クライアントの行動に基づく ビザンチン耐性のある連合学習の監視メカニズム

橋本 俊甫[1,a)]　田中 俊昭[1]　栗原 淳[1]

概要：分散型の機械学習手法である連合学習では，クライアントが自身のデータセットを保持したままローカルで学習し，サーバに学習結果の更新情報を送ることでグローバルモデルを共同で構築する．このとき，信頼のできないクライアント（ビザンチンクライアント）が存在し，それらが偽の学習結果を送信することで，グローバルモデルの信頼性や性能を脅かす可能性がある（ビザンチン攻撃）．本稿では，ビザンチン攻撃に耐性のある連合学習を実現するために，クライアントの行動に基づく監視メカニズムを提案する．提案手法は，クライアントが送信する更新情報から時間的な一貫性や空間的な類似性を評価するために，収束，重みの変動，空間的分布の 3 つの観点を監視することで，クライアントの異常な行動を検出する．このとき，提案手法はより現実的な環境を想定し，チーフクライアントの存在や中央サーバで性能を測るデータセットを不要とする．提案手法を用いることで，偽装攻撃，シビル攻撃，およびデータポイズニングを組み合わせたビザンチン攻撃を検出できることを，実験により明らかにする．さらに，ビザンチンクライアントが存在しない場合と同程度の精度を持つグローバルモデルを構築できることを示す．

キーワード：連合学習，ビザンチン攻撃，ビザンチンフォールトトレランス，監視メカニズム

A Monitoring Mechanism to realize the Byzantine Resilience in Federated Learning via Client Tracking

Shunsuke Hashimoto[1,a)]　Toshiaki Tanaka[1]　Jun Kurihara[1]

Abstract: Federated learning (FL) is a privacy-aware decentralized machine learning approach. In FL, clients perform local training on their datasets and send the model updates to the central server to build the global model. In FL environments, there may exist malicious clients, called Byzantine clients, that aims to corrupt the reliability and the performance of the global model. In this paper, we propose a monitoring mechanism based on client tracking to realize the Byzantine resilience in FL. The proposed method detects anomalous client behavior by monitoring three aspects: convergence, weight variation and spatial distribution, in order to assess temporal consistency and spatial similarity from the updates sent by the clients. The proposed method assumes a more realistic environment and does not require the presence of a chief client or a dataset to measure performance on the central server. Experimental results demonstrates that our method can correctly detect Byzantine attacks combined with Pretence, Sybil and data poisoning attacks and maintain the accuracy of the global model by eliminating them from the FL process.

Keywords: Federated learning, Byzantine attack, Byzantine fault tolerance, monitoring mechanism

1. はじめに

様々なデバイスやアプリケーションから生成される膨大なデータを機械学習で活用するには，プライバシに大きな課題がある．従来の中央集権型の機械学習手法では，複数のデバイスや地理的に離れた組織からデータを中央サーバに一元的に集約する必要があるこのため，データ漏洩や不正アクセスの懸念が生じる．さらに，医療や金融，モバイル端末のデータには，法的および倫理的制約により，容易に中央サーバに集約できない機微な情報が含まれている場合がある．

連合学習 [1–3] は，各クライアントが自身の所有するデータをローカルで学習し，学習して得られるモデルの更新情

[1] 兵庫県立大学大学院情報科学研究科
Graduate School of Information Science, University of Hyogo
[a)] ad23f052@guh.u-hyogo.ac.jp

報のみを中央サーバに共有することで，上記の課題を解決する．すなわち，データをクライアント自身が保持することで，プライバシ保護を実現する．このため連合学習は，中央サーバでのデータの集中管理が難しい場合においても，機械学習の恩恵を享受できる手法として注目されている．

1.1 背景

連合学習では，クライアントが分散化されている環境が前提となる．このような分散システムにおけるセキュリティ上の脅威の 1 つに，ビザンチン攻撃がある．この攻撃は，正当な処理を妨害することを目的として，一部のノードが意図的に不正確な情報を他のノードへ送信することを指す．ビザンチン攻撃を受けたシステムは，システムの信頼性や性能が低下するため，効果的な対策が求められる．なお，無応答なノードは明示的に観測可能であるため，本稿ではビザンチン攻撃の対象外とする．

連合学習におけるビザンチン攻撃 [4–6] では，連合学習に参加する信頼のできないクライアント（ビザンチンクライアント）が，意図的に不正確なモデルの更新情報を送信する．これにより，学習プロセスに悪影響を与え，連合学習で作られるモデルの信頼性や性能を劣化させることが，ビザンチンクライアントの目的である．近年の研究では，ビザンチン攻撃を防御し，連合学習のビザンチン耐性を向上させるための手法が提案されてきた（e.g. [4,7,8]）．しかし既存の手法は，特定の仮定の下にビザンチン耐性を担保しており，その仮定に従わないビザンチン攻撃の防御は難しい．このため，あらゆるビザンチン攻撃に対し，包括的に対応可能な防御手法を実現することが，現行の連合学習の大きな課題の 1 つである．

1.2 貢献

上記の課題を解決するため，本稿では，「クライアントの行動に基づく監視メカニズム」を提案する．提案手法は，各クライアントが送信するモデルの更新情報を監視し，クライアントの異常な行動を検出することにより，連合学習のビザンチン耐性を向上させる．

