加法準同型暗号を用いたプライバシー保護 Extreme Learning Machine

栗　昌平[1]　林　卓也[1,2]　大森　敏明[1]　小澤　誠一[1]　青野　良範[2]　Le Trieu Phong[2]　王　立華[2]
盛合　志帆[2]

概要：クラウドサーバの需要が高まる一方，情報漏洩への懸念が外部サーバ利用の障壁となっている．こうした現状を踏まえて，本稿では，加法準同型暗号を用いたプライバシー保護 Extreme Learning Machine(PP-ELM) を提案する．提案手法では，データ提供者，代理計算サーバ，データ分析者の三者を考える．データ提供者はデータの前処理を行い，加法準同型暗号でデータの暗号化を行う．代理計算サーバは暗号文を受け取り，加法演算を行う．データ分析者は代理計算サーバから演算結果を受け取り，復号した後，それを用いて，ELM のパラメータを求める．これにより代理計算サーバにおけるデータの漏洩のリスク無しに，外部サーバを用いた代理計算によるデータ解析が可能となる．

キーワード：PWS，機械学習，加法準同型暗号，識別器，ニューラルネット

Privacy Preserving Extreme Learning Machine Using Additively Homomorphic Encryption

Shohei Kuri[1]　Takuya Hayashi[1,2]　Toshiaki Omori[1]　Seiichi Ozawa[1]　Yoshinori Aono[2]
Le Trieu Phong[2]　Lihua Wang[2]　Shiho Moriai[2]

Abstract: We propose a privacy preserving Extreme Learning Machine (PP-ELM) using additively homomorphic encryption. We consider a three participants model; data contributors, an outsourced server, and a data analyst. The data contributor preprocesses the data and encrypts it with additively homomorphic encryption. The outsourced server receives the encrypted data and performs summation on the encrypted data. The data analyst receives the summation from the outsourced server and decrypts it, then uses it to obtain an optimized parameter of ELM. The proposed outsourcing model is expected to mitigate a hurdle of personal data usage on a cloud service.

Keywords: PWS, Machine Learning, Additively Homomorphic Encryption, Classifier, Neural Network

1. 序論

近年，IoT や機械学習の発展により，ビッグデータ解析の需要が高まっている．大規模なデータセットを用いた機械学習では，膨大な計算コストを必要とする．そのため，データ解析におけるクラウドサーバの利用が広まっている．ユーザはクラウドサーバに重たい計算や煩雑なデータ管理を委託できるため，アウトソーシングはビッグデータ時代においてますます重要になると考えられる．

一方で，データに含まれる個人情報や機密情報が，クラウドサーバ管理者や外部者へと漏洩してしまうことが危惧されている．情報漏洩により，データの管理や解析を行っていた企業の社会的信用が毀損されてしまう．またデータ提供者も思わぬ損害を受けるリスクを抱えている．そういっ

[1] 神戸大学大学院工学研究科電気電子工学専攻
Department of Electrical and Electronic Engineering, Graduate School of Engineering, Kobe University
[2] 国立研究開発法人　情報通信研究機構
National Institute of Information and Communications Technology

たリスクから，多くの人がクラウドサーバにおけるデータ解析に躊躇する状況がある．さらに，複数の組織間でのデータ統合は正確な分類器の生成や社会問題の解決のきっかけになりうるが，法律やプライバシーへの懸念から，機密情報を含んだデータの統合や解析が難しい現状がある．つまりプライバシーや機密データが，クラウドサーバの利活用や複数の組織間のデータ解析を阻む壁となっている．

こういった現状を踏まえて，様々なモデルのプライバシー保護を可能としたデータ解析手法が提案されている．chauhuri らは差分プライバシー [1] を基にしたロジスティック回帰 [2] を提案している．このロジスティック回帰では，データに適切な大きさのノイズを載せることによって，データのプライバシーを保護し，プライバシー保護と精度のトレードオフについて述べている．暗号文に対する機械学習手法としては，Bost らが secure two-party computation によって，データの秘匿性を実現し，ナイーブベイズや決定木に基づく予測を行った [3]．これらの分類モデルでは，サーバが学習済みの分類器を持ち，クライアントがデータを持つ状況を考える．そして，サーバの分類器の情報やクライアントの持つ情報を漏らすことなく，クライアントは予測結果を得ることができる．しかしながら，サーバとクライアント間の通信が数回必要であり，通信コストが大きくなる傾向がある．Gilad-Bachrach らは暗号文を用いたニューラルネットワーク [4] における予測フェーズを提案している．この手法では，クラウドサーバにおける学習済みのニューラルネットワークを用いて，暗号文を用いた予測ができる．しかし，この手法では暗号文を用いた学習フェーズを考えていない．予測フェーズのみをプライバシー保護可能とする上記二つの手法に対して，青野らは学習と予測の両方を暗号文を用いて行えるロジスティック回帰 [5] を提案している．この手法では，ロジスティック回帰のシグモイド関数を近似することにより，準同型暗号によるクラウドサーバでの演算を利用して，ロジスティック回帰の回帰係数を学習するモデルを構築している．

