Neural Gas and k-means Methods with Reduced Communication Costs for Secure Distributed Processing

Hirofumi Miyajima, Noritaka Shigei, Hiromi Miyajima, and Norio Shiratori

Abstract—**From the viewpoint of developing machine learning as a safe and secure AI method, research in this field has focused on machine learning on a single server using encrypted or noise-added data, or distributed processing on multiple servers using decomposition data or distributed dataset. However, few learning methods have been proposed that achieve both data confidentiality and usability at a high level. The authors proposed a machine learning method based on secure distributed processing with decomposition data and parameters. However, although this distributed processing method is superior in terms of security, it is computationally expensive. To solve this problem, the authors proposed a learning method that updates only one element of each decomposed parameter. In this paper, we propose an improvement method in communication cost for the NG and k-means methods and show its effectiveness. The proposed method means that the computational complexity of the NG and k-means methods with distributed processing can be almost the same as that of the conventional NG and k-means methods with a single server.**

Index Terms—**Secure Distributed Processing, Decomposition Data and Parameters, Neural Gas, k-means, Computational Complexity, Communication costs.**

I. Introduction

THE realization of a super-smart society requires AI-based big data analysis to highly integrate cyberspace and physical space (the real world) [1]. AI-based big data analysis will bring valid information to the real world faster. On the other hand, to build a safe and secure smart society, it is necessary to develop AI methods to protect the privacy of Big Data in cyberspace [2]. In this field, from the viewpoint of developing machine learning as a secure and safe AI method for users, research is being conducted in two directions: machine learning on a single server using encrypted or noise-added data, and distributed processing on multiple servers using decomposition data or distributed dataset, and research combining these two methods [3], [4], [5]. However, few learning methods have been proposed that balance data confidentiality and usability at a high level. In the previous paper, the authors proposed a method that decomposes the data and parameters and performs learning in secure distributed processing [6]. However, this method using distributed processing, while superior in terms of security, is computationally expensive. Hence, to solve this problem, we proposed a learning method that reduces the computational complexity by updating only one element of each decomposed parameter [7].

In this paper, we propose an improvement method in communication cost for the NG and k-means methods and show its effectiveness. The proposed method means that the computational complexity of the NG and k-means methods with distributed processing can be almost the same as that of the conventional NG and k-means methods with a single server. The main contents of this paper are as follows. Chapter II describes the concept of secure distributed processing, the NG method, and the k-means method. Chapter III describes the decomposition data and parameters used in this paper, as well as the learning methods for secure distributed processing that reduce the computational complexity of the NG and k-means methods, and explains the differences in computational complexity. In Chapter IV, we perform numerical simulations of clustering and compare the results of the proposed and conventional methods. Chapter V summarizes and prospects for the research.

II. Preliminaries

A. The concept of Secure Distributed Processing

We show the concept of a method to realize machine learning by secure distributed processing while maintaining data confidentiality using decomposition data and parameters [6]. In this method, we use a system consisting of a central server (denoted as Server 0) and Q servers, as shown in Fig.1. Let x be a scalar data and $f(x)$ be the objective function. First, data x and $f(x)$ are randomly divided into Q elements as $x = \sum_{q=1}^{Q} x^{(q)}$, $f(x) = \sum_{q=1}^{Q} f_q(x^{(q)})$ and each element is stored on a server. The q-th server computes the function $g_q(x^{(q)})$ defined by the parameters and sends the difference $\Delta f_q(x^{(q)}) = f_q(x^{(q)}) - g_q(x^{(q)})$ to Server 0. Server 0 integrates them and computes $\Delta f(x) = \sum_{q=1}^{Q} \Delta f_q(x^{(q)})$. If $|\Delta f(x)|$ is sufficiently small, the calculation process terminates. Otherwise, the error $\Delta f(x)$ is sent to each server, and the function $g_q(x^{(q)})$ is updated. Further, the same calculation process is repeated.

The problem is how to update the function $g_q(x^{(q)})$ for each server to realize $f(x) \simeq \sum_{q=1}^{Q} g_q(x^{(q)})$.

B. Steepest Descent Method

Machine learning aims to estimate the input-output relationship for a given learning data by estimating the parameters for a model. In this section, we explain the steepest descent method (SDM) [8].

SDM is a method designed to find the parameter $\boldsymbol{\theta}$ that minimizes the objective function $J(\boldsymbol{\theta})$.

Manuscript received March 29, 2023; revised 13 April, 2023.

H. Miyajima is with Nagasaki University, 1-14 Bunkyomachi, Nagasaki city, Nagasaki, Japan e-mail: miyajima@nagasaki-u.ac.jp.

