リッチクライアント－エッジサーバ間における プライバシ保護に優れた分散機械学習の検討

高野 紗輝[1] 中尾 彰宏[2] 山口 実靖[3] 小口 正人[1]

概要：近年，federated learning のようにデバイス上にある個人情報を保護しながらそれらのデータをクラウドやエッジサーバ上での機械学習に用いることが盛んに研究されている．しかし，個人情報の一部をエッジデバイスの外へと受け渡す点においてプライバシ保護が十分であるとはいえず，機密性が高くエッジデバイスの外へ情報を一切持ち出したくない個人データを学習に用いることができない．本研究ではエッジサーバと連携しつつエッジデバイス上でも機械学習を動かすプライバシ保護に優れた分散機械学習モデルの検討を行う．本稿では，エッジサーバ上で一般的なデータを用いて学習した結果とエッジデバイス上で個人データを用いて学習した結果を確信度の比較により融合する学習モデルを提案する．Jetson Nano を用いた顔画像認識を行った結果，提案モデルを用いることで画像データの送受信を行うことなく機密性の高いデータも含めた学習が可能となることを確認した．このシステムを用いることで，従来のシステムでは用いることができなかった大量の個人情報が使用できるようになり，一人一人のユーザに適した結果を提示できるようになるという利点がある．

Privacy-Protective Distributed Machine Learning between Rich Client and Edge Server

SAKI TAKANO[1] AKIHIRO NAKAO[2] SANEYASU YAMAGUCHI[3] MASATO OGUCHI[1]

1. はじめに

近年，スマートフォンや IoT デバイスの普及および性能向上により，エッジデバイス上に膨大なデータが蓄積されるようになった．おすすめ表示や画像認識等，多くの場面で機械学習が活用されるようになり，エッジデバイスで収集した個人情報を含む大量のデータのプライバシを強固に守りつつ，これらのデータを含めた機械学習を行うことが期待されている．

現在主流となっているクラウドコンピューティングや新たなコンピューティングモデルとして注目されているエッジコンピューティング [1] では，全ての学習が高性能なサーバ上で行われている．そして，性能の低いエッジデバイス側はあくまでデータを収集し，そのデータをサーバに転送するという役割を果たしてきた．しかし，エッジデバイスで収集するデータには個人情報等の機密性の高い情報が含まれる可能性があり，データをエッジデバイスの外部へと持ち出すことに対してプライバシの問題が生じる．

エッジデバイスの性能向上により，CPU や GPU が搭載され，エッジデバイス内でも機械学習を動かすことのできる程の性能を持つリッチクライアントが登場したことで，より複雑なタスクもエッジデバイス上で実行することが可能となった．エッジデバイスで収集した機密性の高いデータはサーバへと送信せず，これらのデータはエッジデバイス上のみで学習することでプライバシ保護に優れたシステムの構築が期待できる．

一方で，エッジデバイスの性能はサーバと比較してかなり低いため，エッジデバイス上のみでの学習には限界があり，性能の高いサーバとの連携が必要になると考えられる．今回想定する環境は図 1 に示すように，クラウドサーバ，エッジサーバ，エッジデバイスから成り，それぞれが連携

[1] お茶の水女子大学
Ochanomizu University
[2] 東京大学
the University of Tokyo
[3] 工学院大学
Kogakuin University

