セキュアマルチパーティ計算を用いたエッジシステムに対する機械学習法の開発

研究代表者	宮島　洋文	長崎大学　情報データ科学部　准教授
共同研究者	重井　徳貴	鹿児島大学大学院理工学研究科 工学専攻　教授
共同研究者	宮島　廣美	鹿児島大学　名誉教授

1 研究の背景と目的

情報化社会を支える基盤技術の一つであるクラウドコンピューティングの利用が拡大している．一方，クラウドに接続される利用者の端末や機器の数が増加すると，その能力の低下が低下する［1]．この問題を改善するためのアーキテクチャの一つとして，エッジコンピューティングシステムが提案されている．これは，従来システムで遠くに置かれているメインサーバの情報処理を，ユーザに近い場所(エッジ)に置かれたローカルサーバに肩代りさせて，情報のやりとりをショートカットするモデルである．一方で，クラウドシステムにおいては外部サーバにてデータを扱うことから，ユーザが情報漏洩等のリスクを懸念する場合がある［1-3]．本研究では，このようなエッジシステムにおいて，今日の人工知能の基盤技術の一つである機械学習について，データの安全性を考慮した手法についての研究を行う．ここで，サーバ上のデータに対する安全性を高める手法として，SMC（Secure Multiparty Computation）［4-7]，準同型暗号［8]，連合学習［9-11］などが提案されている．しかしながら，通常の機械学習手法は，これらのデータの安全性を高める手法を適用することができるとは限らない．そのため，これらのデータの安全性を高める手法に合わせた機械学習のアルゴリズムについての研究が進められてきた［12-17]．

本研究では，エッジシステムにおいて，SMC と組み合わせることでデータの安全性を考慮した機械学習のアルゴリズムについての研究を行った．研究成果としては，いくつかの基本的な機械学習手法について，SMC と組み合わせたアルゴリズムを提案し，また，数値実験によりそれらの提案手法の有効性を検証した．

2 準備

ここでは，本研究で扱った SMC による分散処理計算の概要について説明する．また，本研究で扱った機械学習手法である Back Propagation (BP)法，および Neural Gas (NG) 法の概要について説明する．

2-1 Secure Multiparty Computation の概要

SMC においては，秘匿分解データを使った分散処理計算によりデータの安全性を高める[4-7]．ここで，本研究において用いる秘匿分解データを使った分散処理計算の概略について説明する．ここでは，図 1 に示す$Q+1$個のサーバ(サーバ 0 とQ個のエッジ)からなるシステムを例として，任意の実数データxに対する任意の関数$f(x)$の計算について説明する．

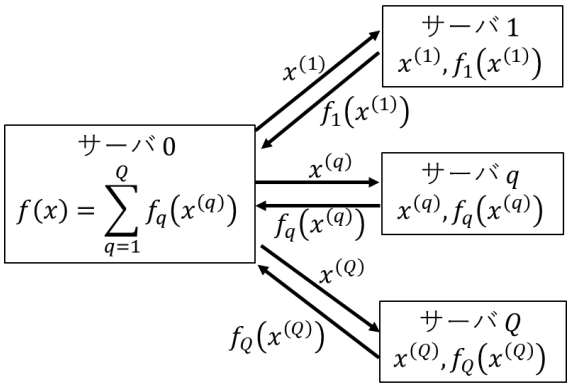


図 1　本研究における分散処理計算の概略

はじめに，実数データxについて，$\sum_{q=1}^{Q} x^{(q)} = x$ を満たすQ個の実数値$x^{(1)}, \cdots, x^{(Q)}$をランダムに選択し，サーバ 0 を除く各サーバに送る．サーバq $(q = 1, \cdots, Q)$では$x^{(q)}$のみが格納されているとき，他の実数値$x^{(1)}, \cdots, x^{(q-1)}, x^{(q+1)}, \cdots, x^{(Q)}$はサーバ$q$内部には格納されていないため，サーバ内部$q$の情報のみでは実数データ$x$を復号化することができない．

次に，関数$f(x)$の計算結果を求める．ここで，関数$f(x)$に対して$\sum_{q=1}^{Q} f_q\left(x^{(q)}\right) = f(x)$ を満たすQ個の関数$f_1\left(x^{(1)}\right), \cdots, f_Q\left(x^{(Q)}\right)$を定義する．サーバ$q$ $(q = 1, \cdots, Q)$内部では $f_q\left(x^{(q)}\right)$ を計算し，結果をサーバ 0 に送る．サーバ 0 では，他のサーバから送られた計算結果を統合し，計算結果$f(x) = \sum_{q=1}^{Q} f_q\left(x^{(q)}\right)$を得る．このとき，計算の途中経過において元のデータxは復号化されておらず，また，各サーバではデータxの断片的な情報のみが格納されているため，データxを復号化することができない．

問題は，関数$f(x)$に対して適切な計算処理$f_1\left(x^{(1)}\right), \cdots, f_Q\left(x^{(Q)}\right)$をどのように定義するかである．本研究では，いくつかの基礎的な機械学習手法について，このような分散処理により実現する．この処理過程においては，各サーバはデータの断片的な情報以外は知ることができず，データの秘匿性が保持される．

2-2 階層型ニューラルネットワークと BP 法の概要

ここでは，図 2 に示す 3 層の階層型ニューラルネットワークに対する出力の計算，および機械学習手法の一つである BP 学習について説明する［18］．ここで，以降では，任意の自然数i に対して $Z_i = \{1,2, \cdots, i\}$, $Z_i^* = \{0,1, \cdots, i\}$ とおく．

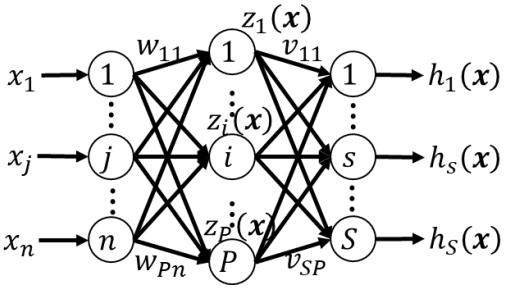


図 2　3 層の階層型ニューラルネットワークの概略

図 2 は，入力要素数n，出力要素数S，中間層ニューロ数Pである 3 層の階層型ニューラルネットワークの例である．$\{w_{ij} | i \in Z_P, j \in Z_n^*\}$ と $\{v_{si} | s \in Z_S, i \in Z_P^*\}$ をそれぞれ中間層と出力層の重みとする．入力 $\boldsymbol{x} = (x_1, \cdots, x_n)$ が与えられたとき，ニューラルネットワークの出力$h_1(\boldsymbol{x}), \cdots, h_S(\boldsymbol{x})$は次式により計算される．