本稿では，準同型暗号を用いたプライバシー保護を可能とする Extreme Learning Machine (Privacy Preserving Extreme Learning Machine: PP-ELM) を提案する．ELM は single layer feedforward neural networks(SLFNs) の一種であり，非線形分類が可能である．また ELM は解析的にパラメータを求めることができ，通常のニューラルネットワークと比較して，速い学習速度を誇る．提案手法では，[5] と同様の代理計算モデルを考える．このモデルでは，データ提供者，代理計算サーバ，データ分析者の三者で構成される．データ提供者は複数いることを仮定し，それぞれのデータ提供者は個々が持つデータに対する ELM の隠れ層の出力までを前処理として計算し，暗号化する．このデータ提供者による前処理は，代理計算サーバにおける全てのデータに対する隠れ層の出力行列の計算を暗号文で可能とする．この計算結果が ELM の求めたいパラメータを得るのに必要となる．最後にデータ分析者が代理計算サーバから暗号文を受け取り，復号したのち，ELM のパラメータを計算する．ELM を使用する最大の利点は，準同型暗号で扱えるような簡易なモデルで，非線形分類器としての高い精度を実現できる点である．提案手法では代理計算サーバに訓練データの情報を漏らすことなく，ELM のパラメータの学習を可能とする．さらに，データ数を N，ELM の隠れ層のノード数を L とすると，データ提供者とデータ分析者の計算コストはそれぞれ $O(L^2)$，$O(L^3)$ である．対して，代理計算サーバにおける計算コストは $O(NL^2)$ である．つまり，提案するモデルではデータ数 N に依存する計算コストを外部化できる．

本稿の貢献は以下の通りである．

(1) 非線形分類器の一つである ELM を，準同型暗号を用いた代理計算モデルで実現した．
(2) 提案手法では，通信は一方向であり，双方向通信を行わない．
(3) データ提供者が ELM の活性化関数による非線形変換といくつかの乗算をデータの前処理として行うことにより，代理計算サーバでの ELM の隠れ層の出力行列の計算を準同型加算のみで実現した．
(4) 複数の異なる企業間でのデータ解析を可能とする．
(5) 提案手法により，データ数 N に依存した計算コストを代理計算サーバに外部化できる．

2. プライバシー保護技術

本章では，プライバシー保護を可能とするために本研究で使用した準同型暗号と代理計算のモデルについて述べる．

2.1 準同型暗号

データ解析においてはデータの数値に対する様々な計算が必要となる．しかしながら，RSA 暗号 [6] のような基本的な暗号によってデータが暗号化されると，復号せずにそのデータに対して計算を行うことはできない．この問題を解決するために，準同型暗号と呼ばれる特別な暗号が提案された [7], [8], [9]．これらの暗号を用いて，安全にクラウドサーバへ計算を委託できる様々な機械学習手法が提案されている [3], [4], [10].

準同型暗号において，秘密鍵 sk と公開鍵 pk を一つの鍵ペアとする．秘密鍵 sk はオーナーにより秘密に保持されなければならない．また対応する公開鍵 pk はすべての人に公開される．平文空間 $\mathcal{M}$ における平文 m_1, m_2 を考える．準同型暗号では，平文における演算 '$\cdot$' に対応する暗号文における演算 '$\odot$' が存在する．

$$m_1 \cdot m_2 = \mathsf{Dec}(sk,\ \mathsf{Enc}(pk,\ m_1) \odot \mathsf{Enc}(pk,\ m_2)), \quad (1)$$

ここで，Enc は暗号化関数，Dec は復号関数とする．$\mathcal{M}$ は

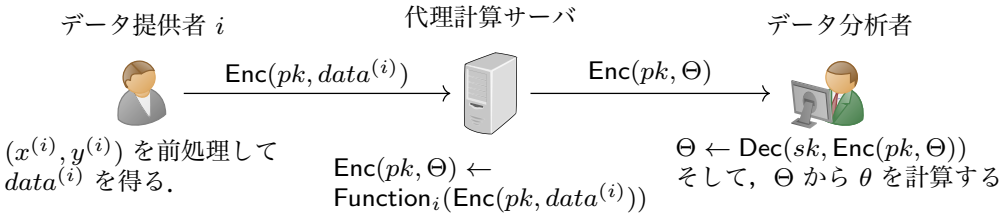

図 1: 代理計算モデル
Fig. 1 Our outsourcing model

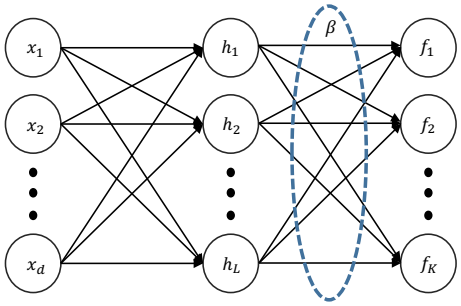

図 2: ELM のモデル
Fig. 2 The structure of ELM

大抵の場合，実数値 $\mathbb{R}$ ではないため，$\mathbb{R}$ から $\mathcal{M}$ への暗号方式に依存するエンコードを行う必要がある．

長らく準同型暗号は加算や乗算のどちらか一方のみを可能としていた．しかし，2009 年に Gentry が完全準同型暗号を実現した [11]．完全準同型暗号は加算，乗算について制限がなく，任意の演算回路を暗号文上で計算できる．これは任意の関数を暗号文で計算できることを意味するが，この暗号方式の実用性には，暗号化や復号，準同型演算にとても大きな計算リソースを必要とするという理由からまだ疑問が残る．そのため，私たちの提案手法では，加法準同型暗号を使用する．加法準同型暗号は Paillier 暗号 [8] や lifted-ElGamal 暗号 [9] などがあり，暗号文での加法演算を非常に高速に行うことができる．

2.2 代理計算のモデル

概要: 代理計算のモデルの概要を図 1 に示す．各データ $(x^{(i)}, y^{(i)})(i = 1, ..., N)$ を持つデータ提供者 i がデータを事前に処理して，$data^{(i)}$ を得る．そして，データ分析者の公開鍵 pk を使って暗号化し，代理計算サーバに送る．代理計算サーバはこれらの暗号文を受け取り，あらかじめ決めた演算を行う．この結果をデータ分析者に送る．データ分析者は自身の持つ秘密鍵を用いて，代理計算サーバから送られた結果を復号したのち，モデルのパラメータである θ を求める．提案モデルでは加法準同型暗号を用いるため，代理計算サーバで行われる演算は暗号文に対する加法演算のみである．