本研究の主な貢献を以下に示す．

(1) 連合学習において，ビザンチン攻撃を検出するための監視メカニズムを提案する．提案手法は，時間的・空間的な観点を組み合わせた監視手法で，多層的な防御機構を有し，既存の集約手法に干渉することなく既存の連合学習を拡張する．このとき，提案手法はより現実的な環境を想定し，チーフクライアントや中央サーバで性能を測るデータセットを不要とする．

(2) 画像分類タスクにおいて，複数の攻撃シナリオでの評価実験を通じて，ビザンチン攻撃によるグローバルモデルの精度の劣化を防ぐことを実験的に明らかにする．

1.3 構成

本稿の残りの部分は以下のように構成されている：本稿の構成は以下のとおり．第 2 節では関連研究を紹介する．第 3 節では，問題の定式化を行い，提案する監視メカニズムを説明する．第 4 節において，提案手法の有効性を実験的に評価する．第 5 節では，実験の結果から，提案手法の有効性を議論する．最後に，第 6 節では本稿の結論と今後の課題を述べる．

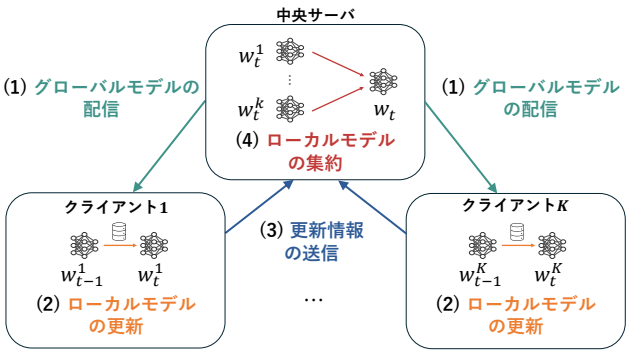


図 1: 連合学習の学習プロセス

Fig. 1 The learning process of federated learning.

表 1: 連合学習で利用するパラメタ

Table 1 Parameters used in federated learning.

k	クライアントを一意に識別する番号
K	クライアント数
t	ラウンド番号
T	最大ラウンド数
P_k	クライアント k が所有するデータセット
w_t	ラウンド t におけるグローバルモデル
w_t^k	ラウンド t におけるクライアント k のローカルモデル

2. 準備

2.1 連合学習

連合学習のプロセスを図 1 に示す．以下のような学習プロセスによって，プライバシを保護しつつ，分散デバイス上で協調的なモデル学習を実現できる．また，表 1 に連合学習で利用するパラメタを与える．なお，本稿では，同じ特徴空間を共有するがサンプルが異なる水平連合学習 [3] を仮定する．

(1) **グローバルモデルの配信**：中央サーバはグローバルモデル w_{t-1} を全てのクライアントに配布する．初回のラウンドでは，ランダムに初期化したグローバルモデル w_0 を配信する．ここで，グローバルモデルは共同で作られるモデルである．

(2) **ローカルモデルの更新**：学習に参加する各クライアントは，自身のローカルデータセットを使用してグローバルモデル w_{t-1} を学習することで，w_t^k を得る．このとき得られるモデルをローカルモデルと呼ぶ．

(3) **更新情報の送信**：各クライアントは学習後の更新情報（ローカルモデルの重みパラメータや勾配など）を中央サーバに送信する．本稿では更新情報をモデルの重みパラメータとし，各クライアントはローカルモデルの更新情報を中央サーバに送信する．

(4) **ローカルモデルの集約**：中央サーバは，クライアントから受け取った更新情報を集約して，新しいグローバルモデル w_t を得る．

(5) **反復**：学習プロセスの (1) から (4) を 1 ラウンドとして，所定のラウンド数 T まで繰り返す．

2.2 関連研究

2.2.1 連合学習におけるビザンチン攻撃

Shi ら [5] は，学習データに基づく攻撃とパラメータに基づく攻撃の 2 つに分類している．以下にそれらを解説する．また，それらの攻撃手法を実行するシナリオをその次に与える．

2.2.1.1 学習データに基づく攻撃（データポイズニング）

この攻撃は，各クライアントがローカルでデータを収集・保管している段階，またはその後の前処理・学習段階で行われる．この種の攻撃は，不正確なデータや誤ったラベルのデータを学習することでモデルの精度を下げることを目的とする．

学習データに基づく攻撃の代表的なものに，**ラベルフリップ攻撃** [5,6,9] がある．ラベルフリップ攻撃は，学習データの特定のクラスのラベルを他のクラスのラベルに変更した誤ったラベルで学習させる．この攻撃は，攻撃者は学習データに関する知識を必要とせずに実行できる．

通常，ラベルフリップ攻撃はグローバルモデルの収束を妨害することを目的として，全クラスに対して実施する．しかし攻撃者は，ビザンチン攻撃を検出されるリスクを最小限に抑えるため，全体的なモデルの性能に大きな影響を与えず，特定のクラスに対してのみ攻撃することも考えられる．特定のクラスに限定することで，他のクラスに対しては正常な挙動をするため，モデル全体の性能に対する大きな異常が現れにくく，攻撃が隠蔽されやすくなる．本稿では，全クラスに対するラベルフリップ攻撃と分けるために，この攻撃を**標的型ラベルフリップ攻撃**と呼ぶ．

2.2.1.2 パラメータに基づく攻撃（モデルポイズニング）

この攻撃は，クライアントがローカルで学習した後の段階で行われる．この種の攻撃は，学習後のローカルモデルを改竄することで，グローバルモデルを直接的に制御することを目的とする．代表的な手法として，サインフリップ攻撃 [10]，IPM [11]，ALIE [12] が挙げられる．

