

SMPC を用いた秘密分散連合学習手法の実装と評価

An efficient method for restoring network traffic matrices using federated learning

ボドナル ソフィア[†]
Bodnar Sofia

水谷 后宏^{†,§}
Mizutani Kimihiro

1. はじめに

近年、AI 学習における課題の一つとして、十分な学習データの確保が挙げられる。この課題に対する解決手段として、複数の組織間で共同学習を行う連合学習が注目されている。連合学習では、データそのものを共有する必要がなく、各組織が保有する学習データをもとにローカルモデルを作成し、それらを一つのモデルに統合することで、複数組織のデータを反映した AI モデルを構築することができる。

しかし、一般的な連合学習では、各組織のローカルモデルを集約する中央サーバーが存在するため、中央集権的な構造とならざるを得ない。中央サーバーへの依存は、単一障害点の存在やサーバー運営組織への信頼の集中といった問題を生じさせる。そこで本研究では、この中央サーバーに依存しない分散的な構造を目指し、各組織のローカルモデルを秘匿した状態で分散的に集約する手法を検討する。

筆者らは前報 [1] において、この分散集約手法の基礎的な実装を行い、クライアント数を変化させた場合の精度およびオーバーヘッドを評価した。しかし、前報の実験条件には、精度低下の要因を切り分けられていないという課題があった。そこで本研究では、この課題を踏まえて実験条件を見直し、精度およびオーバーヘッドを改めて評価する。

2. 構成技術

本研究で用いる構成技術について述べる。提案手法の全体像を図 1 に示す。連合学習を中央サーバーに依存せずに実現する手段として、本研究では SMPC による秘密分散集約を採用した。また、その SMPC を実現するためのフレームワークとして CrypTen を用いた。

2.1 連合学習

連合学習とは、複数の組織や端末が保有するデータを直接共有することなく、それぞれのデータをもとに学習を行い、その結果得られたモデルを統合することで、あたかも全データを集約したかのような学習効果を得る分

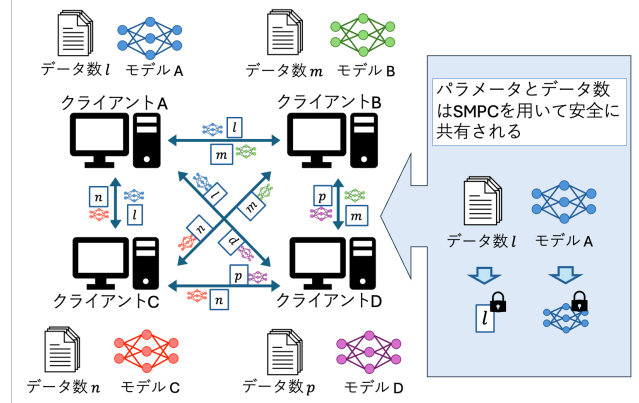


図 1: SMPC による秘密分散連合学習

散学習の枠組みである。一般的な連合学習は、中央サーバーと複数のクライアント（各組織や端末）から構成され、以下のような手順で進められる。

- 1. モデルの配布:** 中央サーバーが、学習対象となるモデルの初期パラメータを各クライアントに配布する。
- 2. ローカル学習:** 各クライアントは、配布されたモデルを自身が保有するローカルデータを用いて学習し、モデルパラメータを更新する。
- 3. モデルの送信:** 各クライアントは、更新後のモデルパラメータ（またはその勾配）を中央サーバーに送信する。データそのものは送信されないため、各組織の情報を秘匿した状態を保つことができる。
- 4. モデルの集約:** 中央サーバーは、各クライアントから受信したモデルパラメータを平均化などの手法により集約し、新たなグローバルモデルを生成する。
- 5. 繰り返し:** 中央サーバーは、生成したグローバルモデルを再び各クライアントに配布し、上記の手順を所定のラウンド数だけ繰り返すことで、モデルの精度を向上させる。

このように、連合学習ではデータそのものを共有することなく複数組織のデータを反映したモデルを構築できるという利点がある一方、モデルの集約を担う中央サーバーが必要となるため、中央集権的な構造にならざるを得ないという課題がある。

[†] 近畿大学大学院情報学研究科, Graduate School of Science and Engineering, Kindai University

[§] 近畿大学情報学研究所, Cyber Informatics Research Institute, Kindai University

2.2 SMPC

SMPC (Secure Multi-Party Computation: 秘密計算) とは、複数の参加者がそれぞれ秘密の入力値を保持している状況において、互いの入力値そのものを明かすことなく、それらを用いた計算結果のみを得ることができる技術である。連合学習における中央サーバーへの依存という課題を解消する手段として、本研究ではこの SMPC を応用し、各クライアントのモデルパラメータを秘匿したまま分散的に集約する。SMPC の代表的な実現方式の一つである秘密分散 (Secret Sharing) に基づく手法では、以下のような手順で秘匿計算が行われる。