安全性モデル: 提案手法では，データ提供者とデータ分析者は正直者であると想定する．すなわち，データ提供者は悪意がなく，バグのあるデータを送らない．またデータ分析者はデータ提供者から信頼されていることとする．一方で代理計算サーバは honest-but-curious で，つまり依頼された計算は正確に行うが，機会があればデータを覗き見すると想定する．提案するモデルの目的は，代理計算サーバに対するデータの情報漏洩を防ぐことである．

妥当性: 上記の想定のもとでは，データ提供者はデータ分析者に直接データを送り，代理計算サーバなしに機械学習手法を適用することが可能である．しかしながら，ビッグデータのようなとてつもなく大規模なデータを処理する場合，データ分析者は安全にそのデータを保持，管理し，データ提供者と通信を行う必要がある．これらはデータ分析者にとって大きな負担となると思われる．提案手法では，データ分析者がクラウドサーバにこれらの負担を安全に委託することができる．そのため，提案手法の使用は妥当性があると考える．

3. Extreme Learning Machine

Extreme Learning Machine(ELM)[12] は Single hidden layer feedforward neural networks(SLFNs) の一種であり，図 2 のように表すことができる．ELM と似たようなネットワーク構造は [13] などでもみられる．ELM は入力層と隠れ層間のパラメータをランダムに設定し，学習しない点で SLFNs と大きく異なる．また ELM は隠れ層と出力層間のパラメータを Moore-Penrose の一般化逆行列 [14] により解析的に求めることができる．そのため，ELM は一般的な SLFNs に比べて，速い学習速度となる．

学習対象となるデータセットを $\boldsymbol{Data} = [(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), \ldots, (\boldsymbol{x}^{(N)}, \boldsymbol{y}^{(N)})]$ とする．ただし，データ数を N，クラス数を K とし，$\boldsymbol{x}^{(i)}(i = 1, 2, \ldots, N)$ は d 次元の実数ベクトル $\boldsymbol{x} = (x_1, ..., x_d)$ であり，$\boldsymbol{y}^{(i)}$ は $\boldsymbol{y}^{(i)} = (y_1^{(i)}, y_2^{(i)}, \ldots, y_K^{(i)})$ となる One-hot vector である．また ELM について，隠れ層のノードの個数を L とし，活性化関数を $G(\boldsymbol{a}, b, \boldsymbol{x})$ とする．ここで，$\boldsymbol{a} \in \mathbb{R}^d$ は入力層と隠れ層間の重みベクトルであり，$b \in \mathbb{R}$ はバイア

Require: $\boldsymbol{Data}$, $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_L)$, $(b_1, \ldots, b_L)$
Ensure: $\boldsymbol{\beta}$
1: 式 (4) の隠れ層の出力行列 $\boldsymbol{H}$ を計算する.
2: **if** $\boldsymbol{H}$ の行ランク $>$ $\boldsymbol{H}$ の列ランク **then**
3: 式 (6) で $\boldsymbol{\beta}$ を計算する.
4: **else**
5: 式 (7) で $\boldsymbol{\beta}$ を計算する.
6: **end if**

図 3: ELM の学習アルゴリズム
Fig. 3 Learning of ELM

スである．また本稿では，活性化関数としてシグモイド関数を用いる．

$$G(\boldsymbol{a},\ b,\ \boldsymbol{x}) = \frac{1}{1+\exp\left(-(\boldsymbol{a}\cdot\boldsymbol{x}+b)\right)}. \tag{2}$$

このとき，ELM の最小化をするコスト関数は次式のようになる．

$$J = ||\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{H}\boldsymbol{\beta} - \boldsymbol{Y}||^2. \tag{3}$$

ここで，$\boldsymbol{\beta}$ は隠れ層と出力層間の重みであり，λ は正則化に影響するハイパーパラメータ，$\boldsymbol{H}$ はすべての入力データに対する隠れ層の出力行列であり，

$$\begin{aligned}\boldsymbol{H} &= \begin{bmatrix} \boldsymbol{h}^{(1)} \\ \vdots \\ \boldsymbol{h}^{(N)} \end{bmatrix} \\ &= \begin{bmatrix} h_1^{(1)} & \ldots & h_L^{(1)} \\ \vdots & \ldots & \vdots \\ h_1^{(N)} & \ldots & h_L^{(N)} \end{bmatrix} \\ &= \begin{bmatrix} G(\boldsymbol{a_1}, b_1, \boldsymbol{x}^{(1)}) & \ldots & G(\boldsymbol{a_L}, b_L, \boldsymbol{x}^{(1)}) \\ \vdots & \ldots & \vdots \\ G(\boldsymbol{a_1}, b_1, \boldsymbol{x}^{(N)}) & \ldots & G(\boldsymbol{a_L}, b_L, \boldsymbol{x}^{(N)}) \end{bmatrix}\end{aligned} \tag{4}$$

Y はラベルデータの行列であり，

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}^{(1)} \\ \vdots \\ \boldsymbol{y}^{(N)} \end{bmatrix} = \begin{bmatrix} y_1^{(1)} & \ldots & y_K^{(1)} \\ \vdots & \ldots & \vdots \\ y_1^{(N)} & \ldots & y_K^{(N)} \end{bmatrix} \tag{5}$$

と表される．

ELM の学習アルゴリズムを図 3 と以下にまとめた．

Step1
入力層と隠れ間の重みベクトルとバイアス $\boldsymbol{a}_l,\ b_l$ $(l = 1,\ 2, \ldots,\ L)$ をランダムに決める．