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

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

ここで，$x_0 = 1$，$z_0(\boldsymbol{x}) = 1$ とする．

重み$\{w_{ij} | i \in Z_P, j \in Z_n^*\}$，$\{v_{si} | s \in Z_S, i \in Z_P^*\}$ の決定方法として，BP 法が広く用いられている．学習用データ$X = \{\boldsymbol{x}^l, \boldsymbol{d}(\boldsymbol{x}^l) | l \in Z_L\}$が与えられたとする．ここで，$\boldsymbol{d}(\boldsymbol{x}^l) = \left(d_1(\boldsymbol{x}^l), \cdots, d_S(\boldsymbol{x}^l)\right)$は入力$\boldsymbol{x}^l = (x_1^l, \cdots, x_n^l)$に対する理想的なニューラルネットワークの出力とする．このとき，学習における評価関数が次式により与えられる．

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

式(3)に対する最小化問題を解くことにより，重み$\{w_{ij}|i \in Z_P, j \in Z_n^*\}$，$\{v_{si}|s \in Z_S, i \in Z_P^*\}$ を適切な値に決定することができる．そのような手法の一つとして，BP 法が提案されている．BP 法においては，式(3)に示す評価関数を重み w_{ij} $(i \in Z_P, j \in Z_n^*)$，v_{si} $(s \in Z_S, i \in Z_P^*)$ で 1 階微分することにより，以下のような重みの更新式を得る［18］.

$$w_{ij} \leftarrow w_{ij} + K_w \sum_{s=1}^{S} \big(d_s(\boldsymbol{x}^l) - h_s(\boldsymbol{x}^l)\big) h_s(\boldsymbol{x}^l)\big(1 - h_s(\boldsymbol{x}^l)\big) v_{si}\, z_i(\boldsymbol{x}^l)\big(1 - z_i(\boldsymbol{x}^l)\big) x_j^l \tag{4}$$

$$v_{si} \leftarrow v_{si} + K_v \big(d_s(\boldsymbol{x}^l) - h_s(\boldsymbol{x}^l)\big) h_s(\boldsymbol{x}^l)\big(1 - h_s(\boldsymbol{x}^l)\big) z_i(\boldsymbol{x}^l) \tag{5}$$

ここで，K_w，K_vは学習係数を表す.

BP 法の概略を以下に示す.

［BP 法の概略］

学習用データ ： $X = \{\boldsymbol{x}^l, \boldsymbol{d}(\boldsymbol{x}^l) | l \in Z_L\}$

重み ： $\{w_{ij}|i \in Z_P, j \in Z_n^*\}$，$\{v_{si}|s \in Z_S, i \in Z_P^*\}$

最大学習回数 ： T，しきい値 ： θ

Step 1

重み$\{w_{ij}|i \in Z_P, j \in Z_n^*\}$，$\{v_{si}|s \in Z_S, i \in Z_P^*\}$ の各値をランダムに設定する．$t \leftarrow 0$とする.

Step 2

学習用データの中からデータを選択する．（データ$\boldsymbol{x}^l, \boldsymbol{d}(\boldsymbol{x}^l)$ $(l \in Z_L)$が選択されたとする）

Step 3

式(4)，(5)に従い，各重みを更新する.

Step 4

式(3)に従い，学習用データに対する平均二乗誤差Eを計算する．$E < \theta$または$t = T$ならば学習終了，それ以外の場合は$t \leftarrow t + 1$として Step 2 へ.

BP 法を実行することで，与えられた学習用データ$X = \{\boldsymbol{x}^l, \boldsymbol{d}(\boldsymbol{x}^l) | l \in Z_L\}$ の入出力関係を表すニューラルネットワークの重み$\{w_{ij}|i \in Z_P, j \in Z_n^*\}$，$\{v_{si}|s \in Z_S, i \in Z_P^*\}$ を得ることができる.

2-3 NG 法の概要

ここでは，機械学習手法の一つである NG 法について説明する［19］. NG 法が実現するベクトル量子化は，大量のデータを少数のデータで近似するものである． ここでは， データ$X = \{\boldsymbol{x}^l | l \in Z_L\}$を有限個の参照ベクトルの集合$W = \{\boldsymbol{w}_i | i \in Z_r\}$で近似する場合について説明する．$e_i(\boldsymbol{x}) \in Z_{r-1}^*$をデータ$\boldsymbol{x}$に対する近傍ランクとする． つまり， $\boldsymbol{w}_i$はデータ$\boldsymbol{x}$に対して$e_i(\boldsymbol{x}) + 1$番目に近い参照ベクトルである． 任意のデータ$\boldsymbol{x}^l = (x_1^l, \cdots, x_n^l)$ $(l \in Z_L)$と任意の参照ベクトル$\boldsymbol{w}_i = (w_{i1}, \cdots, w_{in})$ $(i \in Z_r)$ の近さをユークリッド距離$\|\boldsymbol{x}^l - \boldsymbol{w}_i\|$ で定義するとき， データXを参照ベクトルの集合Wで近似する度合いの評価関数として， 以下のように与えることができる.

$$E = \frac{1}{L} \sum_{l=1}^{L} \sum_{i=1}^{r} \frac{\exp(-e_i(\boldsymbol{x}^l)/\lambda)}{\sum_{I=1}^{r} \exp(-e_I(\boldsymbol{x}^l)/\lambda)} \sum_{j=1}^{n} \big(x_j^l - w_{ij}\big)^2 \tag{6}$$

ここで，λは定数値である.

式(6)に対する最小化問題を解くことにより， 参照ベクトル$\{w_{ij}|i \in Z_r, j \in Z_r\}$を適切な値に決定することができる．そのような手法の一つとして，NG 法が提案されている．NG 法においては，式(6)に示す評価関数に対する 1 階微分を用いることで，以下のような参照ベクトルの更新式を得る［19］.

$$w_{ij} \leftarrow w_{ij} + \varepsilon \cdot \exp\left(-\frac{e_i(\boldsymbol{x}^l)}{\lambda}\right) \cdot \big(x_j^l - w_{ij}\big) \tag{7}$$

ここで，εは学習係数を表す．

NG 法の概略を以下に示す．

[NG 法の概略]

学習用データ ： $X = \{\boldsymbol{x}^l | l \in Z_L\}$

参照ベクトル ： $\{w_{ij} | i \in Z_r, j \in Z_r\}$

最大学習回数 ： T

Step 1

参照ベクトル$\{w_{ij} | i \in Z_r, j \in Z_r\}$ の各値をランダムに設定する．$t \leftarrow 0$とする．

Step 2

学習用データの中からデータを選択する．（データ$\boldsymbol{x}^l$ $(l \in Z_L)$が選択されたとする）

Step 3

式(7)に従い，各参照ベクトルを更新する．

Step 4

$t = T$ならば学習終了，それ以外の場合は$t \leftarrow t + 1$として Step 2 へ．

NG 法を実行することで，与えられた学習用データ$X = \{\boldsymbol{x}^l | l \in Z_L\}$を近似する参照ベクトルの集合$W = \{\boldsymbol{w}_i | i \in Z_r\}$ を得ることができる．また，$\lambda \to 0$とすると，NG 法は k-means 法と一致する．

