Sio-Iong Ao · Oscar Castillo ·
Hideki Katagiri · Alan Chan ·
Mahyar A. Amouzegar *Editors*

Transactions on Engineering Technologies

International MultiConference of Engineers and Computer Scientists 2021

Springer

Transactions on Engineering Technologies

Sio-Iong Ao · Oscar Castillo ·
Hideki Katagiri · Alan Chan ·
Mahyar A. Amouzegar
Editors

Transactions on Engineering Technologies

International MultiConference of Engineers and Computer Scientists 2021

Springer

Editors
Sio-Iong Ao
IAENG Secretariat
International Association of Engineers
Hong Kong, Hong Kong

Hideki Katagiri
Department of Industrial Engineering
and Management
Kanagawa University
Kanagawa, Japan

Mahyar A. Amouzegar
Department of Economics and Finance
University of New Orleans
New Orleans, LA, USA

Oscar Castillo
Computer Science in the Graduate Division
Tijuana Institute of Technology
Tijuana, Baja California, Mexico

Alan Chan
Department of Advanced Design
and Systems Engineering
City University of Hong Kong
Kowloon Tong, Hong Kong

ISBN 978-981-19-7137-2 ISBN 978-981-19-7138-9 (eBook)
https://doi.org/10.1007/978-981-19-7138-9

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

A large international conference on Advances in Engineering Technologies and Physical Science was held in Hong Kong, October 20–22, 2021, under the International MultiConference of Engineers and Computer Scientists 2021 (IMECS 2021). The IMECS 2021 is organized by the International Association of Engineers (IAENG). IAENG is a non-profit international association for the engineers and the computer scientists, which was founded originally in 1968 and has been undergoing rapid expansions in recent few years. The IMECS conference serves as a good platform for the engineering community to meet with each other and to exchange ideas. The conference has also struck a balance between theoretical and application development. The conference committees have been formed with over three hundred committee members who are mainly research center heads, faculty deans, department heads, professors, and research scientists from over 30 countries with the full committee list available at our conference web site (http://www.iaeng.org/IMECS2021/committee.html). The conference is truly an international meeting with a high level of participation from many countries. The response that we have received for the conference is excellent. There have been more than two hundred manuscript submissions for the IMECS 2021. All submitted papers have gone through the peer-review process, and the overall acceptance rate is 51%.

This volume contains ten revised and extended research articles written by prominent researchers participating in the conference. Topics covered include industrial engineering, electrical engineering, engineering mathematics, and industrial applications. The book offers the state of the art of tremendous advances in engineering technologies and physical science and applications, and also serves as an excellent reference work for researchers and graduate students working with/on engineering technologies and physical science and applications.

Sio-Iong Ao
Oscar Castillo
Hideki Katagiri
Alan Hoi-shou Chan
Mahyar A. Amouzegar

Contents

Secure Distributed Processing of BP with Updatable Decomposition Data

Hirofumi Miyajima[1(✉)], Noritaka Shigei[2], Hiromi Miyajima[2], and Norio Shiratori[3]

[1] Nagasaki University, 1-14 Bunkyomachi, Nagasaki City, Nagasaki, Japan
miyajima@nagasaki-u.ac.jp
[2] Kagoshima University, 1-21-24, Korimoto, Kagoshima, Japan
shigei@ibe.kagoshima-u.ac.jp, k2356323@kadai.jp
[3] Chuo University, 1-13-27, Kasuga, Bunkyoku, Tokyo, Japan
norio.shiratori.e8@tohoku.ac.jp

Abstract. Many studies have been conducted to perform machine learning while maintaining data confidentiality. For example, many studies have been conducted in this field on 1) secure multiparty computation (SMC), 2) quasi-homomorphic encryption, and 3) federated learning (FL), and so on. Methods 1) and 2) are extremely confidential in terms of security. However, their utilization of machine learning is limited. Method 3) is highly utilizable for many machine learning problems owing to the simplicity of the procedure. However, the security level is low compared to Methods 1) and 2). Previous studies have shown that both security and utility are essential for machine learning using confidential data. Therefore, it is desirable to develop a method that strikes a balance between confidentiality and utility. For this reason, in the previous paper, a secure distributed processing of BP with divided data has been proposed. However, this method has room for further improvement in that it fixes the decomposed data during learning. In this chapter, to improve the confidentiality of data, we generalize this method and present a secure distributed processing of BP with updatable decomposition data during learning. The effectiveness of the proposed method is shown in numerical simulations.

