Federated Learning を用いた ローンのデフォルト予測に関する一検討

A Study on Federated Learning-based Loan Default Prediction

近藤 浩史[1*]　森 毅[1]　長尾 卓司[2]

Hirofumi Kondo[1], Takeshi Mori[1], Takuji Nagao[2]

[1]株式会社日本総合研究所 先端技術ラボ
[1] Advanced Technology Laboratory, The Japan Research Institute, Limited
[2]株式会社三井住友銀行 データマネジメント部
[2] Data Management Department, Sumitomo Mitsui Banking Corporation

Abstract: Federated Learning はデータのプライバシーを保護しながら，複数の計算参加者が協調して機械学習モデルを訓練する技術である. 各計算参加者が保有する素のデータをお互いに公開せず, 素のデータを秘匿した状態でモデルを学習できることが特徴であり, 金融などのプライバシー保護が特に求められる領域での活用が期待されている．Federated Learning を金融領域に適用した研究は既に行われている一方で, 実際の金融機関のデータを用いて有用性を検証した研究は少ない．本稿では金融機関の実業務データを使用し，ローンのデフォルト予測に Federated Learning を適用した結果を報告する.

1. はじめに

スマートフォンの普及や IoT などの技術的進展に伴い，ビジネスにおいて利活用可能なデータ量が増加している．並行して，AI 技術も日々進化し，収集したデータの利活用に対して期待が高まっている．

一方，データのプライバシー保護に対する社会的な関心も高まっている．個人データ保護やプライバシー保護に関する規則・規制は強化される傾向にあり，今後もこの動きは拡大すると考えられる．したがって，収集したデータのプライバシーをどのように保護しながら利活用するのかが課題となる．

データのプライバシーを保護した状態で，複数の計算参加者（以降，クライアントと呼ぶ）が協調して機械学習モデルを訓練する技術として Federated Learning（以降，FL と略す）[1]が注目されている．従来，複数クライアントが参加して，ある機械学習モデルを訓練する場合，各クライアントが持つ素データを一つのサーバに集約することが一般的であった. FL では, 素データを他クライアントに公開せず, 訓練中のモデルパラメータのみを共有することにより, 素データを秘匿した状態でモデルを訓練できる.

FL はデータのプライバシー保護が特に求められる金融領域において活用が試みられている．例えば情報処理通信機構では，日本国内の銀行 5 行と連携して，不正送金の検知モデルの開発に取り組んでいる（*1）. 他にもローンのデフォルト予測やクレジットカードの不正検知に FL を適用した研究[2,3]がある．これらは，訓練および評価データとして一般公開されている疑似データ（*2）を用いており，実業務データを用いて評価されていない.

本研究では，国内の金融機関の実業務データを使用し，ローンのデフォルト予測に FL を適用した結果を報告する．加えて，クライアントが保有するデータ量，特徴量の数，使用するモデルの観点でいくつかの条件を設定し，どのような場合に FL によってモデルの性能向上が得られるかを検証した.

結果として，訓練データを十分に保有しないクライアントにおいて性能向上のメリットを得やすいことが分かった．また，多層パーセプトロンと比較して，勾配ブースティング決定木をベースとしたモデルが安定した性能を発揮することが分かった.

*連絡先：kondo.hirofumi@jri.co.jp

(*1) https://www.nict.go.jp/press/2020/05/19-1.html
(*2) 元データが分からないように PCA などを用いて何らかの変換がなされたデータ

2. 問題設定

2.1. 適用するユースケース

本研究ではカードローンのデフォルト予測に対して FL を適用する．なお，本稿においてデフォルトとは，約定返済の不履行に至ること（金融機関が保証会社に代位弁済請求すること）を意味し，以降，代位弁済請求と記載する．与信審査を高度化する観点で，金融機関が連携する意義のあるユースケースとして設定した．