3 提案手法

ここでは，本研究で扱った，SMC による秘匿分散処理計算に基づく BP 法および NG 法の概要について説明する．

3-1 秘匿分散処理計算に基づく BP 法と NG 法

BP 法は，教師あり学習手法として広く知られている．また，NG 法は，教師なし学習手法として広く知られている．ここでは，SMC と組み合わせることでデータの安全性を高めた BP 法および NG 法の提案手法について述べる．

3-1-1 秘匿分散処理に基づく BP 法の提案手法

SMC においては，秘匿分解データを使った秘匿分散処理計算によりデータの安全性を高める．ここでは，BP 法において与えられた学習用データ$X = \{\boldsymbol{x}^l, \boldsymbol{d}(\boldsymbol{x}^l) | l \in Z_L\}$を以下のように$Q$個に分解し，各サーバに分けて保存する．

$$x_j^l = \prod_{q=1}^{Q} x_j^{l(q)} \ \ (l \in Z_L, j \in Z_n) \tag{8}$$

$$d_s(\boldsymbol{x}^l) = \sum_{q=1}^{Q} d_s^{(q)}(\boldsymbol{x}^l) \ \ (l \in Z_L, s \in Z_S) \tag{9}$$

また，重み$\{w_{ij} | i \in Z_P, j \in Z_n^*\}$，$\{v_{si} | s \in Z_S, i \in Z_P^*\}$は以下のように$Q$個に分解し，各サーバに分けて保存するものとする．

$$w_{ij} = \prod_{q=1}^{Q} w_{ij}^{(q)} \ \ (i \in Z_P, j \in Z_n^*) \tag{10}$$

$$v_{si} = \prod_{q=1}^{Q} v_{si}^{(q)} \ \ (s \in Z_S, i \in Z_P^*) \tag{11}$$

以降では，式(8)-(11)により分解されたデータおよび重みを用いた BP 法の提案手法について述べる．

はじめに，データを復号化することなくニューラルネットワークの出力$h_1(\boldsymbol{x}^l),\cdots,h_S(\boldsymbol{x}^l)$を求める方法について議論する．式(8)−(11)を用いることで，入力$\boldsymbol{x}^l\ (l\in Z_L)$に対する式(1)の計算は以下のように書き換えることができる．

$$z_i(\boldsymbol{x}^l)=\frac{1}{1+\exp\left(-\sum_{j=0}^{n}\prod_{q=1}^{Q}w_{ij}^{(q)}\,x_j^{l(q)}\right)}\ (i\in Z_P) \tag{12}$$

ここで，式(12)の計算結果$z_i(\boldsymbol{x}^l)$を以下のようにQ個に分解し，各サーバに分けて保存する．

$$z_i(\boldsymbol{x}^l)=\prod_{q=1}^{Q}z_i^{(q)}(\boldsymbol{x}^l)\ (s\in Z_S,i\in Z_P^*) \tag{13}$$

このとき，入力$\boldsymbol{x}^l\ (l\in Z_L)$に対する式(2)の計算は以下のように書き換えることができる．

$$h_s(\boldsymbol{x}^l)=\frac{1}{1+\exp\left(-\sum_{i=0}^{P}\prod_{q=1}^{Q}v_{si}^{(q)}\,z_i^{(q)}(\boldsymbol{x}^l)\right)}\ (s\in Z_S) \tag{14}$$

式(12)−(14)において，入力データ$\boldsymbol{x}^l\ (l\in Z_L)$は復号化されていない．そのため，式(12)−(14)を用いることで，入力データ$\boldsymbol{x}^l\ (l\in Z_L)$を復号化することなく，ニューラルネットワークの出力$h_1(\boldsymbol{x}^l),\cdots,h_S(\boldsymbol{x}^l)$を求めることができる．

また，式(14)の計算結果$h_s(\boldsymbol{x}^l)$を以下のようにQ個に分解し，各サーバに分けて保存する．

$$h_s(\boldsymbol{x}^l)=\prod_{q=1}^{Q}h_s^{(q)}(\boldsymbol{x}^l)\ (s\in Z_S) \tag{15}$$

このとき，式(3)に示す評価関数の値は，次式により求めることができる．

$$E=\frac{1}{L}\sum_{l=1}^{L}\sum_{s=1}^{S}\left\{\sum_{q=1}^{Q}\left(d_s^{(q)}(\boldsymbol{x}^l)-h_s^{(q)}(\boldsymbol{x}^l)\right)\right\}^2 \tag{16}$$

通常のBP法においては，式(3)に示す評価関数を重み $w_{ij}\ (i\in Z_P,j\in Z_n^*)$，$v_{si}\ (s\in Z_S,i\in Z_P^*)$ で1階微分することにより，重みの更新式を得る［5］．そこで，式(16)に示す評価関数を $w_{ij}^{(q)}\ (i\in Z_P,j\in Z_n^*,\ q\in Z_Q)$，$v_{si}^{(q)}\ (s\in Z_S,i\in Z_P^*,q\in Z_Q)$ で1階微分することにより，以下のような$w_{ij}^{(q)}\ (i\in Z_P,j\in Z_n^*,\ q\in Z_Q)$，$v_{si}^{(q)}\ (s\in Z_S,i\in Z_P^*,q\in Z_Q)$ に対する更新式を得る．

$$\begin{aligned}w_{ij}^{(q)}&\leftarrow w_{ij}^{(q)}+K_w\sum_{s=1}^{S}\left\{\sum_{q=1}^{Q}\left(d_s^{(q)}(\boldsymbol{x}^l)-h_s^{(q)}(\boldsymbol{x}^l)\right)\right\}h_s(\boldsymbol{x}^l)\left(1-h_s(\boldsymbol{x}^l)\right)\left\{\prod_{q=1}^{Q}v_{si}^{(q)}z_i^{(q)}(\boldsymbol{x}^l)\right\}\left(1\right.\\&\qquad\left.-z_i(\boldsymbol{x}^l)\right)x_j^l\\&=w_{ij}\left(1+K_w\sum_{s=1}^{S}\left\{\sum_{q=1}^{Q}\left(d_s^{(q)}(\boldsymbol{x}^l)-h_s^{(q)}(\boldsymbol{x}^l)\right)\right\}h_s(\boldsymbol{x}^l)\left(1-h_s(\boldsymbol{x}^l)\right)\left\{\prod_{q=1}^{Q}v_{si}^{(q)}z_i^{(q)}(\boldsymbol{x}^l)\right\}\left(1\right.\right.\\&\qquad\left.\left.-z_i(\boldsymbol{x}^l)\right)\left\{\prod_{q=1}^{Q}w_{ij}^{(q)}x_j^{l(q)}\right\}\right)\cdot\frac{1}{w_{ij}^{(q)}}\end{aligned} \tag{17}$$