Step2
すべての入力データに対応する隠れ層の出力行列 $\boldsymbol{H}$ を計算する．

Step3
次式により隠れ層と出力層間の重み $\boldsymbol{\beta}$ を $\boldsymbol{H}$ の Moore-Penrose の一般化逆行列を用いて求める．$\boldsymbol{H}$ の行ランクが列ランクより大きい場合，

$$\boldsymbol{\beta} = \boldsymbol{H}^T\left(\frac{\boldsymbol{I}}{\lambda} + \boldsymbol{H}\boldsymbol{H}^T\right)^{-1}\boldsymbol{Y}, \tag{6}$$

$\boldsymbol{H}$ の列ランクが行ランクより大きい場合，

$$\boldsymbol{\beta} = \left(\frac{\boldsymbol{I}}{\lambda} + \boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{Y}. \tag{7}$$

次に予測フェーズについて述べる．ELM の出力を $\boldsymbol{f}$ とすると，次式のように表すことができる．

$$\boldsymbol{f} = \boldsymbol{h}\boldsymbol{\beta}. \tag{8}$$

ここで $\boldsymbol{f} = (f_1, \ldots, f_K)$，入力データ $\boldsymbol{x}$ に対する隠れ層の出力は $\boldsymbol{h} = (h_1, \ldots, h_L)$ である．予測するクラスラベル f_c^* は以下の式で求められる．

$$f_c^* = \underset{c\in\{1,\ldots,K\}}{\operatorname{argmax}}\ f_c. \tag{9}$$

4. プライバシー保護を可能とする ELM(PP-ELM)

4.1 代理計算のプロセス

この章では，代理計算における暗号化や処理方法について述べる．各データ提供者 i $(i = 1,\ 2, \ldots,\ N)$ を考え，それぞれのデータ提供者は一つのデータを提供すると考える．ここでは大規模なデータセットを想定するため，N は充分大きいと考えられる．暗号文に対する計算のため，まず式（6），(7）の一部を式展開する．N が充分に大きいことから，$N \gg L$ と仮定できる．すなわち，$\boldsymbol{H}$ の列ランクが行ランクよりも大きいと仮定できるから，ELM の求めたいパラメータ $\boldsymbol{\beta}$ は式（7）で計算できる．

ここで $\boldsymbol{A} = \boldsymbol{H}^T\boldsymbol{H}$, $\boldsymbol{B} = \boldsymbol{H}^T\boldsymbol{Y}$ とすると，これらの式は次のように式展開できる．

$$\boldsymbol{A} = \begin{bmatrix} \sum_{i=1}^N h_1^{(i)}h_1^{(i)} & \ldots & \sum_{i=1}^N h_1^{(i)}h_L^{(i)} \\ \vdots & \ldots & \vdots \\ \sum_{i=1}^N h_L^{(i)}h_1^{(i)} & \ldots & \sum_{i=1}^N h_L^{(i)}h_L^{(i)} \end{bmatrix} \tag{10}$$

$$\boldsymbol{B} = \begin{bmatrix} \sum_{i=1}^N h_1^{(i)}y_1^{(i)} & \ldots & \sum_{i=1}^N h_1^{(i)}y_K^{(i)} \\ \vdots & \ldots & \vdots \\ \sum_{i=1}^N h_L^{(i)}y_1^{(i)} & \ldots & \sum_{i=1}^N h_L^{(i)}y_K^{(i)} \end{bmatrix}. \tag{11}$$

ここで,

$$A_{r1,r2} = \sum_{i=1}^N h_{r1}^{(i)}h_{r2}^{(i)} \tag{12}$$

$$B_{r,k} = \sum_{i=1}^N h_r^{(i)}y_k^{(i)} \tag{13}$$

と定義する．ただし $1 \le r,\ r1,\ r2 \le L,\ 1 \le k \le K$ であ

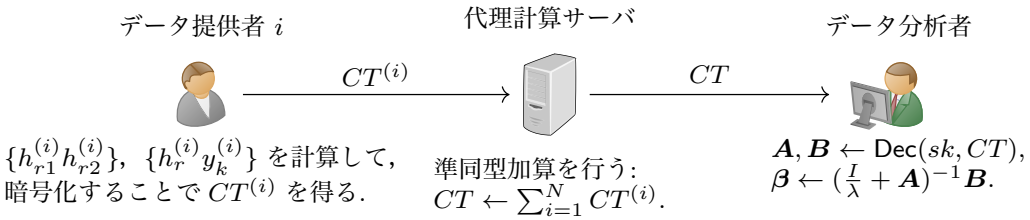

図 4: 具体的な代理計算モデル

Fig. 4 Concrete process for outsourcing

Require: $\boldsymbol{x}^{(i)}$, $\boldsymbol{y}^{(i)}$, $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_L)$, $(b_1, \ldots, b_L)$
Ensure: $\boldsymbol{\beta}$
1: (C) $\boldsymbol{h}^{(i)} = \left(G(\boldsymbol{a_1}, b_1, \boldsymbol{x}^{(i)}), \ldots, G(\boldsymbol{a_L}, b_L, \boldsymbol{x}^{(i)})\right)$ を計算する．
2: (C) $h_{r1}^{(i)}h_{r2}^{(i)}$, $h_r^{(i)}y_k^{(i)}$ を計算する．
3: (C) $h_{r1}^{(i)}h_{r2}^{(i)}$, $h_r^{(i)}y_k^{(i)}$ を式 (15) の $\boldsymbol{D}^{(i)}$ としてベクトル化する．
4: (C) $\boldsymbol{D}^{(i)}$ を $CT^{(i)} = \mathsf{Enc}\left(pk, \boldsymbol{D}^{(i)}\right)$ と暗号化する．
5: (C) $CT^{(i)}$ を代理計算サーバに送る．
6: (S) 式 (17) のように $CT = \sum_{i=1}^{N} CT^{(i)}$ を計算する．
7: (S) CT をデータ分析者に送る．
8: (A) CT を復号し，$\boldsymbol{A} = \boldsymbol{H}^T\boldsymbol{H}$, $\boldsymbol{B} = \boldsymbol{H}^T\boldsymbol{Y}$ を得る．
9: (A) $\boldsymbol{A}$，$\boldsymbol{B}$ を用いて，式 (7) で $\boldsymbol{\beta}$ を計算する．