N. Shigei and H. Miyajima are with Kagoshima University.

N. Shiratori is with Chuo University.

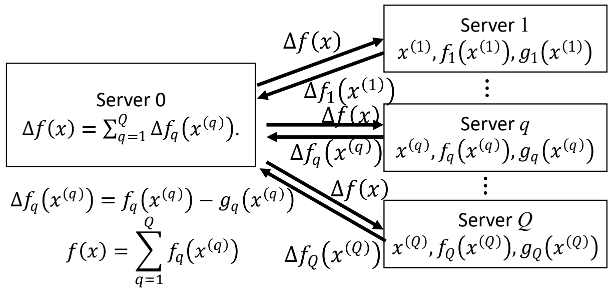
Server 0
$\Delta f(x) = \sum_{q=1}^{Q} \Delta f_q(x^{(q)})$.
Server 1
$x^{(1)}, f_1(x^{(1)}), g_1(x^{(1)})$
Server q
$x^{(q)}, f_q(x^{(q)}), g_q(x^{(q)})$
Server Q
$x^{(Q)}, f_Q(x^{(Q)}), g_Q(x^{(Q)})$
$\Delta f(x)$
$\Delta f_1(x^{(1)})$
$\Delta f(x)$
$\Delta f_q(x^{(q)})$
$\Delta f(x)$
$\Delta f_Q(x^{(Q)})$
$\Delta f_q(x^{(q)}) = f_q(x^{(q)}) - g_q(x^{(q)})$
$f(x) = \sum_{q=1}^{Q} f_q(x^{(q)})$

Fig. 1. An example of a secure distributed system

For the parameters, we repeatedly apply the following equation to get close to the target values using the gradient method.

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta \nabla J(\boldsymbol{\theta}), \tag{1}$$

where η is the learning coefficient, which is a real number that determines the step size of the update steps. In addition, $\nabla J(\boldsymbol{\theta})$ is the amount of update for the parameter $\boldsymbol{\theta}$.

The parameter $\boldsymbol{\theta}$ can be used to obtain a local solution of the function $J(\boldsymbol{\theta})$ by repeating Eq.(1).

If SDM is used for machine learning, three types of learning methods have been reported in the relevant literature [6]: online learning, mini-batch learning, and batch learning, depending on how the objective function $J(\boldsymbol{\theta})$ in Eq.(1) is given. We explain it below.

For any natural number i, let $Z_i = \{1, 2, \cdots, i\}$ and $Z_i^* = \{0, 1, \cdots, i\}$. Let D be the set of learning data, and $|D| = L$. In addition, the set D is partitioned into $D = \bigcup_{l=1}^{N} B_l$ ($B_i \cap B_j = \phi$) and N subsets $B_1, \cdots, B_N$, and $|B_l| = b_l$ ($l \in Z_N$), where $L = \sum_{l=1}^{N} b_l$. The learning method using SDM is as follows [6]. Let T_{max} be the maximum number of learning time. First, we set $t = 1$.

[**S**tep 1]

Select a natural number $l \in Z_N$ randomly and determine a subset B_l of the learning data to be used in updating the parameters.

[**S**tep 2]

Repeat learning steps of using Eq.(1) for the set B_l.

[**S**tep 3]

If $t = T_{max}$, algorithm terminates. Otherwise, $t \leftarrow t + 1$ and go to Step 1.

This method is called online learning in the case of $N = L$, batch learning in the case of $N = 1$, and mini-batch learning in other cases.

Machine learning methods based on SDM include the Back Propagation (BP) method, k-means method, fuzzy modeling, and so on [8], [9].

C. NG and k-means methods

In this section, we describe the NG method, which is one of unsupervised learning methods based on SDM [9]. NG includes k-means method as the special case. Vector quantization, which is realized by NG, approximates a large amount of data with a small amount of data. We describe the case where dataset $X = \{\boldsymbol{x}^l | l \in Z_L\} \subseteq R^n$ is encoded using a finite set of reference vectors $W = \{\boldsymbol{w}_i | i \in Z_r\}$, where n and R are a natural number and the set of real numbers, respectively. Let $e_i(\boldsymbol{x}) \in Z_{r-1}^*$ be the neighborhood rank of the i-th reference vector $\boldsymbol{w}_i$ for data $\boldsymbol{x}$. That is, $\boldsymbol{w}_i$ is the $(e_i(\boldsymbol{x}) + 1)$-th nearest reference vector to the data $\boldsymbol{x}$. Here, the distance between $\boldsymbol{w}_i$ and $\boldsymbol{x}$ is defined by the Euclidean distance $||\boldsymbol{x}^l - \boldsymbol{w}_i||$. The degree to which the dataset X is approximated by the set W is defined by the following function.