$$\begin{aligned}v_{si}^{(q)}&\leftarrow v_{si}^{(q)}+K_v\left\{\sum_{q=1}^{Q}\left(d_s^{(q)}(\boldsymbol{x}^l)-h_s^{(q)}(\boldsymbol{x}^l)\right)\right\}h_s(\boldsymbol{x}^l)\left(1-h_s(\boldsymbol{x}^l)\right)z_i(\boldsymbol{x}^l)\\&=v_{si}\left(1+K_v\left\{\sum_{q=1}^{Q}\alpha\left(d_s^{(q)}(\boldsymbol{x}^l)-h_s^{(q)}(\boldsymbol{x}^l)\right)\right\}h_s(\boldsymbol{x}^l)\left(1-h_s(\boldsymbol{x}^l)\right)\left\{\prod_{q=1}^{Q}v_{si}^{(q)}z_i^{(q)}(\boldsymbol{x}^l)\right\}\right)\cdot\frac{1}{v_{si}^{(q)}}\end{aligned} \tag{18}$$

ここで，K_w，K_vは学習係数を表す．

式(17)，(18)を用いることで，分解された重み $w_{ij}^{(q)}\ (i \in Z_P, j \in Z_n^*,\ q \in Z_Q)$，$v_{si}^{(q)}\ (s \in Z_S, i \in Z_P^*, q \in Z_Q)$ に対する BP 法を実現することができる．このとき，BP 法で必要な情報は式(12)−(18)により求めることができ，データおよび重みを復号化することなく計算することができる．

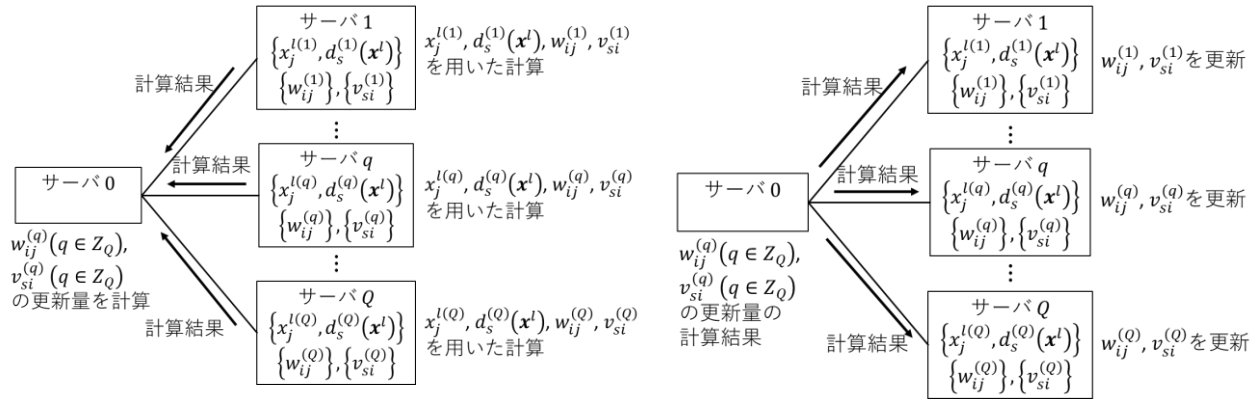


図 2 秘匿分散処理に基づく BP 法の提案手法の概略

図 2 に，秘匿分散処理に基づく BP 法の提案手法の概略を示す．提案手法においては，各サーバ内部において分解されたデータおよび重みを用いた計算を行い，これらの計算結果をサーバ 0 に送る．サーバ 0 においては，各サーバから送られてきた計算結果を統合することで，式(17)，(18)中の重みの更新量を計算する(図 2 (a)を参照)．また，サーバ 0 で計算された重みの更新量を各サーバに送り，各サーバ内部において，分割された重みの更新を行う (図 2 (b)を参照)．

また，BP 法よりも能力の高い機械学習手法として，Particle Swarm Optimization (PSO)を組み合わせた機械学習手法がある [20]．本研究では，PSO と組み合わせた機械学習手法について，秘匿分散処理に基づくアルゴリズムの提案を行った．この提案手法においては，図 2 に示す BP 法と同様に，サーバ 0 で PSO による重みの更新量を計算し，重みの更新量の計算結果を各サーバに送り，各サーバ内部で重みを更新する．

3-1-2 秘匿分散処理に基づく NG 法の提案手法

ここでは，NG 法において与えられた学習用データ$X = \{\boldsymbol{x}^l | l \in Z_L\}$を以下のように$Q$個に分解し，各サーバに分けて保存する．

$$x_j^l = \sum_{q=1}^{Q} x_j^{l(q)}\ (l \in Z_L, j \in Z_n) \tag{19}$$

また，参照ベクトル$\{w_{ij} | i \in Z_r, j \in Z_n\}$は以下のように$Q$個に分解し，各サーバに分けて保存するものとする．

$$w_{ij} = \sum_{q=1}^{Q} w_{ij}^{(q)}\ (i \in Z_r, j \in Z_n) \tag{20}$$

データ$\boldsymbol{x}^l = (x_1^l, \cdots, x_n^l)\ (l \in Z_L)$と参照ベクトル$\boldsymbol{w}_i = (w_{i1}, \cdots, w_{in})\ (i \in Z_r)$の距離$\|\boldsymbol{x}^l - \boldsymbol{w}_i\|$は，式(19)，(20)により分解されたデータと参照ベクトルを用いると次式により求めることができる．

$$\|\boldsymbol{x}^l - \boldsymbol{w}_i\|^2 = \sum_{j=1}^{n} \left(\sum_{q=1}^{Q} \left(x_j^{l(q)} - w_{ij}^{(q)} \right) \right)^2\ (l \in Z_L, i \in Z_r) \tag{21}$$

$\|\boldsymbol{x}^l - \boldsymbol{w}_i\|^2$距離をもとに，データ$\boldsymbol{x}^l\ (l \in Z_L)$に対する参照ベクトル$\boldsymbol{w}_i (i \in Z_r)$の近傍ランク$e_i(\boldsymbol{x}^l)$を求める．このとき，式(6)に示す NG 法の評価関数の値は，次式により求めることができる．