図 5: PP-ELM の学習フェーズ

Fig. 5 Learning phase of PP-ELM

る．$A_{r1,r2}$ は行列 $\boldsymbol{A}$ の $(r1, r2)$ 成分を表し，$B_{r,k}$ は行列 $\boldsymbol{B}$ の (r, k) 成分である．データ提供者 i は式 (12)，(13) の $h_{r1}^{(i)}h_{r2}^{(i)}$ と $h_r^{(i)}y_k^{(i)}$ を計算し，加法準同型暗号でそれらを暗号化する．データ提供者によるこれらの計算により，代理計算サーバは復号することなく，式 (12)，(13) を計算することで，$\boldsymbol{A}$ と $\boldsymbol{B}$ の暗号化されたすべての要素を計算できる．代理計算サーバはデータ分析者にこの加法演算の結果を送る．データ分析者は結果を復号することで，$\boldsymbol{A}$，$\boldsymbol{B}$ を得る．そして ELM のパラメータである $\boldsymbol{\beta}$ を求めることができる．

ELM の代理計算のプロセスの概要を図 4 と図 5 にまとめた．図 5 において，(C)，(S)，(A) はそれぞれデータ提供者，代理計算サーバ，データ分析者を表す．まずそれぞれのデータ提供者は前処理として，自身のデータに対する ELM の隠れ層の出力である $h^{(i)}$ を計算する．そして，$h_{r1}^{(i)}h_{r2}^{(i)}(1 \le r1, r2 \le L)$，$h_r^{(i)}y_k^{(i)}(1 \le r \le L,\ 1 \le k \le K)$ を計算する．さらにデータ提供者はそれらを暗号化して，暗号文 $CT^{(i)}$ を得て，この $CT^{(i)}$ を代理計算サーバに送る．代理計算サーバは $CT = \sum_{i=1}^{N} CT^{(i)}$ を計算し，データ分析者に送る．最後に，データ分析者が CT を復号し，$\boldsymbol{A} = \boldsymbol{H}^T\boldsymbol{H}$, $\boldsymbol{B} = \boldsymbol{H}^T\boldsymbol{Y}$ を得る．データ分析者は $\boldsymbol{A}$，$\boldsymbol{B}$ から Moore-Penrose の一般化逆行列を用いて，ELM のパラメータ $\boldsymbol{\beta}$ を計算する．

以下，より詳細な代理計算のプロセスを説明する．まずデータ提供者による前処理について述べる．全データ提供者は共通の入力層と隠れ層間の重みベクトル $\boldsymbol{a}$ とバイアス b を共有する．ここで提供者 i は自身の持っているデータ $(x^{(i)}, y^{(i)}) \in R^d \times R^K$ から式 (2) を用いて，実数値 $h^{(i)}$ を得る．さらに，データ提供者は $h_{r1}^{(i)}h_{r2}^{(i)}$，$h_r^{(i)}y_k^{(i)}$ の計算を前処理として行う．$h_{r1}^{(i)}h_{r2}^{(i)}$，$h_r^{(i)}y_k^{(i)}$ はそれぞれ $L(L+1)/2$ 個，KL 個ある．よって，データ提供者 i が計算すべき実数値の個数は

$$n_L = \frac{L(L+1)}{2} + KL \tag{14}$$

となる．これらの n_L 個の実数値をベクトル $\boldsymbol{D}^{(i)}$ で以下のように表す．

$$\begin{aligned}\boldsymbol{D}^{(i)} = \Big(&h_1^{(i)}h_1^{(i)}, \ldots, h_{r1}^{(i)}h_{r2}^{(i)}, \ldots, h_L^{(i)}h_L^{(i)}, \\ &h_1^{(i)}y_1^{(i)}, \ldots, h_r^{(i)}y_k^{(i)}, \ldots, h_L^{(i)}y_K^{(i)}\Big).\end{aligned} \tag{15}$$

データ提供者は上記のベクトル $\boldsymbol{D}^{(i)}$ を加法準同型暗号によって暗号化する．

$$CT^{(i)} = \mathsf{Enc}\left(pk,\ \boldsymbol{D}^{(i)}\right). \tag{16}$$

この具体的な暗号化については 5 章で述べる．データ提供者 i はこの暗号文 $CT^{(i)}$ を代理計算サーバに送る．

代理計算サーバは全データ提供者から $CT^{(i)}(1 \le i \le N)$ を受け取り，蓄積する．そして暗号文の加法演算を以下のように行う．

$$CT = \sum_{i=1}^{N} CT^{(i)} \tag{17}$$

この CT をデータ分析者に送る．

データ分析者は秘密鍵を使って，CT を復号する．Enc の加法準同型性により，データ分析者はすべての $1 \le r1, r2 \le L,\ 1 \le k \le K$ で，$A_{r1,\,r2} = \sum_{i=1}^{N} h_{r1}^{(i)}h_{r2}^{(i)}$, $B_{r,k} = \sum_{i=1}^{N} h_r^{(i)}y_k^{(i)}$ を得る．これにより，データ分析者は $\boldsymbol{A}$，$\boldsymbol{B}$ を得るため，式 (7) より推定したいパラメータ $\boldsymbol{\beta}$ を計算できる．

4.2 予測フェーズにおけるデータの安全性

パラメータ $\boldsymbol{\beta}$ を公開できると想定すれば，データ提供者は $\boldsymbol{\beta}$ を使ってラベルを予測できる．また $\boldsymbol{\beta}$ を使うことで安全な予測フェーズを構築できる．予測フェーズの流れを図 6 に示す．この予測フェーズの計算は代理計算サーバと