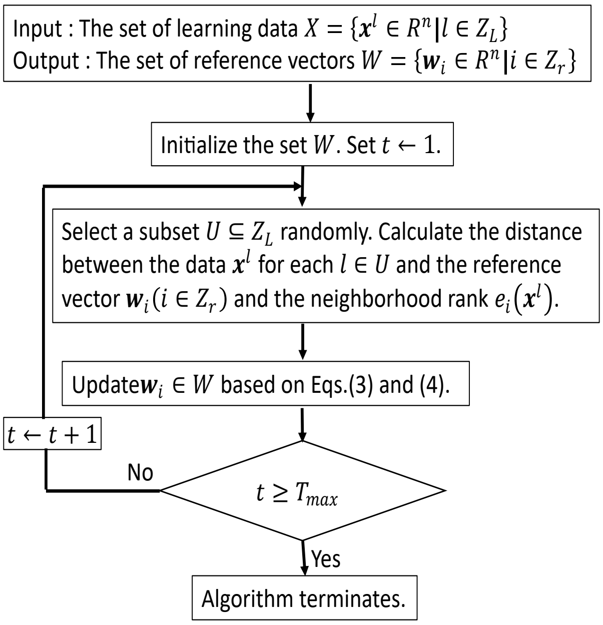
Input : The set of learning data $X = \{\boldsymbol{x}^l \in R^n | l \in Z_L\}$
Output : The set of reference vectors $W = \{\boldsymbol{w}_i \in R^n | i \in Z_r\}$
Initialize the set W. Set $t \leftarrow 1$.
Select a subset $U \subseteq Z_L$ randomly. Calculate the distance between the data $\boldsymbol{x}^l$ for each $l \in U$ and the reference vector $\boldsymbol{w}_i (i \in Z_r)$ and the neighborhood rank $e_i(\boldsymbol{x}^l)$.
Update$\boldsymbol{w}_i \in W$ based on Eqs.(3) and (4).
$t \leftarrow t + 1$
No
$t \geq T_{max}$
Yes
Algorithm terminates.

Fig. 2. The flowchart of NG method. This method is called online learning in the case of $|U| = 1$, batch learning in the case of $|U| = L$, and mini-batch learning in other cases.

$$E = \frac{1}{|X|} \sum_{\boldsymbol{x}^l \in X} \sum_{i=1}^{r} \frac{\exp(e_i(\boldsymbol{x}^l)/\lambda)}{\sum_{i'=1}^{r} \exp(-e_{i'}(\boldsymbol{x}^l)/\lambda)} ||\boldsymbol{x}^l - \boldsymbol{w}_i||^2 \tag{2}$$

where λ is a real value.

Then, each reference vector $\boldsymbol{w}_i \in W$ is updated based on SDM as shown below [9].

A subset $U \subseteq X$ of natural numbers is selected randomly. The update amount $\Delta \boldsymbol{w}_i$ is calculated as follows.

$$\Delta \boldsymbol{w}_i = \varepsilon \sum_{l \in U} \exp(-e_i(\boldsymbol{x}^l)/\lambda)(\boldsymbol{x}^l - \boldsymbol{w}_i) \tag{3}$$

where $\varepsilon \in [0, 1]$.

By using the update amount $\Delta \boldsymbol{w}_i$, each reference vector $\boldsymbol{w}_i$ is updated as follows.

$$\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i + \Delta \boldsymbol{w}_i \tag{4}$$

Eqs.(3) and (4) imply that the closer $\boldsymbol{w}_i$ is to $\boldsymbol{x}$, the closer $\boldsymbol{w}_i$ is to $\boldsymbol{x}$ based on its closeness. NG method is shown in Fig. 2 [9], where ε and T_{max} denote the learning coefficient and the maximum number of learning steps, respectively.

As a special case of NG, k-means is obtained when $\lambda \rightarrow 0$.

III. NG AND K-MEANS METHODS FOR SECURE DISTRIBUTED PROCESSING

In this chapter, we propose the NG and k-means methods for secure distributed processing.

A. Data representation for secure distributed processing

The learning data $\boldsymbol{x}^l = (x_1^l, \cdots, x_j^l, \cdots, x_n^l)$ and reference vector $\boldsymbol{w}_i = (w_{i1}, \cdots, w_{ij}, \cdots, w_{in})$ are decomposed into Q elements (pieces) and each element is stored on a server as follows.

$$x_j^l = \sum_{q=1}^{Q} x_j^{l(q)} \tag{5}$$

$$w_{ij} = \sum_{q=1}^{Q} w_{ij}^{(q)} \tag{6}$$

where $\boldsymbol{x}^{l(q)} = (x_1^{l(q)}, \cdots, x_n^{l(q)})$ and $\boldsymbol{w}_i^{(q)} = (w_{i1}^{(q)}, \cdots, w_{in}^{(q)})$ are the q-th element of $\boldsymbol{x}^l$ and $\boldsymbol{w}_i$ and stored in the q-th server.

We can obtain the distance between the data $\boldsymbol{x}^l$ and the reference vector $\boldsymbol{w}_i$ by using the elements $x_j^{l(q)}$ and $w_{ij}^{(q)}$ for ($l \in Z_L$, $j \in Z_n$, $i \in Z_r$, $q \in Z_Q$). The distance $||\boldsymbol{x}^l - \boldsymbol{w}_i||^2$ between the data $\boldsymbol{x}^l$ and the reference vector $\boldsymbol{w}_i$ is defined by the following formula using decomposition data and parameters.

$$||\boldsymbol{x}^l - \boldsymbol{w}_i||^2 = \sum_{j=1}^{n} (\sum_{q=1}^{Q} (x_j^{l(q)} - w_{ij}^{(q)}))^2 \tag{7}$$

Eq.(7) shows that the distance is calculated in the distributed form.