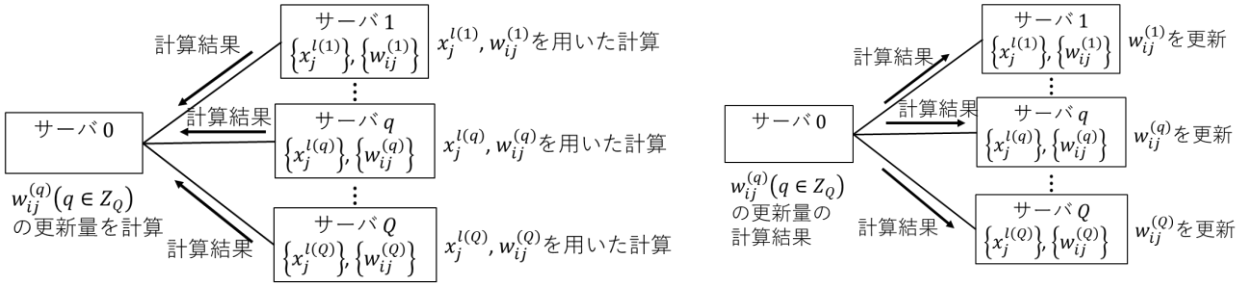
$$E = \frac{1}{L} \sum_{l=1}^{L} \sum_{i=1}^{r} \frac{\exp(-e_i(\boldsymbol{x}^l)/\lambda)}{\sum_{l=1}^{r} \exp(-e_l(\boldsymbol{x}^l)/\lambda)} \sum_{j=1}^{n} \left(\sum_{q=1}^{Q} \left(x_j^{l(q)} - w_{ij}^{(q)} \right) \right)^2 \tag{22}$$

通常の NG 法においては，式(6)に示す評価関数を参照ベクトル $w_{ij}\,(i \in Z_r, j \in Z_n)$ で 1 階微分することにより，式(7)に示す参照ベクトルの更新式を得る［6］．そこで，式(22)に示す評価関数を $w_{ij}\,(i \in Z_r, j \in Z_n)$ で 1 階微分することにより，以下のような $w_{ij}^{(q)}\,(i \in Z_r, j \in Z_n, q \in Z_Q)$ に対する更新式を得る．

$$w_{ij}^{(q)} \leftarrow w_{ij}^{(q)} + \varepsilon \cdot \exp\left(-\frac{e_i(\boldsymbol{x}^l)}{\lambda}\right) \cdot \left(x_j^l - w_{ij}\right) \tag{23}$$

ここで，εは学習係数を表す．

式(23)を用いることで，分解された参照ベクトル $w_{ij}^{(q)}\,(i \in Z_r, j \in Z_n, q \in Z_Q)$ に対する NG 法を実現することができる．このとき，NG 法で必要である各データと参照ベクトルの距離は式(21)により求めることができ，データおよび参照ベクトルを復号化することなく計算することができる．



(a) 参照ベクトルの更新量の計算　　(b) 参照ベクトルの更新

図 3 秘匿分散処理に基づく NG 法の提案手法の概略

図 3 に，秘匿分散処理に基づく NG 法の提案手法の概略を示す．提案手法においては，各サーバ内部において分解されたデータおよび参照ベクトルを用いた計算を行い，これらの計算結果をサーバ 0 に送る．サーバ 0 においては，各サーバから送られてきた計算結果を統合することで，式(22)中の参照ベクトルの更新量を計算する（図 3 (a)を参照）．また，サーバ 0 で計算された参照ベクトルの更新量を各サーバに送り，各サーバ内部において，分割された参照ベクトルの更新を行う（図 3 (b)を参照）．

3-2 秘匿分散処理計算に基づく BP 法と NG 法の改善手法

3-1 に示す提案手法は，分解された重みまたは参照ベクトルをパラメータとした機械学習である．しかしながら，この場合，重みまたは参照ベクトルの分解数が増えるとパラメータ数が増える．パラメータ数が増えると，機械学習における計算時間の増加などの問題が起こる．また，図 2 および図 3 に示すサーバ間の通信も多くなる．

3-2-1 秘匿分散処理に基づく BP 法の改善手法

3-1-1 に示す BP 法の提案手法においては，更新する重み $w_{ij}^{(q)}\,(i \in Z_P, j \in Z_n^*,\ q \in Z_Q)$，$v_{si}^{(q)}\,(s \in Z_S, i \in Z_P^*, q \in Z_Q)$ の個数は $\big(P(n+1)+S(P+1)\big)Q$ 個である．一方で，従来の BP 法において更新する重み $w_{ij}\,(i \in Z_P, j \in Z_n^*)$，$v_{si}\,(s \in Z_S, i \in Z_P^*)$ の個数は $P(n+1)+S(P+1)$ 個である．つまり，式(8)-(11)に示すデータおよび重みの分解数 Q が多いほど，更新する重みの個数が多くなる．更新する重みの個数が多いと，計算時間の増加などの問題がある．また，図 2 (b)について，サーバ 0 は Q 個のサーバに対して通信を行う必要があるため，データおよび重みの分解数 Q が多いほどサーバ間の通信回数が多くなる．

ここでは，3-1-1 に示す手法よりも，更新する重みの個数を削減した改善手法について述べる．

式(4)，(5)に示す重み $w_{ij}\,(i \in Z_P, j \in Z_n^*)$，$v_{si}\,(s \in Z_S, i \in Z_P^*)$ の更新式は，式(8)-(15)を用いると，次式のように書き換えることができる．

$$\Delta w_{ij} = \left(1 + K_w \sum_{s=1}^{S} \left\{ \sum_{q=1}^{Q} \alpha \left(d_s^{(q)}(\boldsymbol{x}^l) - h_s^{(q)}(\boldsymbol{x}^l) \right) \right\} h_s(\boldsymbol{x}^l)\left(1 - h_s(\boldsymbol{x}^l)\right) \left\{ \prod_{q=1}^{Q} v_{si}^{(q)} z_i^{(q)}(\boldsymbol{x}^l) \right\} \left(1 - z_i(\boldsymbol{x}^l)\right) \left\{ \prod_{q=1}^{Q} w_{ij}^{(q)} x_j^{l(q)} \right\} \cdot \left\{ \prod_{q=1}^{Q} \frac{\beta_1}{\left(w_{ij}^{(q)}\right)^2} \right\} \right) \tag{24}$$

$$\Delta v_{si} = \left(1 + K_v \left\{ \sum_{q=1}^{Q} \alpha \left(d_s^{(q)}(\boldsymbol{x}^l) - h_s^{(q)}(\boldsymbol{x}^l) \right) \right\} h_s(\boldsymbol{x}^l)\left(1 - h_s(\boldsymbol{x}^l)\right) \left\{ \prod_{q=1}^{Q} v_{si}^{(q)} z_i^{(q)}(\boldsymbol{x}^l) \right\} \cdot \left\{ \prod_{q=1}^{Q} \frac{\beta_2}{\left(v_{si}^{(q)}\right)^2} \right\} \right) \tag{25}$$

$$\prod_{q=1}^{Q} w_{ij}^{(q)} \leftarrow \prod_{q=1}^{Q} w_{ij}^{(q)} \times \Delta w_{ij} \times \frac{1}{\alpha \beta_1^Q} \tag{26}$$