Require: $\boldsymbol{x}$, $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_L)$, $(b_1, \ldots, b_L)$, $\boldsymbol{\beta}$
Ensure: f_c^*
1: (C) $\boldsymbol{h} = (G(\boldsymbol{a_1},\ b_1,\ \boldsymbol{x}), \ldots, G(\boldsymbol{a_L},\ b_L,\ \boldsymbol{x}))$ を計算する．
2: (C) h_j を暗号化する．
3: (C) $\mathsf{Enc}(pk,\ h_j)$ を代理計算サーバへ送る．
4: (S) $\sum_{j=1}^{L} \mathsf{Enc}(pk,\ h_j)\hat{\beta}_j^{(k)}$ を計算する．
5: (S) $\sum_{j=1}^{L} \mathsf{Enc}(pk,\ h_j)\hat{\beta}_j^{(k)}$ をデータ分析者に送る．
6: (A) $\sum_{j=1}^{L} \mathsf{Enc}(pk,\ h_j)\hat{\beta}_j^{(k)}$ を復号し，$\boldsymbol{f}$ を得る．
7: (A) 式 (9) のように f_c^* を計算する．

図 6: PP-ELM の予測フェーズ
Fig. 6 Prediction phase of PP-ELM

表 1: PP-ELM におけるコストの概算
Table 1 Costs in PP-ELM

(a) Computational costs

	メモリ量	計算コスト
代理計算サーバ	$O(NL^2)$	$O(NL^2)$
データ提供者	N/A	$O(L^2)$
データ分析者	N/A	$O(L^3)$

(b) Communicational costs

	通信コスト
個々のデータ提供者 → サーバ	$O(L^2)$
サーバ → データ分析者	$O(L^2)$

データ分析者間で閉じており，データ提供者はデータを提供するのみである．この計算過程では，ラベルを予測するために $\boldsymbol{f} = \boldsymbol{h}\boldsymbol{\beta}$ を得る必要がある．したがって，それぞれの k に対して，$\sum_{j=1}^{L} h_j \beta_j^{(k)}$ を計算する必要がある．ここで $k = 1, \ldots, K$，$r = 1, \ldots, L$ において，

$$\boldsymbol{\beta} = (\boldsymbol{\beta}^{(1)},\ \ldots, \boldsymbol{\beta}^{(K)}),\ \boldsymbol{\beta}^{(k)T} = \left(\beta_1^{(k)}, \ldots, \beta_L^{(k)}\right)$$
$$\boldsymbol{h} = [h_1, \ldots, h_L],\ h_r = G(a_r,\ b_r,\ x)$$

である．

Enc が暗号文の加法演算のみを可能とする場合，$\beta_j^{(k)}$ を整数 $\hat{\beta}_j^{(k)}$ に変換する必要がある．そして，データ分析者は秘密鍵 sk を用いて，$\sum_{j=1}^{L} h_j \hat{\beta}_j^{(k)}$ を得る．さらに $\sum_{j=1}^{L} h_j \beta_j^{(k)}$ へと変換して，式 (9) の argmax を計算する．データ提供者がラベル y をデータ分析者に送る場合は，データ分析者によって精度を計算することができ，cross validation や grid search も可能となる．

4.3 提案手法におけるコスト

データ数を N，ELM の隠れ層のノードを L とすると，提案手法における計算コスト，通信コスト，メモリ量は表 1 のようになる．提案モデルでは，データ提供者とデータ分析者に比べて，代理計算サーバが重たい計算を担っている．

5. 具体的な暗号方式

前の章では，一般的な加法準同型暗号を使用して，どのように代理計算モデルを構築したかを説明した．この章では，learning with error (LWE) 問題をベースにした加法準同型暗号 [15] を用いて，具体的な暗号方式について説明する．

まず初めに，簡単に準同型暗号のスキームについて説明する．このスキームは鍵生成，暗号化，復号の三つのアルゴリズムに分けられる．

鍵生成：$\mathbb{Z}_p$ を $[-p/2, p/2)$ の整数の集合とし，$\mathbb{Z}_q$ も同様とする．安全性についてのパラメータ $n_{\mathrm{lwe}} \in \mathbb{Z}$ と平文ベクトルの次元 $\ell \in \mathbb{Z}$ に対して，分散を s とする離散ガウス分布から $R, S \in \mathbb{Z}^{n_{\mathrm{lwe}} \times \ell}$ をとり，$A \in \mathbb{Z}_q^{n_{\mathrm{lwe}} \times n_{\mathrm{lwe}}}$ をランダムに生成，$P = pR - AS \in \mathbb{Z}_q^{n_{\mathrm{lwe}} \times \ell}$ を計算する．秘密鍵 $sk = S$ と公開鍵 $pk = (A, P)$ が出力となる．

暗号化：平文 $m \in \mathbb{Z}_p^{\ell}$ に対して，

$$\mathsf{Enc}(pk, m) = e_1[A|P] + p[e_2|e_3] + [0_{n_{\mathrm{lwe}}}|m] \in \mathbb{Z}_q^{1 \times (n_{\mathrm{lwe}} + \ell)} \tag{18}$$

を計算する．ここで $e_1, e_2 \in \mathbb{Z}^{1 \times n_{\mathrm{lwe}}}$ と $e_3 \in \mathbb{Z}^{1 \times \ell}$ は分散 s の離散ガウス分布から生成したものであり，$[A|P]$ は行列の連結を意味する．

復号：暗号文 $c = \mathsf{Enc}(pk, m)$ を $(c_1, c_2)(c_1 \in \mathbb{Z}_q^{1 \times n_{\mathrm{lwe}}}, c_2 \in \mathbb{Z}_q^{1 \times \ell})$ のように分割し，

$$\mathsf{Dec}(sk, c) = (c_1 S + c_2 \bmod q) \bmod p \in \mathbb{Z}_p^{1 \times \ell}. \tag{19}$$