1. **秘密分散:** 各参加者は、自身が保持する秘密の値を複数の断片 (シェア) に分割し、他の参加者にそれぞれ配布する。このとき、一部のシェアのみからは元の値を復元できないように分割される。
2. **分散状態での計算:** 各参加者は、自身が受け取ったシェアに対して、加算や乗算などの演算を行う。この計算は元の値を復元することなくシェアのまま実行される。
3. **結果の統合:** 各参加者が計算した結果のシェアを持ち寄り、それらを統合することで、元の秘密の値を明かすことなく計算結果のみを復元する。

このように、SMPC を用いることで、特定の中央サーバーに全参加者のデータやモデルパラメータを集約させることなく、参加者間の分散的な計算のみによって集約結果を得ることが可能となる。本研究では、この性質を利用し、連合学習における各クライアントのモデルパラメータを中央サーバーに依存せず秘匿した状態で集約する。

2.3 CrypTen

CrypTen [2][3] は、Facebook 社 (現 Meta 社) が開発した、PyTorch 上で SMPC を扱うためのオープンソースフレームワークである。テンソル演算をそのまま秘匿計算に置き換えられるインタフェースを備えており、通常の PyTorch のコードに近い形で SMPC に基づく秘匿計算を実装できる。本研究では、この CrypTen を用いて、各クライアントのモデルパラメータを秘密分散した状態で集約する処理を実装した。

3. 関連研究

竹田ら [4] は、連合学習における一般的なモデル集約手法である FedAvg [5] が、各クライアントのデータセットサイズをサーバーに開示する必要がある、これがプライバシー上の懸念となる点に着目した。FedAvg [5] では、データセットサイズの大きいクライアントのモデルほど大きな重みを与えることで精度を高めているが、この重

み付けにはサーバーが各クライアントのデータセットサイズを把握している必要があり、医療や金融など機微な情報を扱う領域ではこの情報開示自体がプライバシーリスクとなり得る。

そこで竹田ら [4] は、SMPC を用いて各クライアントのデータセットサイズを秘匿したまま比較し、データセットサイズに基づく順位のみをサーバーに送信する手法を提案した (図 2)。サーバーは、この順位の逆数に基づく重み (r_i/R 、ただし r_i はクライアント i の順位の逆数、 R は全クライアントの順位の逆数の総和) を用いてモデルを集約することで、データセットサイズそのものを開示することなく、データセットサイズの大きいクライアントのモデルをより重視した集約を実現している。また、この順位情報はクライアント選択にも応用可能であり、上位クライアントのモデルのみを選択的に統合することで、通信・計算コストの削減にもつながるとしている。SMPC の実装には CrypTen ライブラリが用いられている。

CIFAR-10 データセット [6] を用いた実験では、提案手法が FedAvg と同等の収束速度を示しつつ、86% の精度を達成し、FedAvg の 85% をわずかに上回ることが確認されている。また、クライアント選択の観点からは、強化学習に基づく選択手法と比較して、提案手法は選択に要する計算時間を大幅に削減できることが示された。

ただし、竹田ら [4] の手法はサーバーが各クライアントの順位情報を受け取り集約を行う構成であり、中央サーバーへの依存という構造自体は残されている。これに対し筆者らは前報 [1] において、この中央サーバーへの依存自体を解消することを目的とし、SMPC を用いてモデルパラメータを秘匿しながら分散的に集約する構成を提案した。

筆者らは前報 [1] において、この中央サーバーへの依存自体を解消することを目的とし、SMPC を用いてモデルパラメータを秘匿しながら分散的に集約する手法を提案した。具体的には、各クライアントがローカルデータを用いてモデルを学習した後、学習済みモデルとデータ数を参加クライアント数 n に応じたシェアに分割して秘密分散し、分散状態のまま重み付き平均を計算することで、中央サーバーを介さずにグローバルモデルを再構成する手法である。

前報 [1] の評価実験では、5, 10, 15 クライアントの環境において、CIFAR-10 データセットを用いて 10 エポックのローカル学習を 5 ラウンド繰り返し、SMPC による集約オーバーヘッドおよび精度を平文集約と比較した。その結果、集約に要する時間はそれぞれ 3.28 秒、10.40 秒、21.56 秒であり、SMPC の導入によるオーバーヘッドは管理可能な範囲に収まることが確認された。一方、精

度については SMPC を用いた場合に 74.31%, 72.79%, 48.09%, 平文を用いた場合に 84.90%, 72.60%, 52.20% となり, クライアント数の増加に伴って精度が低下する傾向が確認された.