また本研究では，訓練データを多く保有する銀行と，訓練データの保有量が少ない銀行が連携してモデルを訓練する場合を想定する．すなわち，顧客基盤の大きな銀行と，顧客基盤の小さい銀行の各 1 行が連携する場合を考える．したがって，本研究で FL に参加するクライアント数は 2 である．

2.2. 使用するモデル

MLP（Multi-Layer Perceptron:多層パーセプトロン）と勾配ブースティング決定木（Gradient Boosting Decision Tree: GBDT）のモデルを使用した． FL によって各モデルを学習する具体的な手法については 3 節で述べる．

FL は適用するタスクやクライアントが保有する特徴量により，使用する手法が異なり，主に Horizontal-FL，Vertical-FL，Transfer-FL の 3 つの手法に大別される[4]. 本研究では Horizontal-FL を用いる．Horizontal-FL はクライアントが保有するデータのうち，全クライアントが共通して保有する特徴量のみを用いて FL を実行する．各クライアントが個別でモデルを訓練する場合と比較すると, FL によりモデルが学習するデータのレコード数が増加し，結果的にモデルの性能向上が期待できる手法である．

2.3. 使用するデータ

日本国内のある大手金融機関 X が実業務で使用するデータを使用した．ある時点の顧客の属性情報（例：年齢，年収など）と入出金情報（例：口座残高など）を結合して説明変数（計約 180 個）とし，ある時点から一定期間後に当該顧客のローンが代位弁済請求に至るかどうかを目的変数とするデータである．なお，本稿では代位弁済請求に至るレコードを正例，そうではないレコードを負例と呼ぶ．

本研究では，説明変数を全て使用せず，ランダムに 50 個または 100 個の変数をサンプリングして用いた．これは Horizontal-FL を実行する際に，各クライアントが保有する全特徴量が一致することは希である（クライアントごとにデータの保有状況が異なる）ことを検証条件に反映するためである．

2.1 節で述べた通り, 大きな顧客基盤を持つ金融機関（A 銀行）と，顧客基盤の小さい銀行（4 行，B, C，D，E 銀行）が連携するケースを想定する．A から E 銀行のデータは，大手金融機関 X のデータを都道府県の単位で分割・サンプリングして作成した（表 1）．A 銀行は関東圏（東京，神奈川，千葉，茨城，埼玉，群馬，栃木）の 7 都県からサンプリングし，B から E 銀行のデータは関東圏以外のそれぞれある県のデータをサンプリングして作成した．

A 銀行は顧客基盤が大きく，B から E 銀行と比較して正例・負例ともに多数のレコードを持っている．そのため, 負例をアンダーサンプリングすることで, 正例・負例データの偏りを小さくすることが可能である前提とした. したがって, A 銀行の正例比率は, 実際の正例比率よりも高く設定されている．

B から E 銀行のデータは，実際の正例比率を参考にサンプリングした．A 銀行と B 銀行の件数は同数だが，B 銀行の顧客基盤の小ささを表現するため，正例数は A 銀行よりも少なく設定した．

表 1：作成したデータ

名称	件数	正例数	正例比率
A 銀行	40,000	2,000	5 %
B 銀行	40,000	600	1.5%
C 銀行	15,000	150	1 %
D 銀行	8,000	80	1 %
E 銀行	4,000	40	1 %

3. FL の学習手法

3.1. 秘密計算との組み合わせ

FL では, 中央サーバとクライアント間で送受信されるデータを用いて，あるクライアントの素データの推測・復元を試みる攻撃が可能である．その防御策として秘密計算や差分プライバシーを組み合わせて FL を実行する研究[5,6,7]が存在する．

我々は予備実験として，秘密計算（準同型暗号）と組み合わせた FL を実装・評価した．この結果，秘密計算の有無により最終的に訓練されるモデルの性能に大きな影響がないことを確認した. そのため, 本研究では秘密計算と組み合わせずに FL を実行した場合を考える．