を計算する．

この暗号方式は明らかに加法準同型性を持つ．平文 m，m' において，

$$\begin{aligned}
&\mathsf{Enc}(pk, m) + \mathsf{Enc}(pk, m') \\
&= e_1[A|P] + p[e_2|e_3] + [0_{n_{\mathrm{lwe}}}|m] + \\
&\quad e_1'[A|P] + p[e_2'|e_3'] + [0_{n_{\mathrm{lwe}}}|m'] \\
&= (e_1 + e_1')[A|P] + p[e_2 + e_2'|e_3 + e_3'] + [0_{n_{\mathrm{lwe}}}|m + m']
\end{aligned}$$

となり，復号すると $m + m'$ が得られる．より詳細については [15] を参照のこと．

2.1 章で述べたように，この暗号方式の平文空間は $\mathbb{Z}_p^{\ell}$ であるため，暗号文における実数ベクトルに対する演算のためにエンコードを行う関数 $\mathbb{R}^{\ell} \to \mathbb{Z}_p^{\ell}$ が必要となる．本稿では，[16] のエンコードを用いた．エンコードを行う関数を $\mathbb{R} \to \mathbb{Z}_p$ とし，要素ごとにエンコードを行うことで，容易に $\mathbb{R}^{\ell} \to \mathbb{Z}_p^{\ell}$ を得ることができる．

$prec$ ビットの精度を持つ実数値 $a \in \mathbb{R}$ において，a は整数 $\lfloor a \cdot 2^{prec} \rfloor \in \mathbb{Z}$ により表現でき，充分大きな p に対して，

表 2: データセット
Table 2 Datasets

データセット	#データ数	#次元数	#クラス数
Glass	214	9	6
Digits	1797	64	10
Satellite	6435	36	6
Shuttle	43500	9	7

$\lfloor a \cdot 2^{prec} \rfloor \in \mathbb{Z}_p$ は $\lfloor a \cdot 2^{prec} \rfloor \in \mathbb{Z}$ と同一である．したがって，2^{prec} で割ることにより，$a \in \mathbb{R}$ へと変換できる．二つの平文が加算されたとき，値は $p/2$ よりも大きくなり，$[-p/2, p/2)$ の範囲外になる可能性がある．そのような場合，対応する実数値は壊れ，予期した値を得ることはできない．したがって，p は加算の結果がいつも $[-p/2, p/2)$ の範囲内となるように慎重に選ぶ必要がある．

PP-ELM においては，$p \geq 2N \cdot 2^{prec}$，$\ell = n_L$ とした．

6. 実験

6.1 精度評価

PP-ELM の精度評価として，UCI Machine Learning Repository [17] にて公開されている 3 つのデータセットと Python のオープンソース機械学習ライブラリである scikit-learn から 1 つのデータセットを利用する．UCI からは Glass Identification Data Set, Statlog (Landsat Satellite) Data Set, Statlog (Shuttle) Data Set を，scikit-learn からは digits dataset[18] を利用した．それぞれのデータセットに関する情報を表 2 に示す．また Glass, Satellite, Shuttle の三つのデータセットは平均が 0, 分散が 1 になるように正規化を行い，Digits については各属性を 0.0-1.0 の範囲に正規化した．本稿の実験では，三つの機械学習の手法：PP-ELM, プライバシー保護を可能とするロジスティック回帰の一対他法 (PP-Logistic ovr) [5], 一般的なロジスティック回帰の一対他法 (Logistic ovr) の性能比較を示す．PP-Logistic は本稿の提案モデルと同様の代理計算モデルを使用している．PP-ELM は式の変形や近似などを行っておらず，本来の ELM[12] と同様の性能が保証されているため，本稿では代理計算のプロセスや暗号化などを通さず，これらのアルゴリズムの性能を評価した．PP-ELM が充分に性能を出せるように，ELM の隠れ層のノード数 $L \in \{100, 200, 300\}$ の場合で PP-ELM の性能評価を行った．また 5-fold cross validation で性能の比較を行い，PP-ELM は入力層と隠れ層の間の重みのランダム性から精度もランダム性を持つため，5 回の実験で最も高い精度のものを選択した．結果を表 3 に示す．

表 3 の結果から，どのデータセットについても PP-ELM が高い分類精度を示していることがわかる．また PP-ELM の隠れ層のノード数が多いほど分類精度が高い傾向がある．PP-Logistic ovr は Logistic ovr よりも低い精度を示している．これは PP-Logistic では，準同型暗号を使用するために，シグモイド関数を二次式による近似をしているためである．対して，すでに述べているが，提案手法である PP-ELM では，式変形や近似を必要としないため，非線形分類器としての比較的高い精度を達成する．

6.2 速度評価

準同型暗号を使用した際のコストを測るために，実験で使用された中で最も大きなデータセットである Shuttle dataset に対して，5 章で説明した具体的な暗号方式で実験を行った．データセットに対して充分な 32-bit の精度を持つように，$p = 2^{49} + 1 \geq 2 \cdot 43500 \cdot 2^{32}$, $\ell = n_L \in \{5750, 21500, 47250\}$, $q = 2^{78}$, $n_{\mathrm{lwe}} = 2800$ とした．ここで提案モデルが正しく動作し，現在一般的に推奨される安全性レベルの 128 ビットセキュリティを持つようにパラメータを決めている．表 4 に具体的な暗号を用いた測定結果をまとめた．この測定結果は Core i7-77000K (4.20 GHz) のシングルスレッドで計測されたものである．プログラムは C++で記述し，gcc 6.3.0 でコンパイルを行った．表 4 に示されるように，暗号化と復号は $L = 300$ のとき，数百ミリ秒かかる．これはプライバシーを保護しつつ，高い精度を達成できることを加味すれば，許容できる範囲であると考える．