Keywords: Federated learning · Back propagation method · Updatable decomposition data · Secure distributed processing · Data confidentiality

1 Introduction

It is desirable to build a super-smart society in which big data is processed using AI to automatically retrieve advanced knowledge. To realize this society, the safety and security of data held by users must be guaranteed [1–3]. Hence, many studies haves been conducted on machine learning while maintaining data confidentiality. For example, many studies have been conducted in this field on

S. Ao et al. (Eds.): IMECS 2021, *Transactions on Engineering Technologies*, pp. 1–15, 2023.
https://doi.org/10.1007/978-981-19-7138-9_1

1) secure multiparty computation (SMC) [4], 2) quasi-homomorphic encryption [5], and 3) federated learning (FL) [6], and so on [7–9]. In these cases, Methods 1) and 2) strictly preserve privacy by using data encryption and random numbers, and Method 3) partitions all data into subsets and distributes them to each server to distribute the data. This method executes learning by distributed processing without sending the data from each server. Each approach has advantages and disadvantages. Methods 1) and 2) are extremely confidential in terms of security. However, their utilization of machine learning is limited. Method 3) is highly utilizable for many machine learning problems owing to the simplicity of the procedure. However, the security level is low compared to Methods 1) and 2). Previous studies have shown that both security and utility are essential for machine learning using confidential data. Therefore, it is desirable to develop a method that strikes a balance between confidentiality and utility. In the previous paper, we proposed the method of secure distributed processing of BP with divided data [10,11]. The method randomly divides each of the learning data and parameters into multiple pieces and uses these divided data and parameters as learning data and parameters. However, this method fixes the division of data during learning. From the standpoint of data confidentiality, it is desirable that the division of data is updated during learning. This chapter proposes a secure distributed processing method with updatable decomposition data, "divided data" updated during learning. Specifically, a secure distributed processing of BP with updatable decomposition data is proposed. The accuracy of the proposed method is compared with the conventional methods in numerical simulations. The remainder of this chapter is organized as follows.

In Sect. 2, we define the divided data in the additive and product forms as secure divided data for the BP methods. In Sect. 3, we propose learning methods based on distributed processing using updatable decomposition data independently on each server. In Sect. 4, we compare the accuracy of the conventional BP methods with those of the proposed BP methods by numerical experiments. Finally, in Sect. 5, we summarize the contributions of this study and discuss future prospects.

2 Preliminaries

2.1 Secure Computation and a Configuration Used for the Proposed Method

We explain the system used in the proposed method. In this subsection, we show the system with $Q+1$ servers in Fig. 1. We denote x and $f(x)$ by a scalar data and target function (value). We divide each data x into multiple pieces and each of them is sent to a server.

First, we divide any data x into Q pieces randomly as $x = \sum_{q=1}^{Q} x^{(q)}$. Each piece $x^{(q)}$ is sent to Server q. We calculate each function $f_q(x^{(q)})$ in Server q and it is sent to Server 0, where $f_q(\cdot)$ means a function in Server q. In Server 0, we aggregate these results and calculate $h(x) = \odot_{q=1}^{Q} f_q(x^{(q)})$, where $\odot$ means an

integrated function. If $h(x)$ is sufficiently near to $f(x)$, then we terminate the process else the same process is repeated with updated $f_q(\cdot)$.

The problem is how to determine the calculation process $f_q(\cdot)$.

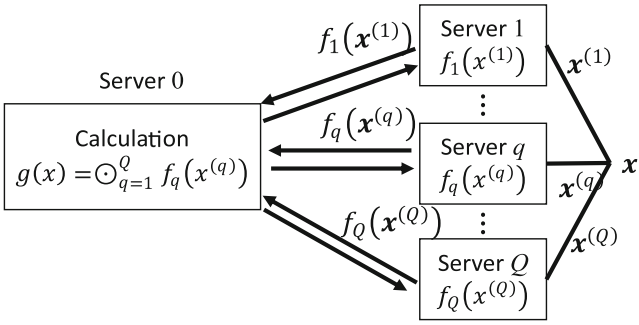

Fig. 1. An example of secure distributed processing method: the data x is divided into Q pieces. Each piece is sent to a server. Each server calculates $f_q(x^{(q)})$ and sends it to Server 0. Server 0 obtains the result $g(x)$ by integrating the partial calculations $f_q(x^{(q)})$. If $g(x)$ is sufficiently near to $f(x)$, the process terminates else the same process is repeated with updated $f_q(\cdot)$.