3.2. MLP を使用する場合

MLP を用いて FL を実行する最もシンプルな手法として FederatedAveraging（または FedAvg）[8]があ

る．FedAvg の概要は以下の通り（図 1）．

① 中央サーバがモデル（重み）を初期化する．
② 中央サーバは全クライアント集合から学習に参加するクライアント集合 s をランダムに抽出する．
③ 中央サーバはクライアント集合 s にモデルの重みを送信する．
④ 集合 s に属するクライアントは予め決めたバッチサイズとエポック数だけローカルでモデルを学習する．
⑤ 集合 s に属するクライアントは学習したモデルの重みと学習したデータのサンプル数をサーバに送信する．
⑥ 中央サーバは各クライアントから受信したモデルの重みを用いて，重みの加重平均を計算する．
⑦ 重みの加重平均を新しいモデルの重みとする．
⑧ 2〜7 を 1 ラウンドとし，モデルのパラメータが収束するまでラウンドを繰り返す．

なお，本研究で参加するクライアント数は 2 であるため，手順 2 においては全クライアントが抽出される．

3. 3. GBDT を使用する場合

本研究では Yang[9]の手法をベースとして勾配ブースティング決定木（GBDT）の FL を実装した（以降，GBDT-FL と記載する）．GBDT-FL は scikit-learn（*3）に実装された HistGradientBoostingClassifier（*4）を拡張する形で実装した．

GBDT-FL では，ある決定木のあるノードの分割点を決定する際に，中央サーバが各クライアントから

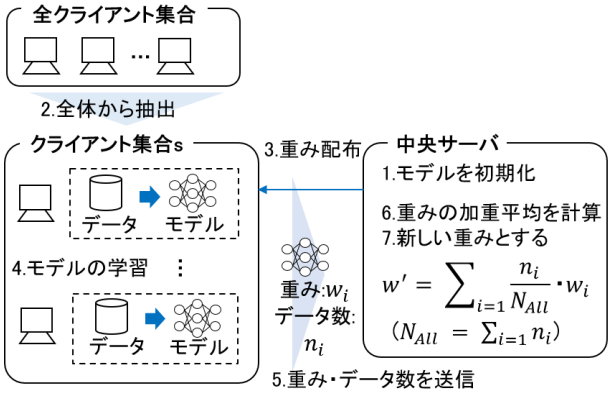
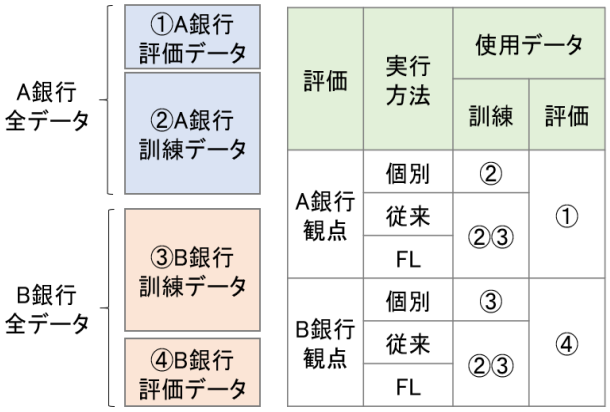


図 1：FedAvg の処理の流れ

(*3) https://scikit-learn.org/stable/
(*4) https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html
(*5) https://lightgbm.readthedocs.io/en/latest



評価	実行方法	使用データ	
		訓練	評価
A銀行観点	個別	②	①
	従来	②③	
	FL		
B銀行観点	個別	③	④
	従来	②③	
	FL		

図 2：訓練・評価で使用するデータの分割方法

分割点のノードに含まれる全データ・全特徴量の Gradient および Hessian のヒストグラムを受信する．中央サーバは受信したヒストグラムを集約し，集約したヒストグラムから最適な分割点を決定する．中央サーバは決定した分割点の情報のみを各クライアントに送信する．これらの手順を繰り返すことでモデルを学習する．