しかし, 前報 [1] の実験ではエポック数を 10, ラウンド数を 5 に固定しており, また使用した CIFAR-10 データセットをクライアント数に応じて等分割していたため, 各クライアントが保有するデータ数がクライアント数に応じて変化する設定となっていた. そのため, クライアント数の増加に伴って確認された精度低下が, クライアント数の増加そのものに起因するものか, 各クライアントが保有するデータ数の減少に起因するものかを区別できていなかった. そこで本研究では, ラウンド数を十分に増やすとともに, クライアント数によらず各クライアントが保有するデータ数を固定した条件のもとで実験を行い, SMPC を用いた分散的なモデル集約における精度およびオーバーヘッドを改めて評価する.

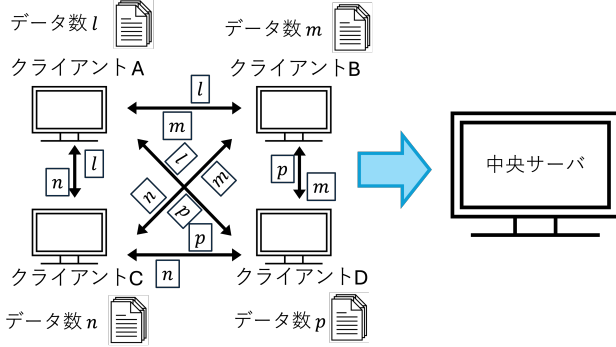


図 2: 先行研究の手法

4. 実験

本章では, 本研究で行った実験の内容および結果について述べる. 実験条件を表 1 に示す.

表 1: 実験条件

項目	内容
データセット	CIFAR-10
クライアント数	5, 10, 15
エポック数	10
ラウンド数	20
データ分割条件	非固定 (等分割) / 固定 (一定数)
集約手法	平文集約 / SMPC (CrypTen)

4.1 精度の推移

非固定条件および固定条件のもとで, 平文集約および SMPC による集約それぞれについて, クライアント数

(5, 10, 15) ごとの精度 (precision) の推移を比較した結果を図 3 (非固定条件), 図 4 (固定条件) に示す. 各図の (a) は平文集約, (b) は SMPC による集約の結果である.

非固定条件 (図 3) では, クライアント数が増加するにつれて収束が遅くなり, 最終的な精度も低下する傾向が, 平文集約・SMPC の両方で確認された. 一方, 固定条件 (図 4) では, クライアント数の違いによる精度の差が大きく縮小し, 5, 10, 15 クライアントいずれの場合もほぼ同様の速度で収束し, 最終的な精度もほぼ一致する結果となった. このことから, 非固定条件において確認されたクライアント数増加に伴う精度低下は, クライアント数の増加そのものよりも, クライアントあたりのデータ数の減少に起因する部分が多いことが示唆された.

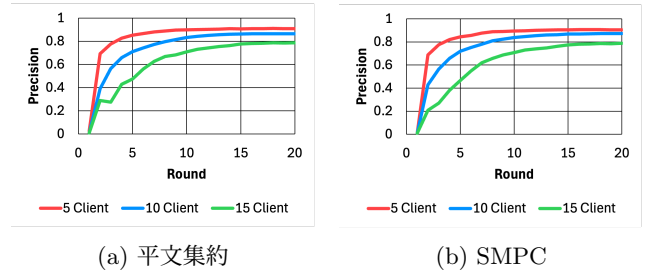


図 3: 非固定条件における精度の推移

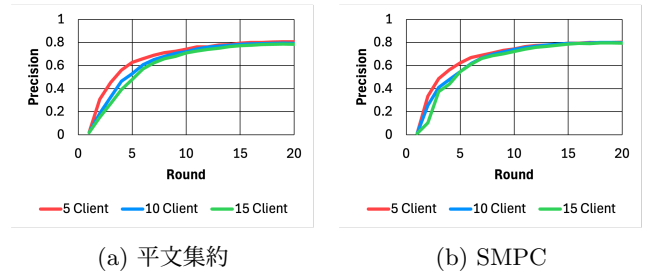


図 4: 固定条件における精度の推移

4.2 固定条件における集約時間のオーバーヘッド

固定条件のもとで, 各クライアント数について, 1 ラウンドあたりの集約に要する時間を, 平文集約と SMPC とで比較した結果を表 2 に示す. SMPC による集約時間は平文集約と比較して大幅に増加しており, クライアント数の増加に伴ってオーバーヘッドも増加する傾向が確認された. この要因については, 4 章にて考察する.