2.2 Decomposition Data for the Proposed Method

In this subsection, we explain how to use the divided data for the proposed method [8]. Let a and b be two real numbers. First, we divide two integers a and b into Q real numbers. We denote $a = \sum_{q=1}^{Q} a^{(q)}$ and $b = \sum_{q=1}^{Q} b^{(q)}$ as additive form and $a = \Pi_{q=1}^{Q} A^{(q)}$ and $b = \Pi_{q=1}^{Q} B^{(q)}$ as the product form. Then we have the following results.

1) $a + b = \sum_{q=1}^{Q}(a^{(q)} + b^{(q)})$, 2) $a - b = \sum_{q=1}^{Q}(a^{(q)} - b^{(q)})$

3) $a \times b = \Pi_{q=1}^{Q}(A^{(q)}B^{(q)})$, 4) $a/b = \Pi_{q=1}^{Q}(A^{(q)}/B^{(q)})$

It means that four basic arithmetic operations hold for integrating results computed independently on each server. In this case, every server cannot know the original data a and b. We explain how to divide and compute the data using an example.

Example 1. Let $a = 5$, $b = 6$ and $Q = 3$. We divide a and b randomly as $5 = 4 + (-1) + 2 = (-2) \times (-1) \times 2.5$ and $6 = 3 + (-2) + 5 = (-3) \times 2 \times (-1)$. In this case, we have $a^{(1)} = 4$, $a^{(2)} = -1$, $a^{(3)} = 2$, $A^{(1)} = -2$, $A^{(2)} = -1$, $A^{(3)} = 2.5$, $b^{(1)} = 3$, $b^{(2)} = -2$, $b^{(3)} = 5$, $B^{(1)} = -3$, $B^{(2)} = 2$ and $B^{(3)} = -1$, respectively. We can calculate $a + b$ as follows (See Table 1(a)).

$$a + b = (a^{(1)} + b^{(1)}) + (a^{(2)} + b^{(2)}) + (a^{(3)} + b^{(3)}) = 11$$

Likewise, we can calculate $a \times b$ as follows (See Table 1(b)).

$$a \times b = (A^{(1)} \times B^{(1)})(A^{(2)} \times B^{(2)})(A^{(3)} \times B^{(3)}) = 30 \qquad \square$$

Table 1. An example of decomposition data for the proposed method

(a) An example of decomposition data in additive form

	Data a	Data b	Addition
Server 1	$a^{(1)} = 4$	$b^{(1)} = 3$	7
Server 2	$a^{(2)} = -1$	$b^{(2)} = -2$	-3
Server 3	$a^{(3)} = 2$	$b^{(3)} = 5$	7
Addition	5	6	11

(b) An example of decomposition data in product form

	Data a	Data b	Multiplication
Server 1	$A^{(1)} = -2$	$B^{(1)} = -3$	6
Server 2	$A^{(2)} = -1$	$B^{(2)} = 2$	-2
Server 3	$A^{(3)} = 2.5$	$B^{(3)} = -1$	-2.5
Multiplication	5	6	30

2.3 Neural Network and BP Method

In this subsection, we explain three layered Neural Network (NN) and BP method without loss of generality [12]. Figure 2 shows an example of three layered NN. For any positive integer i, let $Z_i = \{1, 2, \cdots, i\}$ and $Z_i^* = \{0, 1, \cdots, i\}$. We define an output function $\boldsymbol{h} : J_{in}^n \to J_{out}^R$ for each $\boldsymbol{x} \in J_{in}^n$ as follows. $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), \cdots, h_s(\boldsymbol{x}), \cdots, h_R(\boldsymbol{x}))$, where $J_{in} = [0, 1]$ or $[-1, 1]$, and $J_{out} = [0, 1]$, $[-1, 1]$ or $\{0, 1\}$. In this case, we determine weights of NN by using the set of learning data $X = \{(\boldsymbol{x}^l, \boldsymbol{d}(\boldsymbol{x}^l)) | \boldsymbol{x}^l \in J_{in}^n, \boldsymbol{d}(\boldsymbol{x}^l) \in J_{out}^R, l \in Z_L\}$ as follows, where $\boldsymbol{d}(\boldsymbol{x}^l) = \big(d_1(\boldsymbol{x}^l), \ldots, d_s(\boldsymbol{x}^l), \cdots, d_R(\boldsymbol{x}^l)\big)$ denotes the desired output for the input data $\boldsymbol{x}^l$.

