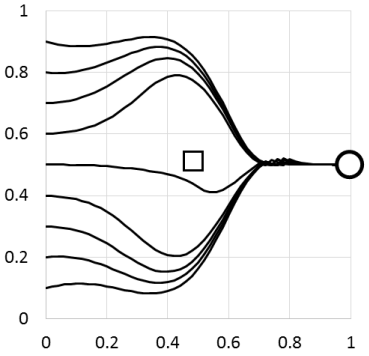
main fuzzy rules for the proposed method are constructed as shown in Table VI. From Table VI, we can get the rules : "If the object approaches to the obstacle, move in the direction away from the object." and "If the object approach to the goal, move toward to the goal.". For example, Rule 1 means that if the object is near the obstacle (r_1 is short.), the object is far from the designated place (r_2 is long.), the object is above the obstacle (θ_1 is plus in Fig.3.), the object is above the designated place (θ_2 is plus.) then the object moves in the direction away from the obstacle (A_y is left(plus) in

TABLE V
INITIAL CONDITION FOR SIMULATION OF OBSTACLE AVOIDANCE.

	Conventional	Proposed
T_{max}	5000	1000
K_c	0.001	0.001
K_b	0.001	0.001
K_w	0.05	0.05
d	3	3
Initial a_{ij}	equal intervals	
Initial b_{ij}	$\frac{1}{2(d-1)}\times$(the domain of input)	
Initial c_i	0.0	
♯parameters	729	105

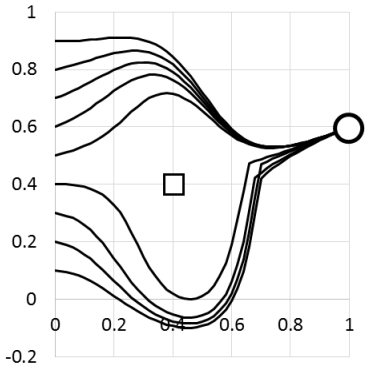


(a) Conventional method

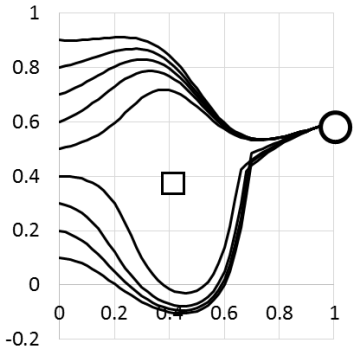


(b) Proposed method with $m = 10$

Fig. 5. Simulation for obstacle avoidance and arriving at the designated place starting from various places after learning for the proposed method with $m = 10$.

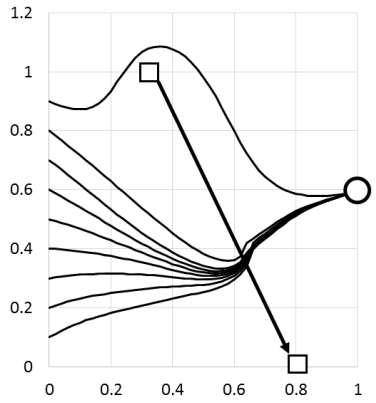


(a) Conventional method

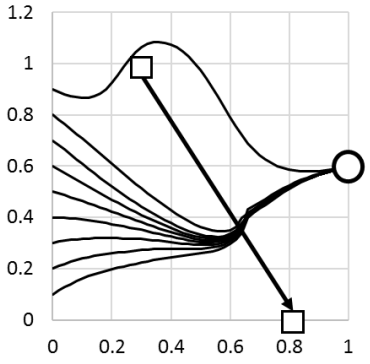


(b) Proposed method with $m = 10$

Fig. 6. Simulation for obstacle placed at the different place $(0.4, 0.4)$ and arriving at the different place $(1.0, 0.6)$.



(a) Conventional method



(b) Proposed method with $m = 10$

Fig. 7. Simulation for moving obstacle avoidance with fixed speed and the different designated place $(1.0, 0.6)$.