4. 実験

4. 1. 実験観点

本実験の観点を整理すると以下 4 つが存在する．これらのうち実行可能な組み合わせに対して網羅的に検証し，得られたモデルの性能を比較する．

① 連携する銀行（B から E 銀行の 4 種類）
② 実行方法（個別/従来/FL の計 3 種類）
③ 使用するモデル（MLP/GBDT の 2 種類）
④ 使用する特徴量の数（50/100 の 2 種類）

ここで実行方法の「個別」および「従来」とは，FL との比較ためにベースラインとして用いた手法のことを指す．実行方法の「個別」とは，各クライアントが自身の持つデータだけを用いて個別にモデルを訓練し，評価する方法である．実行方法の「従来」とは，各クライアントが保有する全データを一つのサーバに集約してモデルを訓練し，評価する方法である．したがって，実行方法の観点では，(1) 個別と比較して従来および FL の性能がどの程度向上するか，(2) FL の性能が従来の精度とどの程度近くなるか，という 2 点に着目して評価することとなる．

なお，使用するモデルが GBDT の場合，ベースラインモデルの実装として LightGBM（*5）を採用した．LightGBM はテーブル形式のデータに対して一般に高い性能を示すため，ベースラインの性能を測る観点で最適と考えたためである．

表 2：100 個の特徴量を用いた場合のモデル評価結果（値:ROC-AUC）

モデル		実行方法	評価データ：A 銀行				評価データ：B～E 銀行のいずれか			
			連携する銀行				連携する銀行			
			A+B	A+C	A+D	A+E	A+B	A+C	A+D	A+E
MLP		個別	0.846	0.846	0.846	0.847	0.789	0.827	0.828	0.797
		従来	0.845	0.849	0.848	0.847	0.838	0.930	0.878	0.822
		FL	0.851	0.850	0.849	0.849	0.837	0.908	0.749	0.792
GBDT	LightGBM	個別	0.872	0.876	0.878	0.874	0.805	0.889	0.810	0.784
		従来	0.869	0.876	0.875	0.876	0.862	0.938	0.868	0.816
	GBDT-FL	FL	0.861	0.868	0.868	0.866	0.845	0.938	0.860	0.820

表 3：50 個の特徴量を用いた場合のモデル評価結果（値:ROC-AUC）

モデル		実行方法	評価データ：A 銀行				評価データ：B～E 銀行いずれか			
			連携する銀行				連携する銀行			
			A+B	A+C	A+D	A+E	A+B	A+C	A+D	A+E
MLP		個別	0.805	0.804	0.804	0.806	0.755	0.815	0.794	0.815
		従来	0.804	0.808	0.807	0.806	0.784	0.829	0.880	0.819
		FL	0.809	0.806	0.808	0.805	0.785	0.789	0.815	0.754
GBDT	LightGBM	個別	0.809	0.808	0.811	0.811	0.751	0.787	0.783	0.690
		従来	0.809	0.811	0.814	0.812	0.787	0.853	0.875	0.822
	GBDT-FL	FL	0.806	0.811	0.812	0.811	0.785	0.859	0.869	0.812

また，使用するモデルが MLP の場合，データを標準化してモデルへの入力とした．実行方法が個別または FL の場合はクライアントごとに訓練データを標準化し，実行方法が従来の場合は全訓練データで標準化した．

4.2. 評価方法

本実験の評価指標は ROC-AUC を用いる．乱数の影響で性能がばらつくことから，モデルの訓練・評価をそれぞれ 10 回実行し，平均の ROC-AUC を評価に用いる．

評価は A 銀行の評価データで評価した場合と，B から E 銀行のいずれかの評価データで評価した場合をそれぞれ記載する．例として A 銀行と B 銀行が連携する場合の訓練・評価データの分割方法を図 2 に示す．実行方法が個別の場合は，自クライアントの訓練データのみを用いる．実行方法が従来および FL の場合は，他クライアントの訓練データも合わせて用いる．評価では，自クライアントの評価データのみを用いる．