B. The conventional NG method for secure distributed processing

Based on the distance $||\boldsymbol{x}^l - \boldsymbol{w}_i||^2$ of Eq.(7), we obtain the neighborhood rank $e_i(\boldsymbol{x}^l)$ of each reference vector $\boldsymbol{w}_i$ for the data $\boldsymbol{x}^l$. In this case, the evaluation function E for NG is obtained as Eq.(8) as follows.

$$\begin{aligned} E &= \frac{1}{|X|} \sum_{\boldsymbol{x}^l \in X} \sum_{i=1}^{r} \frac{\exp(-e_i(\boldsymbol{x}^l)/\lambda)}{\sum_{i'=1}^{r} \exp(-(e_{i'}(\boldsymbol{x}^l))/\lambda)} \\ & \sum_{j=1}^{n} (\sum_{q=1}^{Q} (x_j^{l(q)} - w_{ij}^{(q)}))^2 \end{aligned} \tag{8}$$

Then, we can obtain the update amount $\Delta \boldsymbol{w}_i = (\Delta w_{i1}, \cdots, \Delta w_{in})$ for the element $\boldsymbol{w}_i^{(q)}$ from Eq.(8) in each server as follows.

$$\Delta w_{ij} = \varepsilon_1 \sum_{\boldsymbol{x}^l \in X} \exp(-(e_i(\boldsymbol{x}^l))/\lambda) \sum_{q=1}^{Q} (x_j^{l(q)} - w_{ij}^{(q)}) \tag{9}$$

where ε_1 is the learning coefficient.

The authors proposed NG method for the secure distributed processing using Eq.(9) as shown in TABLE I [6].

As a special case, k-means method is defined as $\lambda \to 0$.

C. The conventional NG and k-means methods reducing computational complexity for secure distributed processing

In this section, we show the NG method reducing computational complexity [7].

Let $w_{ij}^{(q)}(t)$ and $\Delta w_{ij}(t)$ be the divided reference vector $w_{ij}^{(q)}$ and the update amount Δw_{ij} at the step t. The update formula for the reference vector w_{ij} shown in the Eq.(4) becomes the following equation.

$$\sum_{q=1}^{Q} w_{ij}^{(q)}(t+1) = \sum_{q=1}^{Q} w_{ij}^{(q)}(t) + \Delta w_{ij}(t) \tag{10}$$

We can rewrite the Eq.(10) using a natural number $q_0 \in Z_Q$ as follows.

$$\begin{aligned} \sum_{q \neq q_0}^{Q} w_{ij}^{(q)}(t+1) \quad &+ \quad w_{ij}^{(q_0)}(t+1) = \\ \sum_{q \neq q_0}^{Q} w_{ij}^{(q)}(t) \quad &+ \quad (w_{ij}^{(q_0)}(t) + \Delta w_{ij}(t+1)) \end{aligned} \tag{11}$$

From the Eq.(11), we can obtain the update formula of $w_{ij}^{(q)}(t+1)$ as follows.

$$w_{ij}^{(q)}(t+1) = \begin{cases} w_{ij}^{(q_0)}(t) + \Delta w_{ij}(t) & (q = q_0) \\ w_{ij}^{(q)}(t) \quad (q \neq q_0) \end{cases} \tag{12}$$

where the number q_0 is selected arbitrary.

TABLE II shows NG algorithm based on Eq.(12) [7]. The difference between TABLEs I and II is that the former method updates all elements of each parameter, while the latter updates only one element of it.

D. The proposed NG and k-means methods reducing communication consts for secure distributed processing

In the method of TABLE II, each server needs to send and receive some results. For example, as Steps 2 and 3 of TABLE II show, for the calculation of D_{ij}^l, each server needs to send $|U| \times |r| \times |n|$ results to Server 0. Therefore, the larger the cardinality of set U, the larger the amount of data sent from each server to Server 0.

In this section, we propose the NG method reducing communication costs of TABLE II as follows.

In TABLE II, Server q ($q \in Z_Q$) sends the calculation result $D_{ij}^{l(q)}$ ($l \in Z_L$, $i \in Z_r$, $j \in Z_n$, $q \in Z_Q$) to Server 0 as follows.

$$D_{ij}^{l(q)} = x_{ij}^{l(q)} - w_{ij}^{(q)} \tag{13}$$

In Server 0, each of $D_{ij}^{l(q)}$ ($l \in Z_L$, $i \in Z_r$, $j \in Z_n$) as shown Eq.(14) needs to calculate the distance between the data $\boldsymbol{x}^l$ and the reference vector $\boldsymbol{w}_i$.

$$\begin{aligned} D_{ij}^l &= \sum_{q=1}^{Q} D_{ij}^{l(q)} \\ &= \sum_{q=1}^{Q} (x_{ij}^{l(q)} - w_{ij}^{(q)}) \end{aligned} \tag{14}$$

Let $D_{ij}^{l(q)}(t)$ be the value $D_{ij}^{l(q)}$ at the step t. Assume that Server $q_0 (\in Z_Q)$ is selected at the step $t+1$. From Eq.(12), as

TABLE I
SECURE DISTRIBUTED PROCESSING OF NG WITH DECOMPOSITION DATA [6].