2.2.1.3 ビザンチン攻撃の攻撃シナリオ

偽装攻撃 [13] は，悪意のあるクライアントが正常なクライアントを装って攻撃する手法である．この攻撃は，検出を回避するために，クライアントの行動を偽装することに焦点を当てている．偽装攻撃をするクライアントは，最初の一定のラウンドでは良性のクライアントのように振舞いながら，学習プロセスの後半でビザンチン攻撃を開始する．

また，**シビル攻撃** [4,7] は，高度なビザンチン攻撃のシナリオを実現する手段の 1 つである．この攻撃は，複数の悪意のあるクライアントが連携して行われる攻撃である．悪性のクライアントが結託して攻撃すれば，単独のクライアントによる攻撃よりも効果的に連合学習システムを破壊できることが知られている．

2.2.2 ビザンチン耐性のある集約手法

Blanchard ら [4] によって，連合学習の典型的な集約手法である FedAvg [1] は，単一のビザンチンクライアントの存在を許容できないことを明らかにされた．この問題に対し，ビザンチンクライアントが存在する場合でも安定して学習を進められるような，ビザンチン耐性のある集約手法が提案されている（e.g. [4,7,8]）．Shi ら [5] は，ビザンチン攻撃を防御するうえで依存する原理によって集約手法を距離，性能，統計，最適化に基づく 4 つの手法に分類している．一方で，これに当てはまらないクライアントの行動に基づく手法 [13] が提案されている．

2.2.2.1 距離に基づく手法

本手法は，各クライアントから送信される更新間の距離を比較することで，異常な更新情報や性能の悪い更新情報を識別し，除去する手法である．代表的な手法として，Multi-Krum [4]，FoolsGold [7] が挙げられる．

2.2.2.2 性能に基づく手法

本手法は，中央サーバが所有する準検証データセット [8] におけるパフォーマンスに基づいて悪意のある更新情報をフィルタリングする．代表的な手法として，Zeno [14]，FLTrust [15] が挙げられる．

2.2.2.3 統計に基づく手法

本手法は，更新情報から算出する統計的な情報を利用して異常を検出し，ビザンチン攻撃を緩和する．代表的な手法として，Trimmed-mean [16]，GeoMed [17]，Bulyan [18] が挙げられる．

2.2.2.4 最適化に基づく手法

本手法は，グローバルモデルのビザンチン耐性を向上させるために，学習で利用する目的関数とは別の目的関数を最適化する．代表的な手法として，RSA [10] が挙げられる．

2.2.2.5 クライアントの行動に基づく手法

本手法は，各クライアントの更新情報を分析することでビザンチンクライアントを検出するアプローチである．Mallah ら [13] は，連合学習プロセスにおける学習中のクライアントの行動を考慮した戦略として，学習中のクライアントの行動を時間的に評価してビザンチン攻撃を検出し，排除するアルゴリズムを提案した．具体的にはローカルモデルを用いて，グローバルモデルへの収束，更新間の角度距離，性能変化を評価している．

Mallah らの手法は，距離に基づく手法と性能に基づく手法に時系列情報を組み合わせている．距離に基づく手法は，悪性のクライアントの更新が良性のクライアントの更新から逸脱しているという仮定に依存しているため，明らかな異常を検出するには適している一方で，そうでない微細な異常の場合には性能が低下する問題がある [5]．性能に基づく手法は，モデルの性能を直接測ることができるため，時系列情報を組み合わせることで，過去のラウンドの性能と比較して性能が向上しないクライアントを排除できる．これにより，微細な異常を検出できる可能性がある．しかし，中央サーバが準検証データセットを所有することは，プライバシ漏洩の懸念があるため，現実の環境では中央サーバにデータを用意することは難しい．したがって，中央サーバの準検証データセットに依存しない防御手法が必要である．

表 2: ビザンチン攻撃シナリオの場合分け
Table 2 Classification of Byzantine attack scenarios.

攻撃パターン	偽装攻撃	シビル攻撃	データポイズニング
1	なし	なし	なし
2	あり	なし	ラベルフリップ
3	あり	なし	標的型ラベルフリップ
4	なし	あり	ラベルフリップ
5	なし	あり	標的型ラベルフリップ
6	あり	あり	ラベルフリップ
7	あり	あり	標的型ラベルフリップ

3. 提案手法

本節では問題の定式化を行い，ビザンチン耐性のある連合学習の監視メカニズムを提案する．

3.1 問題の定式化

連合学習は，1 つの中央サーバと複数のクライアントから構成される．このとき，クライアントには，以下の 2 種類が存在する．

- **良性のクライアント**: ローカルモデルのパラメータを更新情報として，常に正直に中央サーバへ送る．良性のクライアントの目的は，グローバルモデルの精度を向上させるために，自身のデータセットを活用して連合学習に貢献することである．
- **悪性のクライアント**: 誤ったラベルのデータを学習したローカルモデルのパラメータを更新情報として，中央サーバへ送ることができる．悪性のクライアントの目的は，グローバルモデルの収束を妨害することと，または特定のクラスのみ予測精度を下げることである．また，悪性のクライアントとビザンチンクライアントを互換的に表現する．

本稿では，上記の良性・悪性のクライアントが混在する連合学習を想定する．このとき，悪性のクライアントは，連合学習の学習プロセスへ第 2.2.1 節で述べた攻撃および攻撃シナリオを実行する．本稿で扱うビザンチン攻撃のシナリオは，以下に定義する偽装攻撃およびシビル攻撃の 2 種類とする．