5. 考察

5.1 精度への影響

4.1 節の結果より, 非固定条件において確認されたクライアント数増加に伴う精度低下は, クライアント数の増

表 2: 固定条件における集約時間の比較

クライアント数	5	10	15
平文 (s)	0.10	0.66	0.43
SMPC (s)	3.68	10.62	21.52
オーバーヘッド (s)	3.58	9.95	21.10

加そのものよりも、各クライアントが保有するデータ数の減少に起因する部分が多いことが示された。固定条件下では、SMPC による集約と平文集約との間で精度に大きな差は見られず、SMPC の導入自体が精度に与える影響は限定的であると考えられる。

5.2 オーバーヘッドの要因

表 2 で確認された SMPC による集約時間の増加の要因について整理する。

CrypTen における秘密分散では、各シェアの乱数成分の生成に事前に共有されたシードを用いるため、秘密分散そのものや、加算・スカラー倍といった線形演算の実行時には、参加者間の通信は発生しない。本研究における集約処理は主に重み付き平均であり、これらの演算はローカルで完結する。

一方、オーバーヘッドの主要因となるのは、集約後のグローバルモデルを復号する際の通信である。参加者 (rank) 数を R 、復号対象となるデータの総量を N [Byte] とすると、1 つの rank から見た送信量はおよそ

$$2 \times \frac{R-1}{R} \times N \quad (1)$$

となる。この式が示すように、復号時の通信量は参加者数 R の増加に伴って増加し、 $R \rightarrow \infty$ の極限では $2N$ に漸近する。

また、SMPC では秘密分散の性質上、平文で扱うデータ型 (`torch.float32`) とは異なり、整数型である `torch.int64` としてデータを扱う必要があり、これによってもデータ量が増加する。

表 2 に示したクライアント数の増加に伴うオーバーヘッドの増加傾向は、復号時の通信量の増加およびデータ型の違いによるデータ量の増加によって説明できると考えられる。

6. 結論・今後の課題

本研究では、連合学習における中央サーバーへの依存を解消するため、SMPC を用いてモデルパラメータを秘匿しながら分散的に集約する手法について、クライアント数 (5, 10, 15) を変化させた場合の精度およびオーバーヘッドを評価した。前報 [1] の課題を踏まえ、集約ラウンド数を増やすとともに、クライアント数によらず各

クライアントが保有するデータ数を固定した条件のもとで実験を行った結果、クライアント数増加に伴う精度低下は、クライアント数の増加そのものよりも、クライアントあたりのデータ数の減少に起因する部分が多いことが明らかになった。また、SMPC によるオーバーヘッドは、主にグローバルモデルの復号時における通信に起因することを整理した。

今後の課題として、本研究では CrypTen のデフォルトの通信方式をそのまま用いたが、より通信量の少ない、あるいは安定した通信方式が存在するかについては検討できていない。そのため、通信方式の選択がオーバーヘッドに与える影響について、今後検証する必要がある。

謝辞

本研究は JSPS 科研費 JP23K16876 の助成を受けたものである。

参考文献

- [1] S. Bodnar and K. Mizutani, “Privacy-Preserving Decentralized Model Aggregation in Federated Learning Systems,” Proc. International Conference on Emerging Technologies for Communications 2025, Session P2, P2-29, Nov. 2025., DOI: 10.34385/proc.94.P2-29
- [2] B. Knott *et al.*, “CrypTen: Secure Multi-Party Computation Meets Machine Learning,” Proc. 35th Conf. on Neural Information Processing Systems, no. 379, pp. 4961–4973, 2021.
- [3] Facebook AI Research, “CrypTen,” <https://github.com/facebookresearch/CrypTen>, accessed Dec. 2025.
- [4] Y. Takeda and K. Mizutani, “Privacy-preserving Ranking-based Model Integration for Federated Learning,” *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 20, no. 11, 2025.
- [5] H. B. McMahan *et al.*, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” Proc. 20th Int. Conf. on Artificial Intelligence and Statistics (AISTATS), JMLR: W&CP, vol. 54, 2017.
- [6] A. Krizhevsky and G. Hinton, “Learning Multiple Layers of Features from Tiny Images,” Technical Report, University of Toronto, 2009.