評価データとしては，表 1 のデータを銀行ごとに訓練データ：評価データが 8:2 の比率となるように分割して用いた．このとき，分割後の訓練データと評価データの正例比率が同一となるように調整した．データの分割は実験全体を通じて同一とする．

4.3. 結果

表 2 に 100 個の特徴量を用いた場合のモデルの評価結果を記載する．

A 銀行の評価データを用いた場合，モデルの観点では，GBDT を使用する場合の性能が良い．また，若干のばらつきはあるものの，使用するモデルが同じであれば，実行方法によらず同程度の性能が得られている．つまり，A 銀行の視点では，個別でモデルを訓練しても，他銀行と連携してモデルを訓練しても性能が変わらない．そのため，他銀行と連携するメリットが無いと言える．

次に B から E 銀行の評価データを用いてモデルを評価した場合を確認する．実行方法の個別と従来を比較すると，モデルに依らず従来のほうが性能は高く，B から E 銀行の視点では A 銀行と連携するメリットが得られている．実行方法の従来と FL を比較すると，GBDT の場合には同程度の性能が得られている．そのため，プライバシーを保護したままモデルを構築できる点で，FL を利用するメリットがある．一方で MLP の場合は，B 銀行を除いて FL の場合に性能が低下している．この点については次節にて原因を考察する．

表 3 に 50 個の特徴量を用いた場合のモデルの評価結果を示す．一部を除いて，総じて 100 個の特徴量

表 4：100 個の特徴量を用いたときに標準化方法を変えた場合のモデルの評価評価（値:ROC-AUC）

モデル	実行方法	評価データ：A 銀行				評価データ：B～E 銀行のいずれか			
		連携する銀行				連携する銀行			
		A+B	A+C	A+D	A+E	A+B	A+C	A+D	A+E
MLP	個別	0.846	0.846	0.846	0.847	0.789	0.827	0.828	0.797
	従来	0.845	0.849	0.848	0.847	0.838	0.930	0.878	0.822
	FL	0.851	0.850	0.849	0.849	0.837	0.908	0.749	0.792
	FL(全体標準化)	0.853	0.851	0.846	0.850	0.839	0.923	0.867	0.866

を用いた場合より性能が低下している．一般に機械学習では，使用可能な特徴量が多いほうが，得られたモデルの性能は良い傾向があり，その傾向が FL でも表れている．

5. 考察

本実験では，FL により A 銀行ではモデルの性能向上は得られず，B から E 銀行では性能向上が得られた．A 銀行は他行に比べて多くの訓練データ(特に正例数)を保有済みで，単独でモデルを構築しても一定の性能が得られる．一方，B から E 銀行は，A 銀行と比較すると訓練データ量が少ないことが特徴であった．これらを踏まえると，FL を適用してメリットの得られるケースは，自社のみではデータ量の確保が十分ではない（自社データのみではモデル構築が難しい）という点が必要になると考えられる．その意味で，FL は単独の銀行では訓練データを確保することが難しいようなユースケース（例：不正送金の検知）において親和性が高いのではないかと考えられる．

また，本実験では C から E 銀行では，使用する特徴量の数によらず，MLP（FL)モデルにおいて性能低下が見られた．これはデータの分布差異が影響したものと考えられる．実際にデータの標準化方法を変更して実験したところ， B～E 銀行のいずれかの評価データを用いた場合でも，性能向上が見られた(表 4 の FL（全体標準化）に示す)．ここで標準化方法を変更したとは，クライアントごとに訓練データを標準化するのではなく，他クライアントを含めた全体データの平均・分散が事前に把握できる前提で訓練データを標準化することを指す．なお，標準化方法の変更により A 銀行側の性能低下は見られなかった．