- **偽装攻撃**：最大ラウンド数 T の半分のラウンドまでは良性のクライアントとして行動し，半分を超えると悪性のクライアントとして行動する．
- **シビル攻撃**：学習に参加するクライアントのうち，過半数を超えるクライアントが結託し，同じタイミングで同じ攻撃を実施する．悪性のクライアントが過半数を超える場合，グローバルモデルに与える影響は著しく強くなるため，過半数を超える場合にシビル攻撃と呼ぶ．

本稿では，偽装攻撃とシビル攻撃を組み合わせ，データポイズニングによって連合学習を攻撃する攻撃者を仮定する．このとき，データポイズニング攻撃に関して，攻撃者は，第 2.2.1.1 節で与えたラベルフリップ攻撃，または標的型ラベルフリップ攻撃を用いて攻撃する．このときの攻撃パターンは，表 2 のように場合分けされる．

Algorithm 1 提案手法を適用した連合学習フロー（T: ラウンド数，K: クライアント数，w_t^k: ラウンド t におけるクライアント k のローカルモデル，w_t: ラウンド t におけるグローバルモデル）

Input: Initial global model w_0
Output: Aggregated global model w_T
$\mathcal{B} \leftarrow \{\}$
for $t \leftarrow 1$ to T **do**
  Execute `FederatedLearningProcess()` at each client
  for $k \leftarrow 1$ to K **do**
    if k in $\mathcal{B}$ **then**
      continue
    end if
    `is_reliable` $\leftarrow$ `TimeSeriesConvergence`(w_t^k, w_{t-1})
    if `is_reliable` is **False then**
      Insert k into $\mathcal{B}$
      continue
    end if
    `is_reliable` $\leftarrow$ `TimeSeriesSimilarity`(w_t^k, w_{t-1}^k)
    if `is_reliable` is **False then**
      Insert k into $\mathcal{B}$
    end if
  end for
  $\mathcal{T} \leftarrow$ `TrustScoredClustering`$(\{1, \ldots, K\} \setminus \mathcal{B})$
  $w_t \leftarrow$ `Aggregate`$(\{w_t^k | k \in \mathcal{T}\})$
end for
return w_T

各パターンにおいて，グローバルモデルの学習結果の精度が，悪性クライアントが存在しない場合と比べて同等であることを担保できるとき，その連合学習は当該のパターンに対する「ビザンチン耐性」を有すると呼ぶ．全パターン網羅的に，ビザンチン耐性を実現する手法を与えるのが，提案手法の目的である．

3.2 提案手法の概要

本稿では，連合学習におけるビザンチンクライアントの行動を特徴づけるために，時系列情報を活用する．クライアントから送信される更新情報を時系列的に評価することで，クライアントの行動の一貫性を評価する．このため，既存の防御手法とは異なり，提案手法は集約時の防御に依存しない．提案手法を適用した連合学習全体のフロー概要を，Algorithm 1 で与える．Algorithm 1 は，`TimeSeriesConvergence`, `TimeSeriesSimilarity` および `TrustScoredClustering` の 3 つの新たなビザンチンクライアントの検出手法を従来の連合学習フローに対して統合する．これにより，ビザンチンクライアントと判定されたクライアントを除くローカルモデル出力のみを用いて，グローバルモデルを学習する．それぞれの要素技術は，次の機能を持つ．

(1) **モデルの収束監視**(`TimeSeriesConvergence`(w_t^k, w_{t-1})): ラウンド t におけるクライアント k のローカルモデル w_t^k と，その前のラウンド $t-1$ におけるグローバルモデル w_{t-1} を入力とし，両者のユークリッド距離の差分の変化量を計測し，学習の収束状況を評価する．本

手法により，偽装攻撃が検知できることを示す．

(2) **特定の層の重みの変動監視**（TimeSeriesSimilarity(w_t^k, w_{t-1}^k)）：ラウンド t，$t-1$ におけるクライアント k のローカルモデル w_t^k，w_{t-1}^k を入力とし，特定のノードに繋がるエッジの重みの時間的な変化量を計測することで，異常な更新を検知する．本手法により，偽装攻撃を検知できることを示す．また，一部のクラスを対象とするラベルフリップ攻撃にも対応可能であることを示す．

(3) **信頼スコアに基づくビザンチンクライアントの集団の検出**（TrustScoredClustering）：ローカルモデルにクラスタリングを適用し，信頼スコアの総和に基づいてビザンチンクライアントの集団を特定する．本手法により，シビル攻撃を検知できることを示す．

「(1) モデルの収束監視」は，文献 [13] の手法と同一である．一方，悪性のクライアントの行動を精密に監視するために，「(2) 特定の層の重みの変動監視」を新たに考案した．さらに，集団の逸脱行動を監視することを目的として「(3) 信頼スコアに基づくビザンチンクライアントの集団の検出」も新たに考案している．これらの 3 つの要素技術を組み合わせることにより，異常検知の柔軟性を高め，多様な攻撃シナリオに対応できるようにした．

Algorithm 1 では，ラウンド t において，クライアントごとに TimeSeriesConvergence および TimeSeriesSimilarity を実行し，そのクライアントがビザンチンクライアントか否かを判定する．続いて，ここまでで良性クライアントであると判定したクライアント集合に対し，TrustScoredClustering を実行し，疑わしいクライアントをさらに排除する．最後に，排除せず残った良性のクライアント集合の持つローカルモデルを集約し，ラウンド t におけるグローバルモデルを導出する．