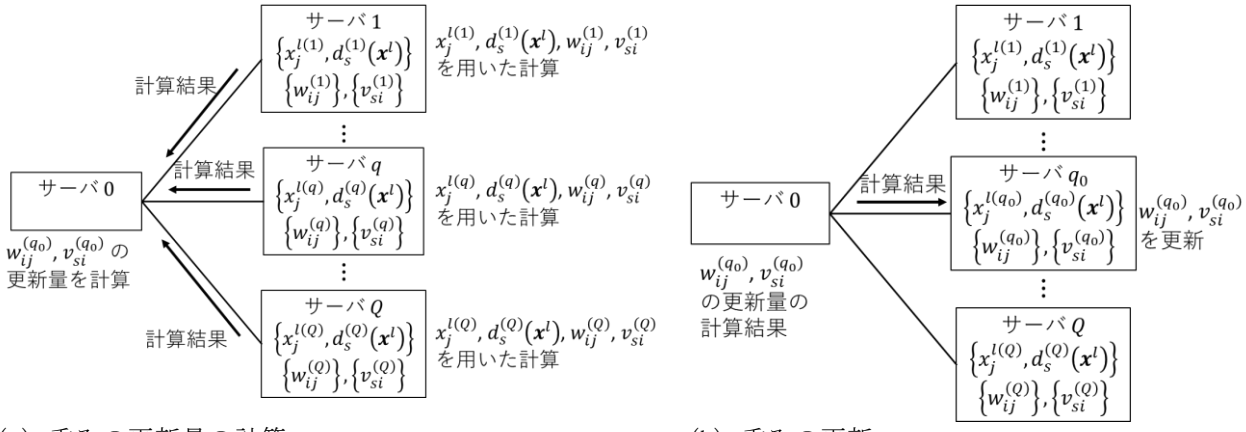
$$\prod_{q=1}^{Q} v_{si}^{(q)} \leftarrow \prod_{q=1}^{Q} v_{si}^{(q)} \times \Delta v_{si} \times \frac{1}{\alpha \beta_2^Q} \tag{27}$$

ここで，α，β_1，β_2はランダムに選ばれた実数値であり，分解データと重みを秘匿するために使用する．このとき，任意の整数値$q_0 \in Z_Q$に対して，次式(28)は式(26)を満たし，次式(29)は式(27)を満たす．

$$w_{ij}^{(q)} \leftarrow \begin{cases} w_{ij}^{(q)} \times \Delta w_{ij} \times \dfrac{1}{\alpha \beta_1^Q} & (q = q_0) \\ w_{ij}^{(q)} & (q \neq q_0) \end{cases} \tag{28}$$

$$v_{si}^{(q)} \leftarrow \begin{cases} v_{si}^{(q)} \times \Delta v_{si} \times \dfrac{1}{\alpha \beta_2^Q} & (q = q_0) \\ v_{si}^{(q)} & (q \neq q_0) \end{cases} \tag{29}$$

つまり，式(28)，(29)により$w_{ij}^{(q_0)}\ (i \in Z_P, j \in Z_n^*)$，$v_{si}^{(q_0)}\ (s \in Z_S, i \in Z_P^*)$を更新することは，通常のBP法において重み$\{w_{ij} | i \in Z_P, j \in Z_n^*\}$，$\{v_{si} | s \in Z_S, i \in Z_P^*\}$を更新することと等価となる．



(a) 重みの更新量の計算　　(b) 重みの更新

図 4 秘匿分散処理に基づく BP 法の改善手法の概略

図 4 に，秘匿分散処理に基づく BP 法の改善手法の概略を示す．提案手法においては，各サーバ内部において分解されたデータおよび重みを用いた計算を行い，これらの計算結果をサーバ 0 に送る．サーバ 0 においては，各サーバから送られてきた計算結果を統合することで，式(24)，(25)に示す重みの更新量を計算する（図 4 (a)を参照）．また，サーバ 0 で計算された重みの更新量をサーバq_0にのみ送り，サーバq_0内部において，重み$w_{ij}^{(q_0)}\ (i \in Z_P, j \in Z_n^*)$，$v_{si}^{(q_0)}\ (s \in Z_S, i \in Z_P^*)$の更新を行う（図 4 (b)を参照）．このとき，$w_{ij}^{(q_0)}\ (i \in Z_P, j \in Z_n^*)$，$v_{si}^{(q_0)}\ (s \in Z_S, i \in Z_P^*)$ 以外の重み$w_{ij}^{(q)}\ (i \in Z_P, j \in Z_n^*, q \neq q_0)$，$v_{si}^{(q)}\ (s \in Z_S, i \in Z_P^*, q \neq$

q_0)は更新しない．つまり，更新する重み$w_{ij}^{(q_0)}$ $(i \in Z_P, j \in Z_n^*)$，$v_{si}^{(q_0)}$ $(s \in Z_S, i \in Z_P^*)$の個数は$P(n+1)+S(P+1)$個であり，データおよび重みの分解数Qの個数には依存しない．

3-2-2 秘匿分散処理に基づく NG 法の改善手法

3-1-2 に示す NG 法の提案手法においては，更新する参照ベクトル$w_{ij}^{(q)}$ $\left(i \in Z_r, j \in Z_n, q \in Z_Q\right)$の個数は$rnQ$個である．一方で，従来の NG 法において更新する参照ベクトルw_{ij} $(i \in Z_r, j \in Z_n)$の個数はrn個である．つまり，式(19)，(20)に示すデータおよび参照ベクトルの分解数Qが多いほど，更新する参照ベクトルの個数が多くなる．更新する参照ベクトルの個数が多いと，計算時間の増加などの問題がある．また，図 3 (b)について，サーバ 0 はQ個のサーバに対して通信を行う必要があるため，データおよび参照ベクトルの分解数Qが多いほどサーバ間の通信回数が多くなる．ここでは，3-1-2 に示す手法よりも，更新する重みの個数を削減した改善手法について述べる．

式(7)に示す参照ベクトルw_{ij} $(i \in Z_r, j \in Z_n)$ の更新式は，式(19)，(20)を用いると，次式のように書き換えることができる．