	Server 0	Server q
Initialize	Determine the values λ. Set $t = 1$ and the ε_{int} and ε_{fin}.	Store $\{x_j^{l(q)} \| l \in Z_L, j \in Z_n\}$. Initialize $\{w_{ij}^{(q)} \| i \in Z_r, j \in Z_n\}$.
Step 1	Select a subset $U \subseteq Z_L$ randomly and send to each server.	
Step 2		Calculate $D_{ij}^{l(q)} = x_j^{(q)} - w_{ij}^{(q)}$ $(l \in U, i \in Z_r, j \in Z_n)$ and send to Server 0.
Step 3	Calculate $\|\|\boldsymbol{x}^l - \boldsymbol{w}_i\|\|^2$ and Δw_{ij} based on Eqs.(7) and (9). Send $\Delta w_{ij}^{(q)}$ to each server.	
Step 4		Update $\{w_{ij}^{(q)} \| i \in Z_r, j \in Z_n\}$ as follows. $w_{ij}^{(q)} \leftarrow w_{ij}^{(q)} + \Delta w_{ij}$
Step 5	If $t = T_{max}$, the algorithm terminates. Otherwise, set $t \leftarrow t+1$ and go to Step 1.	

TABLE II
ALGORITHM OF THE PROPOSED NG METHOD.

	Server 0	Server q
Initialize	Determine the values ε_{int} and ε_{fin}. Set $t = 1$.	Store $\{x_j^{l(q)} \| l \in Z_L, j \in Z_n\}$. Initialize $\{w_{ij}^{(q)} \| i \in Z_r, j \in Z_n\}$.
Step 1	Select the set of natural numbers $U \subseteq Z_L$ randomly and send to each server.	
Step 2		Calculate $D_{ij}^{l(q)} = (x_j^{l(q)} - w_{ij}^{(q)})$ $(l \in U, i \in Z_r, j \in Z_n)$ and send to Server 0.
Step 3	Calculate $D_{ij}^l = \sum_{q=1}^{Q} D_{ij}^{l(q)}$ and $\|\|\boldsymbol{x}^l - \boldsymbol{w}_i\|\| = \|D_{ij}^l\|$. Based on $\|\|\boldsymbol{x}^l - \boldsymbol{w}_i\|\|^2$ and Eq.(9), calculate $\exp(-e_i(x^{(l)})/\lambda)$ and Δw_{ij}. Select a number $q_0 \in Z_Q$ and send Δw_{ij} to Server q_0.	
Step 4		If $q = q_0$, update $\{w_{ij}^{(q)} \| i \in Z_r, j \in Z_n\}$ as follows. $w_{ij}^{(q)} \leftarrow w_{ij}^{(q)} + \Delta w_{ij}$
Step 5	If $t = T_{max}$, the algorithm terminates. Otherwise, Set $t \leftarrow t+1$ and go to Step 1.	

$w_{ij}^{(q_0)}(t+1) = w_{ij}^{(q_0)}(t) + \Delta w_{ij}(t)$ and $w_{ij}^{(q)}(t+1) = w_{ij}^{(q)}(t)$ for $q \neq q_0$. Therefore, we can obtain the following relation.

$$
\begin{aligned}
D_{ij}^l(t+1) &= \sum_{q=1}^{Q} (x_{ij}^{l(q)} - w_{ij}^{(q)}(t+1)) \\
&= \sum_{q=1, q\neq q_0}^{Q} (x_{ij}^{l(q)} - w_{ij}^{(q)}(t)) \\
&\quad + (x_{ij}^{l(q)} - w_{ij}^{(q)}(t+1)) \\
&= \sum_{q=1, q\neq q_0}^{Q} (x_{ij}^{l(q)} - w_{ij}^{(q)}(t)) \\
&\quad + (x_{ij}^{l(q)} - (w_{ij}^{(q)}(t) + \Delta w_{ij}(t)) \\
&= \sum_{q=1, q\neq q_0}^{Q} (x_{ij}^{l(q)} - w_{ij}^{(q)}(t)) \\
&\quad + (x_{ij}^{l(q)} - w_{ij}^{(q)}(t)) - \Delta w_{ij}(t) \\
&= \sum_{q=1}^{Q} (x_{ij}^{l(q)} - w_{ij}^{(q)}(t)) - \Delta w_{ij}(t) \\
&= D_{ij}^l(t) - \Delta w_{ij}(t) \qquad (15)
\end{aligned}
$$

That is, the value D_{ij}^l at the step $t+1$ can be updated by using the update amount Δw_{ij} at the step t for the reference vector as follows.

$$D_{ij}^l \leftarrow D_{ij}^l - \Delta w_{ij} \qquad (16)$$

where

$$\Delta w_{ij} = \varepsilon \sum_{l \in U} \exp(-(e_i(\boldsymbol{x}^l))/\lambda) \sum_{q=1}^{Q} D_{ij}^{l(q)} \qquad (17)$$

As a result, the Step 2 of TABLE II is omitted without the calculation of D_{ij}^l at the first step. That is, the update of D_{ij}^l can be computed in Server 0 using Δw_{ij}.