上記のように，提案手法は，既存の連合学習の枠組みに 3 つの要素技術からなる監視メカニズムを追加することで，ビザンチン攻撃に対する多層防御手法を実現する．また，提案手法はチーフクライアントや準検証データセットを必要とせず，従来の集約手法に干渉しないため，現実的で広範な適用が期待できる．次節では，提案手法の核となる 3 つの要素技術について詳細に説明する．

3.3 提案手法の詳細

提案手法の監視メカニズムでは，監視の対象を学習の進行に伴い送信される各クライアントのローカルモデルとし，時間的な一貫性や空間的な類似性を評価する．具体的には，各クライアント k がラウンド t で送信するローカルモデルのパラメータ w_t^k を収集する．このモデルパラメータは，クライアント k が所有するデータセットによる学習結果が反映されているため，ビザンチン攻撃が行われているかを判断する指標とする．

3.3.1 モデルの収束監視

本要素技術は，クライアントから送信されるローカルモデルとグローバルモデルとのユークリッド距離の時間的変動を監視する．このことにより，クラインアントの学習過程が一貫しているか否かを確認でき，ビザンチン攻撃がなされているかを検出する．

学習過程の一貫性の判定のために，各クライアントのモデルパラメータとグローバルモデルのモデルパラメータの比較をする．まず，文献 [13] と同様に，各ラウンドにおけるローカルモデル w_t^k とその前ラウンドのグローバルモデル w_{t-1} との間のユークリッド距離 D_t^k を $D_t^k = ||w_t^k - w_{t-1}||$ と定義する．

次に，ラウンド $t-1, t$ におけるユークリッド距離の変動を線形回帰で評価する．これにより，距離の増減の速度を評価する．具体的には，以下の傾き G を計算する．

$$G = \frac{D_t^k - D_{t-1}^k}{t-(t-1)} = D_t^k - D_{t-1}^k. \tag{1}$$

この傾き G が所定のしきい値 α を超える場合，クライアント k の学習は一貫して進行していないと判断され，ビザンチン攻撃をしている可能性があると判断する．

3.3.2 特定の層の重みの変動監視

本要素技術は，各クライアントのモデルパラメータのうちの，特定のノードへ繋がるエッジに着目する．このことにより，クライアントの更新情報を詳細に分析し，特定の層の重み変動の監視を実現することで，偽装攻撃および標的型ラベルフリップ攻撃の検知を可能とする．

従来の距離に基づく手法 [4, 19] では，モデルのすべてのパラメータの変化量を総和して評価する．これは明らかに逸脱した異常を検出するには適しているが，全体に対する影響が小さい異常を検出するには困難な場合がある．したがって，従来手法は通常のラベルフリップ攻撃の検出に適しているものの，特定のラベルのみ反転される標的型ラベルフリップ攻撃は適していない．これを鑑み，特定の層の重みの変動監視では，ノードごとの重みの変化に注目し，当該の層の任意のノードで，ラウンド間での時系列的なコサイン類似度が著しく低い場合，異常と判定する．これにより，標的型ラベルフリップ攻撃を検出する．

具体的には次のとおり．ラウンド t におけるクライアント k の特定のノード i の重みベクトルを $v_t^{k,i}$，ラウンド $t-1$ における重みベクトルを $v_{t-1}^{k,i}$ とし，これらのベクトル間のコサイン類似度を $\mathrm{Sim}(v_t^{k,i}, v_{t-1}^{k,i})$ とする．

本稿では，特に，出力確率に直接関係する出力層を対象とし，出力層の重みの変動を監視する．ここで，出力層の全ノード，つまり予測するクラスの数についてコサイン類似度を計算し，コサイン類似度が所定のしきい値 β を下回る場合に層の重みが変動していると判断する．もし 1 つでも異常なノードが検出されると，そのクライアントは異常であると見なす．

3.3.3 信頼スコアに基づくビザンチンクライアントの集団の検出

本要素技術は，以下のステップによって，ビザンチンクライアントの集団を特定することで，シビル攻撃を検知する．

(1) 各ラウンドにおいて，各クライアントのモデルパラメータを t-SNE を用いて 2 次元の特徴空間に写像する．

(2) 得られた特徴量に対して k-means 法を適用し，クライ

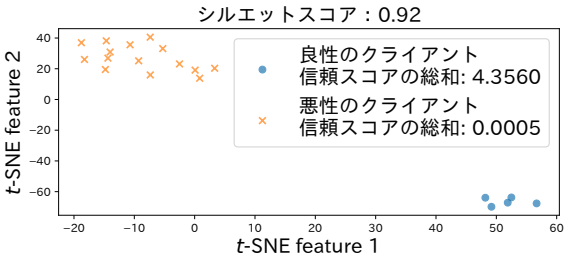


図 2: t-SNE で次元削減した更新情報の可視化
Fig. 2 t-SNE visualization of model updates.