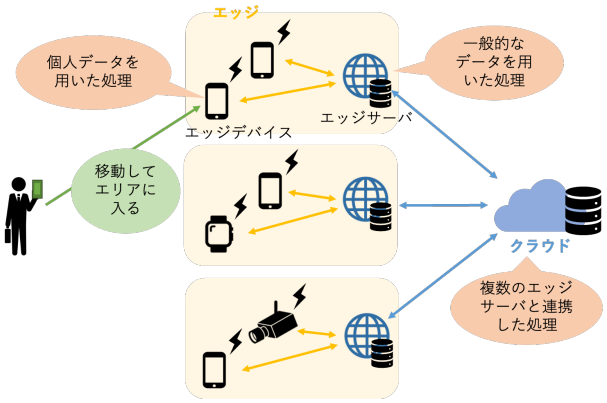

図 1 本研究の想定環境

する．エッジデバイスで収集した個人情報等の機密性の高いデータの所有者はエッジデバイスの所有者と一致するため，エッジデバイスは信用することができる．しかし，エッジサーバやクラウドサーバ及びそれぞれの通信経路は必ずしも信用できないため，機密性の高いデータはエッジデバイス内のみで処理することとする．エッジサーバが点在しており，エッジデバイスが移動して各エッジサーバのエリアに入るとエッジサーバからそのエリアに関する学習結果を受け取る．エッジデバイスでは機密性の高いデータも含めてデータ処理を行い，この学習結果をエッジサーバより受け取った学習結果と融合する．将来的にはエッジサーバ－クラウドサーバ間の連携も視野に入れているが，まずはエッジデバイス－エッジサーバ間の連携に着目する．

本研究では，個人情報を含む学習をエッジデバイス上のみで行うことにより，機密性の高いデータも含めた学習が可能となることを目指す．エッジデバイスの性能はエッジサーバに比べかなり低いため，エッジサーバ上で一般的なデータを用いて学習を行った結果をエッジデバイスへと送信することで補完する学習モデルを検討する．本論文ではエッジデバイス上で確信度の比較を行うことを検討し，学習データとして顔画像，エッジデバイスとして Jetson Nano を用いた実験において，画像データの送受信を行うことなく機密性の高いデータも含めた学習が可能となることを確認した．

本稿の構成は以下の通りである． 第 2 章で提案モデルの元のアイディアとなるエッジ/フォグコンピューティングを紹介し，第 3 章で研究課題について述べる．第 4 章で関連研究としてリッチクライアントを用いた分散機械学習の 1 つである Federated learning を紹介する．第 5 章で解決手法を提案し，第 6 章で Jetson Nano を用いた実装及び評価を行う．第 7 章で結論を述べ，第 8 章でまとめる．

2. エッジ/フォグコンピューティング

エッジコンピューティングとは，ネットワークエッジにエッジサーバを配置し，データ処理を最大限エッジで行うコンピューティングモデルである [2] [3]．現在主流となっているクラウドコンピューティングではユーザは地理的に遠く離れたクラウドにデータを送信し，クラウド内で処理された結果を応答として受け取る．しかし，エッジデバイス－クラウド間の遅延は数百ミリ秒に及ぶ場合があり，帯域も多く必要とするため，リアルタイムアプリケーションや大量のデータを送受信するアプリケーションの実装には向いていない．一方で，エッジコンピューティングは遠隔にあるクラウドのサーバと比較して物理的に近い位置で処理を行うことにより，利点として低遅延である点やエッジデバイスで処理を行うことでクラウドサーバにかかる負荷を分散できる点，エッジデバイスからクラウドサーバへ送信されるデータ量を削減し，トラフィックの混雑を解消できる点が挙げられる [4]．

論文 [5] ではエッジコンピューティングと似たモデルであるフォグコンピューティングについて一般的なモデルとアーキテクチャについて分析し，クラウドコンピューティングでは数十億のデバイスとクラウド間の長距離通信には通信遅延と帯域幅の圧迫という 2 つの問題が生じるが，クラウドで処理していたタスクをネットワークエッジに設置したフォグサーバにオフロードすることで解決されることが示されている．このような利点を活かし，エッジコンピューティングはスマートシティ [6] [7] や高度道路交通システム [8] などで IoT アプリケーションに応用され，クラウドコンピューティングでは実装することができなかったリアルタイムに応答するシステムが構築されている．

一方で，エッジコンピューティングの課題の一つにエッジデバイスが収集した生データの取り扱い方がある．生データを機械学習処理のために収集源であるエッジデバイスからエッジサーバへと送信すると，データをエッジサーバなどデバイスの外部へと受け渡すことによるプライバシの問題や通信コストが高くなるという問題があり，ユーザ認証プロトコルの導入 [9] やエッジデバイス上でのデータの圧縮・特徴量の抽出 [10] などが考えられている．

3. 研究課題

従来のエッジコンピューティングの研究においては，エッジデバイス上でのデータの加工は考えられているものの，性能の低いエッジデバイス側はあくまでデータを収集し，そのデータをエッジサーバに転送するという役割を果たしてきた．一方で，エッジデバイスには機密性が高くデバイスの外へ情報を一切持ち出したくない個人情報が含まれている可能性が高いため，従来のデータを全てエッジサーバに転送して学習する方法ではこのようなデータを学習に用いることができない．特に近年，欧州で EU 一般データ保護規則 (GDPR, General Data Protection Regulation) [11] が定められるなど，プライバシ保護への関心が高まっており，エッジデバイスで収集される個人データをサーバへ受