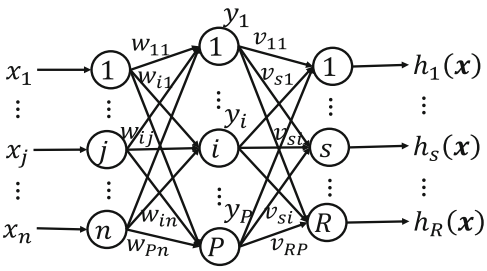

Fig. 2. An example of three layered Neural Network

We denote two sets $\{w_{ij} | i \in Z_P, j \in Z_n^*\}$ and $\{v_{si} | s \in Z_R, i \in Z_P^*\}$ by W and V, respectively. In this case, we calculate an output of NN as follows.

$$y_i(\boldsymbol{x}) = \frac{1}{1 + \exp\left(-\left(\sum_{j=0}^{n} w_{ij} x_j\right)\right))} \tag{1}$$

$$h_s(\boldsymbol{x}) = \frac{1}{1 + \exp\left(-\left(\sum_{i=0}^{P} v_{si} y_i(\boldsymbol{x})\right)\right)} \tag{2}$$

where $\boldsymbol{x} = (x_1, \cdots, x_j, \cdots, x_n)$ $(i \in Z_P,\ s \in Z_R)$ and w_{i0} and v_{s0} are thresholds, $x_0 = 1$ and $y_0 = 1$.

Then, we define the evaluation function as follows.

$$E(X, W, V) = \frac{1}{2L} \sum_{l=1}^{L} \sum_{s=1}^{R} \left(d_s(\boldsymbol{x}^l) - h_s(\boldsymbol{x}^l) \right)^2 \tag{3}$$

We update each weight of W and V using the update amounts of Eqs. (4) and (5).

$$\triangle w_{ij} = \alpha \sum_{s=1}^{S} (d_s(\boldsymbol{x}^l) - h_s(\boldsymbol{x}^l))(1 - h_s(\boldsymbol{x}^l)) v_{si} y_i(\boldsymbol{x}^l)(1 - y_i(\boldsymbol{x}^l)) x_j^l \tag{4}$$

$$\triangle v_{si} = \alpha (d_s(\boldsymbol{x}^l) - h_s(\boldsymbol{x}^l)) h_s(\boldsymbol{x}^l)(1 - h_s(\boldsymbol{x}^l)) y_i(\boldsymbol{x}^l) \tag{5}$$

Then, we show the flowchart of BP algorithm as Fig. 3. In this case, we denote the set of learning data, the maximum number of learning time, threshold, and learning rate by X, t_{max}, θ, and α, respectively.

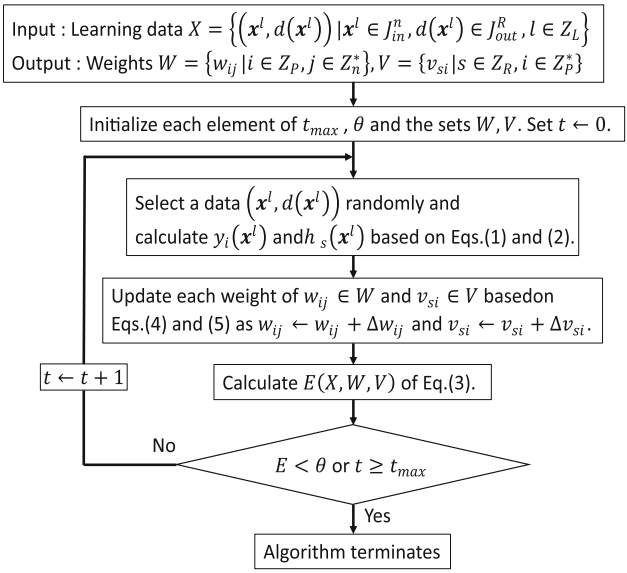

Fig. 3. The flowchart for BP method [12]

Likewise, batch and mini-batch types of BP are defined [12].

3 BP Method for Secure Distributed Processing with Updatable Decomposition Data

In this section, we propose a BP method for secure distributed processing with updatable decomposition data. The features of the proposed method are as follows.