アントを 2 つのクラスタに分類する．このとき，クラスタリング結果のシルエットスコアを算出し，所定の閾値 γ を下回る場合はクラスタの品質が低いと判断し，検出を中断して次ラウンドへと進む．

(3) クラスタリングが成功した場合，各クライアントにおいて，文献 [7] で使われる計算式で信頼スコアを計算する．各クライアントに対する信頼スコアの初期値は $R_k = 1$ である．各クライアント k について，他の全クライアントと比較し，コサイン類似度を用いて信頼スコア

$$R_k = \prod_{l \neq k, l \in \{1,\ldots,K\}} \left(1 - \mathrm{Sim}\left(\mathrm{Flat}(v_t^k), \mathrm{Flat}(v_t^l)\right)\right), \quad (2)$$

を更新する．ここで，v_t^k はラウンド t におけるクライアント k のモデルパラメータを表す行列である．また，$\mathrm{Flat}(v_t^k)$ は，行列 v_t^k の各行を一列に並べた 1 次元ベクトルを表す．これにより，クライアント k の信頼スコア R_k が，他のクライアントとの類似性に反比例するように調整される．

(4) 最後に，全クライアントの信頼スコアを次式で正規化する．

$$\hat{R}_k = \frac{R_k}{\max\{R_1, \ldots, R_K\}}. \quad (3)$$

正規化により，すべてのクライアントの信頼スコアが 0 から 1 の範囲に収まり，各クライアントにおける相対的な信頼度が得られる．

(5) 計算された信頼スコアを用いて，クラスタごとの信頼スコアの総和

$$R_{\mathrm{sum}} = \sum_{k \in C_c} \hat{R}_k, \quad (4)$$

を算出する．ここで，C_c はクラスタ c に属するクライアントの集合である．

(6) 総和 R_{sum} が低いほうのクラスタを異常なクラスタとみなす．このとき，異常なクラスタと決定されたクラスタに属するクライアントの影響を減少させるため，該当のクライアントを排除する．

シビル攻撃への対策として，本提案手法の妥当性は次の例から示すことができる．図 2 は，ラベルフリップ攻撃を実行したラウンドでの各クライアントのローカルモデルを t-SNE で次元削減した特徴量を可視化している．悪性のクライアントは良性のクライアントから乖離しており，

表 3: シミュレーション評価環境およびパラメータ
Table 3 Environments and parameters in our experiments

Hyperparameters

local batch size	10
learning rate	0.01
# of local epochs	10
optimizer	SGD
SGD momentum	0.5

Settings of federated learning

# of clients K	20
# of rounds T	10
aggregation	FedAvg
dataset	MNIST [20]
distribution	IID [1]

Parameters of the proposed scheme

threshold α (Sect. 3.3.1)	2.0
threshold β (Sect. 3.3.2)	0.85
threshold γ (Sect. 3.3.3)	0.85

k-means 法を適用したところ，シルエットスコアは 0.92 であった．さらに，各クラスタにおける信頼スコアの総和を算出したところ，悪性のクライアントのクラスタのほうが低い値になっていることがわかる．これにより，t-SNE で得られる特徴量を観察することで，ビザンチン攻撃を受けているクラスタを判別可能であると考えられる．

4. 実験

4.1 実験設定

4.1.1 連合学習の設計と実装

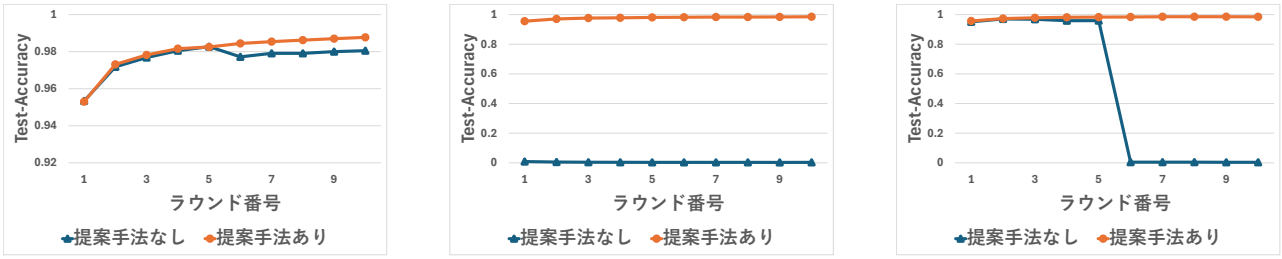
表 2 で与えた攻撃シナリオにおいて，提案手法がビザンチン耐性を有することを，シミュレーション評価によって検証する．表 3 に，本実験の環境およびパラメータを示す．データは IID [1] 条件下で各クライアントにランダムに分割され，各クライアントは同等のデータ量・分布を持つ．ここで，各クライアントは，2 つの畳み込み層を持つシンプルな CNN モデルを使用し，各々が所有するデータセットで学習する．連合学習では，中央サーバが各ラウンドにおける更新を FedAvg によって平均化して集約し，グローバルモデルを更新する．このプロセスを最大ラウンド数 10 まで繰り返し，ビザンチン攻撃が存在する中でのモデルの収束を評価する．なお，本実験の環境および提案手法は，文献 [21] を参考に PyTorch [22] を用いて実装した．

4.1.2 ビザンチン攻撃の設定

本実験では，ビザンチン攻撃として，以下のようにシビル攻撃，偽装攻撃，ラベルフリップ攻撃，および標的型ラベルフリップ攻撃を設定する．

- **シビル攻撃**：クライアント数 $K = 20$ のうち，15 クライアント (75%) を攻撃者とする（シビル攻撃「あり」），あるいは 1 クライアントのみを攻撃者とする（シビル攻撃「なし」）．
- **偽装攻撃**：最大ラウンド数 $T = 10$ において，第 5 ラウンド以降から攻撃を開始（偽装攻撃「あり」）するか，第 1 ラウンドから開始（偽装攻撃「なし」）する．
- **ラベルフリップ攻撃**：ラベル l を $L - l - 1$ へ反転させる（L：クラス数，$L = 10$，$0 \leq l \leq L - 1$）．
- **標的型ラベルフリップ攻撃**：クラス 5 のラベルをクラス 3 に反転させる．