GBDT ベースのモデルでは，標準化等の前処理は不要であり，本実験の結果では性能も安定していた．テーブルデータに対して FL を実行する際は，まずは GBDT ベースのモデルで検討することが良いと考えられる．

最後に，使用する特徴量数が多いほうが FL の結果として得られたモデルの性能が高い傾向であった．FL の場合は，クライアント間で素データを公開可能な場合と比較すると，クライアント間で使用する特徴量を決定するプロセスには手間がかかると想定される．しかし，高い性能のモデルを構築するには，一致する特徴量を多く確保するようにクライアント間で調整することが重要であると考えられる．

6. まとめと今後の課題

本研究では金融機関で使用される実業務データを用いて，カードローンの代位弁済請求を予測するタスクに対して FL を適用した．その際に連携するクライアントが保有するデータ量や，使用するモデルなどの条件を変更し，どのような場合に FL によって性能向上が得られるか検証した．

結果として訓練データを十分に持つクライアントに対しては FL の効果は小さく，訓練データが少ないクライアントに対して FL の効果が大きいことが分かった．また，MLP よりも GBDT ベースのモデルのほうが性能は高く，安定した性能を示した．

今後の課題として 2 点考えられる．1 点目として，クライアントごとに背景にあるデータ分布が大きく異なる（例：顧客層が大きく異なる）ケースでの検証が挙げられる．本研究では 1 つの金融機関のデータを地域ごとに分割し，複数の金融機関のデータを仮想的に作成した．FL では各クライアントが保有するデータの分布が大きく異なる場合に性能低下することが報告[10]されており，そのようなケースでの検証が必要と考える．

2 点目は Horizontal-FL においてクライアント間で共通する特徴量が少ない場合に，性能低下を抑えつつ FL を実行する手法の検討である．共通しなかった特徴量の情報を反映して Horizontal-FL を実行する手法を検討したい．

参考文献

[1] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh and Dave Bacon. Federated Learning: Strategies for Improving

Communication Efficiency. NIPS Workshop on Private Multi-Party Machine Learning. 2016.

[２] Geet Shingi. A federated learning based approach for loan defaults prediction. 2020 International Conference on Data Mining Workshops (ICDMW). 2020.

[３] Wenbo Zheng, Lan Yan, Chao Gou and Fei-Yue Wang. Federated Meta-Learning for Fraudulent Credit Card Detection. Twenty-Ninth International Joint Conference on Artificial Intelligence (IJICAI-20), Special Track on AI in FinTech. 2020.

[４] Qiang Yang, Yang Liu, Tianjian Chen and Yongxin Tong. Federated Machine Learning: Concept and Applications. ACM Transactions on Intelligent Systems and Technology (TIST), Vol.10, Issue.2. 2019.

[５] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang and Shiho Moriai. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption, IEEE Transactions on Information Forensics and Security, Vol.13, No.5, pp.1333-1345. 2018.

[６] Fuki Yamamoto, Lihua Wang and Seiichi Ozawa, New Approaches to Federated XGBoost Learning for Privacy-Preserving Data Analysis. Neural Information Processing, ICONIP 2020. 2020.

[７] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steike, Heiko Ludwig, Rui Zhang and Yi Zhou. A Hybrid Approach to Privacy-Preserving Federated Learning. arXiv preprint. arXiv:1812.03224. 2019.

[８] H.Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson and Blaise Agüera y Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, 20th International Conference on Artificial Intelligence and Statistics (AISTAT). 2017.

[９] Mengwei Yang, Linqi Song, Jie Xu, Congduan Li and Guozhen Tan. The Tradeoff Between Privacy and Accuracy in Anomaly Detection Using Federated XGBoost. arXiv preprint. arXiv:1907.07157. 2019.

[１０] Hangyu Zhu, Jinjin Xu, Shiqing Liu and Yaochu Jin. Federated Learning on Non-IID Data: A Survey. arXiv preprint. arXiv:2106.06843. 2021.