TABLE III shows the improved algorithm by using Eq.(16). In Steps 1 and 2, $D_{ij}^{l(q)}$ $(l \in Z_L,\ i \in Z_r,\ j \in Z_n)$ is calculated and stored in Server 0. In Step 3, the update amount Δw_{ij} is calculated in Server 0. In Step 4, the value $D_{ij}^{l(q)}$ is updated in Server 0.

In the methods shown in TABLEs I and II, $D_{ij}^{l(q)} = x_j^{(q)} - w_{ij}^{(q)}$ in each server is needed to calculate the value D_{ij}^l. On the other hand, in the proposed method shown in TABLE III, the value $D_{ij}^{l(q)}$ can be updated in Server 0 without using the calculation result in other servers. As a result, the proposed method can reduce communication costs compared to the conventional methods.

TABLE IV shows the numbers of parameters and communication costs between Server 0 and other servers, respectively. On TABLE IV, $|W|$ and T_{max} mean the number of reference vectors and the maximum numbers of learning, respectively. $T_R(q)$ and $T_S(q)$ for $q \in Z_Q$ mean the total amount of numbers of transmissions and receptions among Server 0 and q servers, respectively. Further, # Parameters means the number of parameters.

TABLE III
ALGORITHM OF THE PROPOSED NG METHOD.

	Server 0	Server q
Initialize	Determine the values ε_{int} and ε_{fin}. Set $t = 1$.	Store $\{x_j^{l(q)} \mid l{\in}Z_L, j \in Z_n\}$. Initialize $\{w_{ij}^{(q)} \mid i{\in}Z_r, j \in Z_n\}$.
Step 1		Calculate $D_{ij}^{l(q)} = (x_j^{l(q)} - w_{ij}^{(q)})$ $(l{\in}Z_L, i{\in}Z_r, j{\in}Z_n)$ and send to Server 0.
Step 2	Calculate $D_{ij}^l = \sum_{q=1}^{Q} D_{ij}^{l(q)}$	
Step 3	Select the set of natural numbers $U{\subseteq}Z_L$ randomly. Based on D_{ij}^l $(l{\in}U)$ and Eq.(9), calculate Δw_{ij}. Select a number $q_0{\in}Z_Q$ and send Δw_{ij} to server q_0.	
Step 4	Update $\{D_{ij}^l \mid l{\in}Z_L, i{\in}Z_r, j{\in}Z_n\}$ as follows. $D_{ij}^l{\leftarrow}D_{ij}^l - \Delta w_{ij}$	If $q = q_0$, update $\{w_{ij}^{(q)} \mid i{\in}Z_r, j{\in}Z_n\}$ as follows. $w_{ij}^{(q)}{\leftarrow}w_{ij}^{(q)} + \Delta w_{ij}$
Step 5	If $t = T_{max}$, the algorithm terminates. Otherwise, Set $t{\leftarrow}t + 1$ and go to Step 3.	

TABLE IV
COMPARISON OF THE NUMBER OF PARAMETERS AND THE COMMUNICATION COSTS FOR CONVENTIONAL AND PROPOSED NG METHODS.

	# Parameters	Communication cost
A	$\lvert W\rvert{\times}\lvert U\rvert$	0
B	$\lvert W\rvert{\times}\lvert U\rvert{\times}Q$	$\lvert W\rvert{\times}\lvert U\rvert(T_R(Q) + T_S(Q))T_{max}$
C	$\lvert W\rvert{\times}\lvert U\rvert$	$\lvert W\rvert{\times}\lvert U\rvert(T_R(1) + T_S(1))T_{max}$
D	$\lvert W\rvert{\times}\lvert U\rvert$	$\lvert W\rvert{\times}\lvert U\rvert T_S(1)T_{max}$

TABLE V
THE DATASET FOR NUMERICAL SIMULATIONS

	Iris	Wine	Sonar	BCW	Spam
#data : L	150	178	208	683	4601
#input : n	4	13	60	9	57
#class : R	3	3	2	2	2

From the result of TABLE IV, the computational complexity of Method A is considered to be almost the same as the computational complexity of Method D.

Methods A, B, C and D mean ones defined in Fig.2, TABLEs I, II and III, respectively. As a result, Method D is superior in both the number of parameters and the communication costs to the conventional methods B and C.

In this chapter, we proposed a method to reduce communication costs for the NG method in TABLE II (as well as the k-means method). This method is also applicable to the NG method in TABLE I. Furthermore, it seems to be equally applicable to the BP method for secure distributed processing in Ref.[7].