7. 結論

本稿では，加法準同型暗号を用いることで，安全に機械学習に必要な演算を外部委託できる ELM(privacy preserving ELM: PP-ELM) を提案した．データ提供者は ELM の隠れ層の出力を計算し，その乗算を前処理として行う．この前処理により，代理計算サーバは復号することなく，暗号文のまま，すべてのデータに対する隠れ層の出力の和の行列を計算できる．また提案する代理計算モデルでは，代理計算サーバがデータ数 N に依存する重い計算を担う．データ数がとても大きな場合，代理計算において，提案モデルの実用性は充分にあると考える．PP-ELM の精度を示すため，ロジスティック回帰の一対他法 (Logistic ovr) とプライバシー保護を可能とするロジスティック回帰の一対他法 (PP-Logistic ovr)[5] と PP-ELM で性能評価の実験を行った．この PP-Logistic は提案する代理計算モデルと同様のモデルで提案されているものである．PP-ELM は本稿の実験で使用したすべてのデータセットにおいて，他の二つの手法よりも高い精度を示した．PP-ELM は式変形や近似を必要としないため，既存研究ですでに示されているような比較的高い精度を示す．したがって，加法準同型暗号を用いた代理計算モデルにおいて，提案手法は単純な構造で非線形分類器としての高い精度を実現した．さらに，Shuttle dataset に対して，提案手法を具体的な暗号を使用して，実

表 3: 分類精度の比較
Table 3 Classification accuracy

データセット	PP-ELM			PP-Logistic ovr	Logistic ovr
	$L = 100$	$L = 200$	$L = 300$		
Glass	0.654 ± 0.045	0.675 ± 0.045	0.684 ± 0.089	0.596 ± 0.099	0.604 ± 0.070
Digits	0.921 ± 0.032	0.941 ± 0.019	0.965 ± 0.021	0.889 ± 0.037	0.925 ± 0.027
Satellite	0.850 ± 0.016	0.860 ± 0.008	0.875 ± 0.007	0.758 ± 0.019	0.827 ± 0.018
Shuttle	0.993 ± 0.001	0.996 ± 0.001	0.997 ± 0.001	0.873 ± 0.002	0.933 ± 0.002

表 4: LWE ベース暗号における時間コスト
Table 4 Timings of LWE-based encryption for Shuttle dataset (milliseconds)

アルゴリズム	$L = 100$	$L = 200$	$L = 300$
暗号化	33.073	95.301	196.524
復号	21.901	84.678	185.098
加算	0.008	0.025	0.048

験を行った．暗号化と復号に要する時間は，数百ミリ秒と許容できる範囲だと考える．

提案手法では信頼できるデータ分析者の存在を仮定する必要があった．今後の研究として，信頼すべきデータ分析者の存在がなくても成立するプライバシー保護を可能とした代理計算モデルの構築などが挙げられる．データ所有者以外は誰もデータを見ることができないため，より高度に機密な情報に対する機械学習の外部委託が可能となるだろう．

謝辞 本研究の成果は，JST CREST 研究領域「イノベーション創発に資する人工知能基盤技術の創出と統合化」の研究課題「複数組織データ利活用を促進するプライバシー保護データマイニング」(JPMJCR168A) により得られたものです．

参考文献

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” *In Theory of Cryptography Conference*, pp. 265–284, 2006.

[2] K. Chaudhuri and C. Monteleoni, “Privacy-preserving logistic regression,” in *Advances in NIPS 21, Proceedings of the 22nd Annual Conference on NIPS*, pp. 289–296, 2008.

[3] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, “Machine learning classification over encrypted data,” in *NDSS 2015*, 2015.

[4] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. E. Lauter, M. Naehrig, and J. Wernsing, “Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy,” in *ICML 2016*, pp. 201–210, 2016.

[5] Y. Aono, T. Hayashi, L. T. Phong, and L. Wang, “Privacy-preserving logistic regression with distributed data sources via homomorphic encryption,” *IEICE Transactions*, vol. 99-D, no. 8, pp. 2079–2089, 2016.

[6] R. L. Rivest, A. Shamir, and L. M. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[7] R. L. Rivest, L. Adleman, and M. L. Dertouzos, “On data banks and privacy homomorphisms,” *Foundations of Secure Computation, Academia Press*, pp. 169–179, 1978.

[8] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in *Advances in Cryptology - EUROCRYPT ’99, International Conference on the Theory and Application of Cryptographic Techniques*, pp. 223–238, 1999.

[9] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Trans. Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[10] H. Takabi, E. Hesamifard, and M. Ghasemi, “Privacy preserving multi-party machine learning with homomorphic encryption,” in *Private Multi-party Machine Learning – Workshop on 29th Annual Conference on NIPS*, 2016.

[11] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pp. 169–178, 2009.

[12] G. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 42, no. 2, pp. 513–529, 2012.

[13] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, 65, pp. 386–408, 1958.

[14] C. R. Rao and S. K. Mitra, “Generalized inverse of a matrix and its applications,” in *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics*. Berkeley, Calif.: University of California Press, pp. 601–620, 1972.

[15] Y. Aono, T. Hayashi, L. T. Phong, and L. Wang, “Efficient key-rotatable and security-updatable homomorphic encryption,” in *SCC@AsiaCCS 2017*, pp. 35–42, 2017.

[16] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, “Privacy-preserving deep learning: Revisited and enhanced,” in *ATIS 2017*, pp. 100–110, 2017.

[17] M. Lichman, “UCI machine learning repository.” http://archive.ics.uci.edu/ml, 2013.

[18] “scikit learn - the digit dataset.” http://scikit-learn.org/stable/auto_examples/datasets/plot_digits_last_image.html.