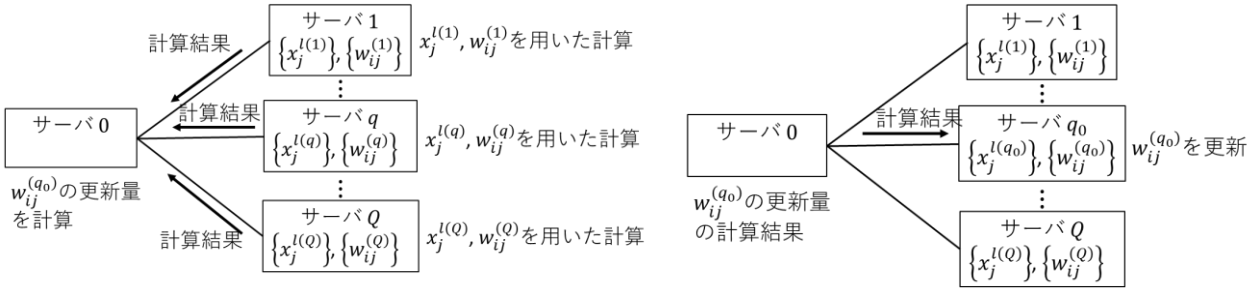
$$\Delta w_{ij} = \varepsilon \cdot \exp\left(-\frac{e_i(\boldsymbol{x}^l)}{\lambda}\right)\sum_{q=1}^{Q}\left(x_j^{l(q)} - w_{ij}^{(q)}\right) \tag{30}$$

$$\sum_{q=1}^{Q} w_{ij}^{(q)} \leftarrow \sum_{q=1}^{Q} w_{ij}^{(q)} + \Delta w_{ij} \tag{31}$$

このとき，任意の整数値$q_0 \in Z_Q$に対して，次式(32)は式(31)を満たす．

$$w_{ij}^{(q)} \leftarrow \begin{cases} w_{ij}^{(q)} + \Delta w_{ij} (q = q_0) \\ w_{ij}^{(q)} \qquad (q \neq q_0) \end{cases} \tag{32}$$

つまり，式(32)により$w_{ij}^{(q_0)}$ $(i \in Z_r, j \in Z_n)$を更新することは，通常の NG 法において参照ベクトル$\{w_{ij} | i \in Z_r, j \in Z_n\}$を更新することと等価となる．



(a) 参照ベクトルの更新量の計算　　(b) 参照ベクトルの更新

図 5 秘匿分散処理に基づく BP 法の改善手法の概略

図 5 に，秘匿分散処理に基づく NG 法の改善手法の概略を示す．提案手法においては，各サーバ内部において分解されたデータおよび参照ベクトルを用いた計算を行い，これらの計算結果をサーバ 0 に送る．サーバ 0 においては，各サーバから送られてきた計算結果を統合することで，式(30)に示す重みの更新量を計算する（図 5 (a)を参照）．また，サーバ 0 で計算された参照ベクトルの更新量をサーバq_0にのみ送り，サーバq_0内部において，参照ベクトル$w_{ij}^{(q_0)}$ $(i \in Z_r, j \in Z_n)$の更新を行う（図 5 (b)を参照）．このとき，$w_{ij}^{(q_0)}$ $(i \in Z_r, j \in Z_n)$ 以外の参照ベクトル$w_{ij}^{(q)}$ $(i \in Z_r, j \in Z_n, q \neq q_0)$は更新しない．つまり，更新する参照ベクトル$w_{ij}^{(q_0)}$ $(i \in Z_r, j \in Z_n)$の個数はrn個であり，データおよび重みの分解数Qの個数には依存しない．

3-3 数値実験

ここでは，提案手法と，データを分割しない従来手法の比較のために，数値実験を行った結果を掲載する．

3-3-1 BP 法における比較

ここでは，2-2 に示す BP 法の従来手法，および 3-1-1，3-2-1 に示す BP 法の提案手法を用いて，ベンチマークデータである Iris データに対して 5-fold cross validation による検証を行い，更新する重みの個数，学習用データおよびテスト用データに対する誤分類率の比較を行った [21]．ここで，Iris データのデータ数は 150，入力要素数は 4，分類数は 3 である．実験条件は，中間層ニューロ数が 10，最大学習回数 50 万回，しきい値は 0.03 とした．また，提案手法については，データおよび重みを 5 個に分割した．

表 1 に，各手法における，更新する重みの個数，学習用データおよびテスト用データの誤分類率(%)を示す．ここで，表 1 中の# Para は 1 度に更新する重みの個数，Learn は学習用データに対する誤分類率(%)，Test はテスト用データに対する誤分類率(%)を示す．表中の各値は 20 回試行の平均値である．

表 1　BP 法を用いた数値実験の結果

	# Para	Learn	Test
従来手法 (2-2)	83	1.6	3.4
提案手法 (3-1-1)	415	2.2	4.9
提案手法 (3-2-1)	83	1.7	3.5

表 1 より，更新する重みの個数は，3-1-1 に示す提案手法は 2-2 に示す BP 法の手法よりも多いが，3-2-1 に示す提案手法は 2-2 に示す BP 法の提案手法と同等の個数である．また，学習用およびテスト用データに対する誤分類率は，3-1-1 および 3-2-1 に示す提案手法は 2-2 に示す BP 法の従来手法と同等の分類精度を示している．

3-3-2 NG 法における比較

ここでは，2-3 に示す NG 法の従来手法，および 3-1-2，3-2-2 に示す NG 法の提案手法を用いて，表ベンチマークデータである Iris データに対するクラスタリングを行った [21]．実験条件は，参照ベクトルの個数を 3，最大学習回数は 15000 とした．提案手法については，データおよび参照ベクトルを 5 個に分割した．

表 2 に実験結果を示す．ここで，表 2 中の# Para は 1 度に更新する参照ベクトルの個数，MR はクラスタリングによるデータの分類を行った結果に対する誤分類率(%)，MSE は式(6)，(22)に示す評価関数の値($\times 10^{-3}$)を示す．表中の各値は 20 回試行の平均値である．

表 2　NG 法を用いた数値実験の結果

	# Para	MR	MSE
従来手法 (2-2)	12	4.1	6.46
提案手法 (3-1-1)	60	4.1	6.49
提案手法 (3-2-1)	12	4.1	6.43

表 2 より，更新する参照ベクトルの個数は，3-1-2 に示す提案手法は 2-3 に示す NG 法の手法よりも多いが，3-2-2 に示す提案手法は 2-3 に示す NG 法の提案手法と同等の個数である．また，データの誤分類率および評価関数の値については，3-1-2 および 3-2-2 に示す提案手法は 2-3 に示す BP 法の従来手法と同等の分類精度を示している．

4 まとめ

本研究では，エッジシステムにおいて機械学習を安全に行う方法として，SMC と組み合わせた機械学習のアルゴリズムについての研究に取り組んだ．提案モデルにおいては，機械学習における学習用データおよびパラメータが複数に分解されて管理されている状況を想定し，これらのデータおよびパラメータを復号化することなく BP 法および NG 法による機械学習を実現するアルゴリズムについての研究を行った．著者らは提