IV. NUMERICAL SIMULATIONS

In this chapter, we perform numerical simulations on clustering of five datasets, Iris, Wine, Sonar, BCW and Spam as benchmark problems [10], to compare the proposed method with the conventional methods. TABLE V shows the details of data used in the simulations, where #data, #input and #class mean the numbers of data, inputs and classes, respectively.

A. Results for NG methods

In this section, we compare the proposed NG method with the conventional ones.

In the proposed method, each of data and reference vectors is divided into five elements, i.e., Q=5. The maximum numbers of learning times are 15000 for Iris, 18000 for Wine, 21000 for Sonar, 70000 for BCW and 50000 for Spam, respectively. TABLE VI shows the results of the evaluation function (denoted as MSE)($\times 10^{-2}$) as Eq.(2) and global purity (denoted as GP) for each dataset as Eq.(18), respectively.

$$\mathrm{GP} = \frac{1}{L}\sum_{i\in Z_r}\max_{j\in Z_r}(n_{ij}){\times}100(\%) \tag{18}$$

where n_{ij} is the number of data belonging to the i-th cluster and the j-th actual class.

In general, the higher the accuracy and the smaller the number of parameters, the more desirable. Here, each value in TABLE VI is the average of 20 trials. Methods A1, A2 and A3 mean the conventional NG methods as shown in Fig.2 for $|U| = 1$, $|U| = L$ and $|U| = L/3$, respectively. That is, A1, A2 and A3 are the cases of online, batch and mini-batch learning, respectively. Likewise, methods B1, B2 and B3 mean the NG methods as shown in TABLE I for $|U| = 1$, $|U| = L$ and $|U| = L/3$, respectively. Methods C1, C2 and C3 mean the NG methods as shown in TABLE II for $|U| = 1$, $|U| = L$ and $|U| = L/3$, respectively. Methods D1, D2 and D3 mean the proposed NG methods as shown in TABLE III for $|U| = 1$, $|U| = L$ and $|U| = L/3$, respectively.

The results in TABLEs IV and VI show that the proposed method reduces the computation costs of the NG method while maintaining accuracy compared to the conventional methods.

B. Results for k-means methods

In this section, we perform numerical simulations of clustering for the k-means methods. The evaluation values and clustering accuracies shown in Eq.(2) with $\lambda{\to}0$ are compared for the conventional methods. In the proposed method, each of data and reference vectors is divided into five elements, i.e., Q=5. The maximum numbers of learning in numerical simulations are the same as the case of NG. TABLE VII shows the result for the k-means methods as Eq.(2) ($\times 10^{-2}$) and global purity as Eq.(18), respectively.

Each value in TABLE VII is the average of 20 trials. Methods A1, A2 and A3 are defined as the conventional NG methods with $\lambda{\to}0$ in Fig.2 for $|U| = 1$, $|U| = L$ and $|U| = L/3$, respectively. Likewise, methods B1, B2 and B3 mean the NG methods with $\lambda{\to}0$ in TABLE I for $|U| = 1$, $|U| = L$ and $|U| = L/3$, respectively. Methods C1, C2

TABLE VI
SIMULATION RESULTS FOR THE CONVENTIONAL AND PROPOSED NG METHODS

		Iris	Wine	Sonar	BCW	Spam
A1	GP(%)	4.1	6.2	45.1	3.6	24.9
	MSE	0.6	4.8	69.1	14.7	8.7
A2	GP(%)	4.0	6.7	45.2	3.5	27.0
	MSE	0.6	4.7	67.5	14.3	8.5
A3	GP(%)	4.0	7.3	45.0	3.5	24.2
	MSE	0.6	4.7	67.6	14.3	8.5
B1	GP(%)	4.1	7.2	45.0	3.6	25.4
	MSE	0.6	4.8	69.2	14.6	8.7
B2	GP(%)	4.0	6.8	45.2	3.5	28.0
	MSE	0.6	4.7	67.9	14.3	8.5
B3	GP(%)	4.0	7.1	45.4	3.5	25.1
	MSE	0.6	4.7	68.2	14.3	8.5
C1	GP(%)	4.0	7.0	45.3	3.7	27.3
	MSE	0.6	4.8	69.3	14.6	8.7
C2	GP(%)	4.0	6.6	45.4	3.5	26.0
	MSE	0.6	4.7	67.9	14.3	8.5
C3	GP(%)	4.0	7.3	44.9	3.5	25.1
	MSE	0.6	4.7	67.6	14.3	8.5
D1	GP(%)	4.0	7.1	45.2	3.4	21.6
	MSE	0.6	4.8	68.5	14.5	8.6
D2	GP(%)	4.0	6.7	45.1	3.5	23.2
	MSE	0.6	4.7	67.5	14.3	8.5
D3	GP(%)	4.0	7.2	45.0	3.5	24.1
	MSE	0.6	4.7	67.6	14.3	8.5