TABLE VI
MAIN FUZZY RULES OBTAINED FOR THE PROPOSED METHOD OF $m = 10$.

	r_1	r_2	θ_1	θ_2	A_y
Rule 1	short	long	plus	plus	left
Rule 2			minus	minus	right
Rule 3	middle	middle	plus	plus	right
Rule 4			minus	minus	left

Fig.3). The result shows that obtained fuzzy rules are near our intuition.

V. CONCLUSION

In this paper, we proposed a learning method(fuzzy modeling) of fuzzy inference system for SMC and proved the validity of it. Further, the performance of the proposed method was shown in numerical simulations. The advantage of fuzzy modeling compared to other learning methods is that input and output relation for learning data is represented as fuzzy rules and it is easy to understand the interpretability of input and output learning data. The idea of our study is to perform interpretable fuzzy modeling as "privacy preserving fuzzy modeling=shared data + parallel algorithm". That is, we performed to find the representation of shared data and to construct parallel algorithm. In the future work, we will consider improved methods to reduce the computation of client and develop AUI(Application User Interface) for the client.

REFERENCES

[1] C. C. Aggarwal, and P. S. Yu, "Privacy-Preserving Data Mining: Models and Algorithms", ISBN 978-0-387-70991-8, Springer-Verlag, 2009.
[2] S. Subashini, and V. Kavitha, "A survey on security issues in service delivery models of cloud computing", J. Network and Computer Applications, Vol.34,pp.1-11, 2011.
[3] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices", STOC2009, pp.169-178, 2009.
[4] A. Shamir, "How to share a secret", Comm. ACM, Vol. 22, No. 11, pp. 612-613, 1979.
[5] J. Yuan, S. Yu, "Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing", IEEE Trans. on Parallel and Distributed Systems, Vol.25, Issue 1, pp.212-221, 2013.
[6] A. Beimel, "Secret-sharing schemes: a survey", in Proc. of the Third international conference on Coding and cryptology (IWCC 11), 2011.
[7] R. Canetti, et al., "Adaptively secure multi-party computation", STOC' 96, pp. 639-648, 1996.
[8] A. Ben-David, et al., "Fair play MP: a system for secure multi-party computation", ACM CCS' 08, 2008.
[9] S. S. Rathna, T. Karthikeyan, "Survey on Recent Algorithms for Privacy Preserving Data mining", International Journal of Computer Scienceand Information Technologies, Vol. 6 (2), pp. 1835-1840, 2015.
[10] K. Chida, et al., "A Lightweight Three-Party Secure Function Evaluation with Error Detection and Its Experimental Result", IPSJ Journal Vol. 52 No. 9, pp. 2674-2685, Sep. 2011(in Japanese).
[11] Y. Miyanishi, A. Kanaoka, F. Sato, X. Han, S. Kitagami, Y.Urano, N. Shiratori, "New Methods to Ensure Security to Increase User's Sense of Safety in Cloud Services", Proc. of The 14th IEEE Int. Conference on Scalable Computing and Communications (ScalCom-2014), pp.859-865, Bali, Dec.2014.
[12] H. Miyajima, N. Shigei, H. Miyajima, Y. Miyanishi, S. Kitagami and N. Shiratori, "New Privacy Preserving Back Propagation Learning for Secure Multiparty Computation", IAENG International Journal of Computer Science, Vol.43, No.3, pp.270-276, 2016.
[13] H. Miyajima, N. Shigei, H. Miyajima, Y. Miyanishi, S. Kitagami, N. Shiratori, "New Privacy Preserving Clustering Methods for Secure Multiparty Computation", Artificial Intelligence Research, Vol. 6, No. 1, 2017.
[14] M. M. Gupta, L. Jin and N. Homma, "Static and Dynamic Neural Networks", IEEE Press, 2003.

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)