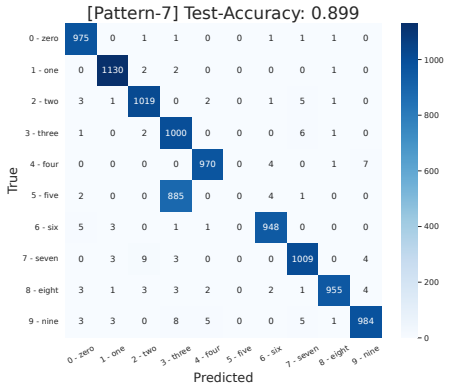
これらの設定の下，各攻撃シナリオにおける提案手法の有無による精度（Test-Accuracy）を，ビザンチンクライアン



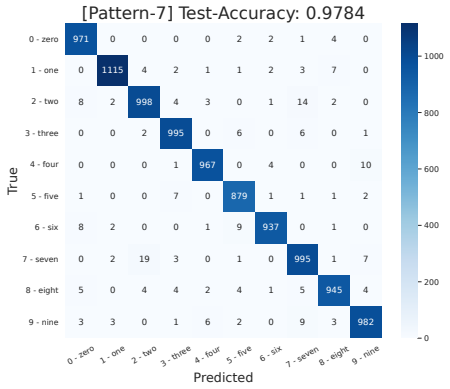
(a) パターン 2：偽装攻撃　(b) パターン 4：シビル攻撃　(c) パターン 6：偽装，シビル攻撃

図 3: ラベルフリップ攻撃における提案手法の有無による精度の比較

Fig. 3 Comparison of Test-Accuracy with and without the proposed method in Label Flip Attack.



(a) パターン 7：提案手法なし



(b) パターン 7：提案手法あり

図 4: 標的型ラベルフリップ攻撃のヒートマップ

Fig. 4 Heatmap of Targeted Label Flip Attack.

トが存在しない理想的な状態で測定した結果と比較する．

4.2 結果

本節では，表 2 で与えた各攻撃シナリオにおける提案手法の有無に基づく実験結果を述べる．すべてのパターンに対して，提案手法なしの実験と提案手法ありの実験を行い，それぞれの結果を比較した．パターン 2，4，6 の実験結果は図 3 に，パターン 7 の結果は図 4 に示す．一方，パターン 3，5 の実験結果は，後述するように，パターン 7 と同様の結果であることから，紙面の都合上割愛する．

提案手法なしの場合：理想的なシナリオであるパターン 1 では，攻撃が行われないため，グローバルモデルは 0.9869 という高精度を達成した．一方，パターン 2（図 3a）およびパターン 6（図 3c）の偽装攻撃のシナリオでは，ビザンチン攻撃が第 5 ラウンド以降に開始された結果，グローバルモデルの精度が徐々に低下した．さらに，パターン 4（図 3b）とパターン 6（図 3c）で行われたシビル攻撃では，攻撃者がクライアントの過半数を占めるため，グローバルモデルの精度は著しく低くなった．また，パターン 7（図 4a）の標的型ラベルフリップ攻撃では，Test-Accuracy は 0.899 と見かけ上は高い精度を示したものの，クラス 5 のデータがすべてクラス 3 に誤分類される結果となった．同様に，偽装攻撃のシナリオを含むパターン 5 においても，誤分類される結果となっている．

提案手法ありの場合：提案手法を適用したことで，どの攻撃シナリオにおいてもグローバルモデルの精度が著しく改善された．パターン 2（図 3a）およびパターン 6（図 3c）の偽装攻撃シナリオでは，提案手法により第 5 ラウンド以降もグローバルモデルの精度が維持され，パターン 1 と同等の精度が達成された．また，パターン 4（図 3b）とパターン 6（図 3c）のシビル攻撃シナリオでは，提案手法が攻撃を受けたクライアントを適切に検出・排除することで，グローバルモデルの破壊を防ぐことができた．さらに，パターン 7（図 4b）の標的型ラベルフリップ攻撃においても，提案手法の適用によりクラス 5 のデータが正しく分類され，パターン 1 の実験と同程度の 0.9784 と精度を維持できた．同様に，パターン 5 においても，本手法を用いることにより，精度を維持しつつ，正しく分類できることを確認している．なお，パターン 3 については，単一の不正クライアントによる標的型ラベルフリップ攻撃であるため，グローバルモデルに対する変化は微小であり，実質攻撃の影響はないことが判明している．

提案手法の有無による結果の差異から，提案手法は表 2 で与えた攻撃シナリオに対してビザンチン耐性を有することが確認された．

5. 議論

パターン 6 の偽装攻撃のシナリオでは，ビザンチンクライアントの割合が 50%を超えているにもかかわらず，提案手法によりグローバルモデルの精度がパターン 1 の精度と同等になった．これは，提案手法がクライアントごとに監視しているからである．したがって，提案手法は偽装攻撃において，ビザンチンクライアントの割合には依存しないことがわかる．

一方，パターン 4 のシビル攻撃のシナリオでは，ビザンチンクライアントは最初のラウンドから一貫して攻撃を続けるため，行動の一貫性を評価する監視手法ではシビル攻撃の検知は困難である．そこで，学習に参加したすべてのクライアントを対象に，第 3.3.3 節の手法を用いて，空間的な類似性を評価することにより，パターン 4 のシビル攻撃のシナリオも対応可能であることがわかる．これにより，提案手法はビザンチンクライアントが結託した場合でも検出が可能となる．