案手法の一つとして，分解したパラメータすべてを更新する機械学習アルゴリズムを提案した．この提案手法においてはデータおよびパラメータは復号化されないためにデータの安全性が高められている．また，この提案手法は BP 法および NG 法における機械学習の理論に即しており，数値実験においても従来手法と同等の高い精度を示した．しかしながら，この手法においては，更新するパラメータの個数が多くなり，計算手順の増加やサーバ間の通信回数の増加が懸念される．そこで，パラメータ数について考慮した改善手法を提案した．この提案手法においては，複数に分解したパラメータのうち一部のみを更新することでパラメータ数を抑える．この提案手法においてはデータおよびパラメータは復号化されないためにデータの安全性が高められている．また，この提案手法は従来の BP 法および NG 法における機械学習の計算と等価であり，数値実験においても従来手法と同等の高い精度を示した．

今後の課題としては，他の機械学習手法についても，SMC と組み合わせることでデータの安全性を高めたアルゴリズムについても検討する必要がある．

【参考文献】

[1] C. C. Aggarwal, et.al.,"Privacy Preserving Data Mining: Models and Algorithms", ISBN 978-0-387-70991-8, Springer-Verlag, 2009.

[2] A. Shamir, "How to share a secret", Comm. ACM, vol. 22, no. 11, pp.612-613, 1979.

[3] A. Beimel, "Secret-sharing schemes: a survey", Proc. of the Third international conference on Coding and cryptology (IWCC 11), 2011.

[4] D. Evans, et.al., "A Pragmatic Introduction to Secure Multi-Party Computation", Foundations and Trends in Privacy and Security, vol.2, Issue 2-3, pp. 70-246, 2022.

[5] R. Canetti, U. Feige, O. Goldreich, and M. Naor, Adaptively secure multi-party computation, STOC' 96, pp. 639-648, 1996.

[6] R. Cramer, et.al., "General secure multi-party computation from any linear secret-sharing scheme", EUROCRYPT', pp.331-339, 2000.

[7] A. Ben-David, et.al., "Fair play MP: a system for secure multi-party computation", ACM CCS' 08, pp. 257-266, 2008.

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

[9] Q. Yang, et.al.,"Federated Machine Learning：Concept and Applications", ACM Trans. Intell. Syst. Technol., vol. 10, no. 2, Article 12, 2019.

[10] J. Koneˇcn´y, et.al.,"Federated Learning: Strategies for Improving Communication Efficiency", arXiv:1610.05492, 2017.

[11] B. McMahan, et.al., "Communication-Efficient Learning of Deep Networks from Decentralized Data", Proc. of Machine Learning Research, vol. 54, pp.1273-1282, 2017.

[12] J. Yuan, et.al., "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.

[13] N. Schlitter, "A Protocol for Privacy Preserving Neural Network Learning on Horizontal Partitioned Data", Privacy Statistics in Databases (PSD), 2008.

[14] X. Ma, et.al., "Privacy preserving multi-party computation delegation for deep learning in cloud computing", Information Sciences, vol. 459, pp. 103-116, 2018.

[15] O. Nassef, et.al., "A survey: Distributed Machine Learning for 5G and beyond", Computer Networks, 108820, ISSN 1389-1286, 2022.

[16] J. Duan, et.al., "Privacy-preserving and verifiable deep learning inference based on secret sharing", Neurocomputing, vol. 483, pp.221-234, 2022.

[17] X. Yin, et.al., "A Comprehensive Survey of Privacy-preserving Federated Learning: A Taxonomy, Review, and Future Directions", ACM Computing Surveys, vol.54, no. 6, pp. 1-36, 2022.

[18] M. M. Gupta, et.al.,"Static and Dynamic Neural Networks", IEEE Pres, Wiley-Interscience, 2003.
[19] T. M. Martinetz, et.al.," '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.
[20] J. Kennedy, et.al.," Particle swarm optimization", Proc. International Conference on Neural Networks, pp.1942-1948, 1995.
[21] UCI Repository of Machine Learning Databases and Domain Theories, https://archive.ics.uci.edu/ml/datasets.php.

〈発 表 資 料〉

題　　名	掲載誌・学会名等	発表年月
Learning algorithms for vector quantization using vertically partitioned data with IoT	Artificial Life and Robotics, pp.283-290	2021 年 8 月
IoT に対する安全な PSO を用いた学習法の提案	第 74 回電気・情報関係学会九州支部連合大会	2021 年 9 月
Securely Distributed Computation with Divided Data for Particle Swarm Optimization	International MultiConference of Engineers and Computer Scientists	2021 年 10 月
Federated Learning with Divided Data for BP	International MultiConference of Engineers and Computer Scientists	2021 年 10 月
秘匿分解データを用いた新しい機械学習	第 189 回マルチメディア通信と分散処理研究発表会	2021 年 12 月
The generalized BP method of secret distributed processing with divided data	International Symposium on Artificial Life and Robotics	2022 年 1 月
Machine Learning with Distributed Processing using Secure Divided Data: Towards Privacy-Preserving Advanced AI Processing in a Super-Smart Society	Journal of Networking and Network Applications, vol.2, issue.1, pp.48-60	2022 年 4 月
Secure Learning Systems using Vertically Partitioned Data with IoT	IAENG International Journal of Computer Science, vol.49, no.1, pp.61-68	2022 年 5 月
Securely Distributed Computation with Divided Data and Parameters for Hybrid Particle Swarm Optimization	IAENG International Journal of Applied Mathematics, vol.52, no.3, pp.541-549	2022 年 9 月
秘匿分解データによる BP 学習アルゴリズムの改善	第38回ファジィシステムシンポジウム	2022 年 9 月
Secure Distributed Processing of BP with updatable decomposition Data	Transactions on Engineering Technologies, pp.1-15	2022 年 10 月
Secure Distributed Processing of NG with Updatable Decomposition Data and Parameters	International Conference on Networking and Network Applications	2022 年 12 月
Simplified Secure Distributed Processing of BP with Decomposition Data	International Symposium on Nonlinear Theory and Its Applications	2022 年 12 月
秘匿分散処理による機械学習法の計算量削減	バイオメディカル・ファジィ・システム学会年次大会	2022 年 12 月
秘匿分解データによる BP 学習アルゴリズムの計算量削減	日本知能情報ファジィ学会誌, vol.35, no.1, pp.506-510	2023 年 2 月
Scalability improvement of simplified,	Recent Progress in Nonlinear	2023 年 4 月

secure distributed processing with decomposition data	Theory and Its Applications, vol.14, no.2, pp.140-151	
Neural Gas and K-Means Methods Improving the Computational Complexity for Secure Distributed Processing	Biomedical Soft Computing and Human Sciences, vol.28, no.1, pp.15-22	in print (2023年3月 掲載決定)