TABLE VII
SIMULATION RESULTS FOR THE CONVENTIONAL AND PROPOSED K-MEANS METHODS

		Iris	Wine	Sonar	BCW	Spam
A1	GP(%)	4.3	8.5	45.0	4.0	26.9
	MSE	2.0	14.2	135.5	28.5	17.1
A2	GP(%)	6.9	8.3	45.4	3.9	26.8
	MSE	2.1	14.2	135.0	28.3	16.9
A3	GP(%)	6.9	7.0	44.9	3.9	22.9
	MSE	2.1	14.0	134.9	28.3	17.0
B1	GP(%)	6.9	8.4	44.5	3.9	28.9
	MSE	2.1	14.2	135.5	28.3	17.1
B2	GP(%)	8.4	6.5	45.8	3.9	26.8
	MSE	2.2	14.0	135.0	28.3	16.9
B3	GP(%)	5.6	6.2	44.7	3.9	28.7
	MSE	2.0	14.0	134.9	28.3	16.9
C1	GP(%)	4.4	6.7	44.7	3.9	25.7
	MSE	2.0	14.0	135.6	28.4	17.0
C2	GP(%)	7.6	6.5	45.5	3.9	27.8
	MSE	2.1	14.0	134.9	28.3	16.9
C3	GP(%)	5.5	6.8	44.7	3.9	27.8
	MSE	2.0	14.0	134.9	28.3	16.9
D1	GP(%)	4.4	6.6	44.8	3.9	27.9
	MSE	2.0	14.0	135.6	28.4	17.1
D2	GP(%)	6.9	6.5	45.2	3.9	31.0
	MSE	2.1	14.0	134.9	28.3	18.1
D3	GP(%)	5.5	6.8	44.7	3.9	29.7
	MSE	2.0	14.0	134.9	28.3	17.0

and C3 mean the NG methods with $\lambda \rightarrow 0$ in TABLE II for $|U| = 1$, $|U| = L$ and $|U| = L/3$, respectively. Methods D1, D2 and D3 mean the proposed NG methods with $\lambda \rightarrow 0$ in TABLE III for $|U| = 1$, $|U| = L$ and $|U| = L/3$, respectively. The results of TABLE VI for the NG method hold true for the k-means method as well.

The results in TABLE IV for the k-means method and VII show that the proposed method reduces the communication costs of k-means method while maintaining accuracy compared to the conventional methods.

V. CONCLUSION

In this paper, the authors proposed a method to reduce communication costs for the NG and k-means methods, which are secure distributed processing methods, and demonstrated their effectiveness. Conventionally, the authors have distributed each element of the decomposition data and parameters to servers and updated each of them to realize a secure and usable learning method. In this case, increasing the number of elements (number of servers) to be distributed increases the security of the learning method, but increases the computational complexity of the learning method. To improve this problem, we proposed a learning method that updates only one element of each decomposed parameter. In this paper, we improve the learning method by introducing a method to reduce the communication costs to this method. With this method, the computational complexity of the NG and k-means methods with distributed processing can be almost the same as that of the conventional NG and k-means methods with a single server. This method can also be applied to the conventional NG method. Furthermore, it seems to be equally applicable to the BP method for secret distributed processing.

In the future, we plan to study similar learning methods for BP and other learning methods.

REFERENCES

[1] Cabinet Office of Japan, Society 5.0, https://www.cao.go.jp/, 2021.

[2] C. C. Aggarwal, S. P. Yu, "Privacy-Preserving Data Mining: Models and Algorithms", ISBN 978-0-387-70991-8, Springer-Verlag, 2009.

[3] D. Evans, V. Kolesnikov, M. Rosulek, "A Pragmatic Introduction to Secure Multi-Party Computation," *Foundations and Trends in Privacy and Security*, vol.2, issue.2-3, pp. 70-246, 2018.

[4] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," *STOC2009*, pp. 169-178, 2009.

[5] Q. Yang, Y. Li, Q. Chwn, Y. Tongh, "Federated Machine Learaning : Concept and Applications," *ACM Trans. Intell. Syst. Technol*, vol.10, no.2, article 12, 2019.

[6] H. Miyajima, N. Shigei, H. Miyajima, N. Shiratori, "Machine Learning with Distributed Processing using Secure Divided Data: Towards Privacy-Preserving Advanced AI Processing in a Super-Smart Society," *J. Networking and Network Applications*, vol.2, issue 1, pp.48-60, 2022.

[7] H. Miyajima, N. Shigei, H. Miyajima, N. Shiratori, "Scalability Improvement of Simplified, Secure Distributed Processing with Decomposition Data," *Special Section on Recent Progress in Nonlinear Theory and Its Applications*, vol.E14-N, no.2, Apr. 2023.

[8] M. M. Gupta, L. Jin, N. Honma, "Static and Dynamic Neural Networks," *IEEE Pres*, Wiley-Interscience, 2003.

[9] T. M. Martinetz, S. G. Berkovich, K. J. Schulten, "Neural Gas Network for Vector Quantization and its Application to Time-series Prediction," *IEEE Trans. Neural Network*, vol.4, no.4, pp.558-569, 1993.

[10] UCI Repository of Machine Learning Databases and Domain Theories, https://archive.ics.uci.edu/ml /datasets.php.