本稿では，ビザンチン攻撃をしていると判断されたクライアントは連合学習から排除し，以降のラウンドには参加させないという方法を採用した．このことにより，良性のクライアントが排除される場合も存在した．一律に排除する代わりに，検出結果を集約時のクライアントの重み付けアルゴリズムへ反映する方法も考えられる．これにより，悪性のクライアントと判断されたクライアントを即座に排除するのではなく，信頼度に応じて重み付けすることで，攻撃の影響を軽減しつつ，より柔軟な対応が可能となる．また，本稿での実験は，集約時に平均化処理を行っている．このため，既存のビザンチン耐性のある集約手法に差し替えれば，本稿では扱っていない攻撃シナリオにおいても有効で，より包括的な防御手法になると考えられる．

本稿では IID 条件下を仮定しているため，非 IID 条件下では信頼スコアに基づくビザンチンクライアントの集団の検出アルゴリズムが機能しない可能性がある．これは，非 IID 条件下では良性のクライアント間の更新で大きな違いが発生するため，良性・悪性の 2 つのクラスタへの分離が難しくなると考えられる．したがって，より現実的な環境を想定し，非 IID 条件下でも有効な防御手法が求められる．

6. 結論と今後の課題

本稿では，ビザンチン耐性のある連合学習を実現するために，クライアントの行動に基づく監視メカニズムを提案した．提案手法は，モデルの収束監視，特定の層の重みの変動監視，信頼スコアに基づくビザンチンクライアントの集団の検出の 3 つの要素技術から構成され，多層的な防御機構を有する．既存の手法とは異なり，提案手法は準検証データセットを必要としない手法であるため，現実的で広範な適用が期待できる．画像分類タスクにおいて，IID の条件下で異なる攻撃シナリオを実験し，提案手法の有効性を確認した．今後の研究として，モデルポイズニング攻撃での検証や，非 IID 条件下においても機能する防御手法の設計が挙げられる．

謝辞　本研究は JSPS 科研費 (JP22K11994)，NICT 委託研究 (22401)，KDDI 財団の調査研究助成を受けたものです．

参考文献

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. AISTATS’17*, 2017, pp. 1273–1282.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[4] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, “Machine learning with adversaries: Byzantine tolerant gradient descent,” in *Proc. NeurIPS’17*, 2017, pp. 118–128.

[5] J. Shi, W. Wan, S. Hu, J. Lu, and L. Y. Zhang, “Challenges and approaches for mitigating Byzantine attacks in federated learning,” in *Proc. IEEE TrustCom’22*, 2022, pp. 139–146.

[6] M. Fang, X. Cao, J. Jia, and N. Gong, “Local model poisoning attacks to Byzantine-robust federated learning,” in *Proc. USENIX Security’20*, 2020, pp. 1605–1622.

[7] C. Fung, C. J. Yoon, and I. Beschastnikh, “Mitigating sybils in federated learning poisoning,” *arXiv preprint arXiv:1808.04866*, 2018.

[8] X. Pan, M. Zhang, D. Wu, Q. Xiao, S. Ji, and Z. Yang, “Justinian’s GAAvernor: Robust distributed learning with gradient aggregation agent,” in *USENIX Security’20*, 2020, pp. 1641–1658.

[9] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, “Data poisoning attacks against federated learning systems,” in *Proc. ESORICS’20, Part I*, 2020, pp. 480–501.

[10] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, “RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets,” in *Proc. AAAI’19*, vol. 33, no. 01, 2019, pp. 1544–1551.

[11] C. Xie, O. Koyejo, and I. Gupta, “Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation,” in *Proc. UAI’20*, 2020, pp. 261–270.

[12] G. Baruch, M. Baruch, and Y. Goldberg, “A little is enough: Circumventing defenses for distributed learning,” in *Proc. NeurIPS’19*, vol. 32, 2019.

[13] R. A. Mallah, D. López, G. Badu-Marfo, and B. Farooq, “Untargeted poisoning attack detection in federated learning via behavior attestational,” *IEEE Access*, vol. 11, pp. 125 064–125 079, 2023.

[14] C. Xie, S. Koyejo, and I. Gupta, “Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance,” in *Proc. ICML’19*, 2019, pp. 6893–6901.

[15] X. Cao, M. Fang, J. Liu, and N. Z. Gong, “FLTrust: Byzantine-robust federated learning via trust bootstrapping,” *arXiv preprint arXiv:2012.13995*, 2020.

[16] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, “Byzantine-robust distributed learning: Towards optimal statistical rates,” in *Proc. ICML’18*, 2018, pp. 5650–5659.

[17] C. Xie, O. Koyejo, and I. Gupta, “Generalized Byzantine-tolerant SGD,” *arXiv preprint arXiv:1802.10116*, 2018.

[18] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, “The hidden vulnerability of distributed learning in Byzantium,” in *Proc. ICML’18*, 2018, pp. 3521–3530.

[19] Q. Xia, Z. Tao, Z. Hao, and Q. Li, “FABA: an algorithm for fast aggregation against Byzantine attacks in distributed neural networks,” in *Proc. IJCAI’19*, 2019, pp. 4824–4830.

[20] Y. LeCun, C. Cortes, and C. J. Burges, “MNIST handwritten digit database,” http://yann.lecun.com/exdb/mnist/, 2010.

[21] A. R. Jadhav, “Federated-learning (PyTorch),” https://github.com/AshwinRJ/Federated-Learning-PyTorch, 2021.

[22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in PyTorch,” in *Proc. NeurIPS’17 Autodiff Workshop*, 2017.