け渡すことへの抵抗がさらに大きくなると予想される．その結果，エッジサーバで学習した一般的なデータの学習モデルしか利用することができず，個人情報も含めたそれぞれのデバイスに最適な学習モデルを利用することができなくなると考えられる．

さらに，エッジデバイスで収集されるデータはそのデバイスの持ち主に関する個人情報に偏る可能性がある．しかし，学習によって得られる結果は幅広い一般的なデータと個人データの両方に対応できることが求められる．例えば，見守りサービスなどにおいて家庭に設置されるセンサで収集された動画像から動作特定を行う際には，設置されたセンサからは個人情報である個人の動きや物の配置といったその家庭特有な情報のみが収集される．その結果，個人情報に関する知見が多く得られ，個人の動作を正確に判断できる．一方で，一般的なデータに関する知見を持っていない場合，他人が家に来たときやいつもと異なる動きをした際には動作特定が難しくなる可能性がある．そのため，一般的なデータと個人データ両方に関する知見を持つ必要がある．エッジデバイス上で偏ったデータを使用して学習する方法では，幅広い一般的なデータに関する知見を含む学習結果を得ることができないため，課題が残る．

そこで，近年エッジデバイスの性能向上は著しく，エッジデバイスでのデータ処理能力がさらに上がる事が期待されているため，一般的なデータで学習を行うエッジサーバと連携しつつ，エッジデバイス上でも個人情報を用いた重いデータ処理を行うことに挑戦する．

4. 関連研究 (Federated learning)

近年，デバイスの性能向上により，高性能な CPU や GPU を搭載したリッチクライアントが登場し，エッジデバイス上でもサーバが行なっていた機械学習処理を実行できるようになった．そしてデバイス上で機械学習を行うモデルとして，Federated learning (連合学習) という分散型機械学習が提案された [12] [13] [14]．Federated learning では，まずクラウド上のデータで学習を行って得られた学習モデルを各デバイスに配布し，各デバイスはそれぞれが収集した固有のデータを利用してさらに学習を進めた上で変更点の情報のみを暗号化を行なった上でクラウドに送信する．そして，クラウドは各デバイスから収集した変更点を平均化し，元の学習モデルを改善してより良いモデルを作成する．このように各デバイスで収集した生データをデバイスの外部に受け渡さないため，プライバシを担保しつつデバイスにあるデータを機械学習に活用することが可能となる．Federated learning はエッジコンピューティングとは異なり，プライバシに配慮しながらエッジデバイスの情報をクラウドに集約し，クラウドが一括管理するコンピューティングモデルとなっている．

論文 [15] では，Federated learning を Google キーボードに応用した例が実装されており，デバイスの持つ固有のデータを受け渡すことなく，デバイス－クラウド間にまたがる分散機械学習が可能であることが示されている．その他にも，Federated learning は機密性の高いデータを扱う医療現場における情報共有 [16] やヘルスケアアプリケーション [17]，自動車運転時における通信 [18]，スマートシティでのセンサから取得したデータの利用 [19] といった様々な分野での応用が期待され，近年盛んに研究が行われている．

また，それぞれのデバイスごとに保有するデータ量やデータ分布に偏りが存在する場合にはマイノリティなデータが反映されないという問題が発生するが，その偏りに対応したモデルの作成を可能とする学習方法 (Agnostic Federated Learning) も提案された [20]．

しかし，Federated learning におけるプライバシの保護は十分であるとは言えず，クラウドに送信されるパラメタから元画像を鮮明に復元できてしまうという研究報告がある [21] [22]．

5. 解決手法の提案

リッチクライアントの登場により，機械学習等の複雑な処理もエッジデバイス上で行うことが可能になったことと合わせ，上記の課題の解決を目指したリッチクライアントに適した分散機械学習モデルを提案する．具体的には，エッジコンピューティングモデルにおいて，従来エッジサーバ上で行っていたタスクの一部をエッジデバイスにオフロードすることでエッジデバイス上でも機械学習処理を行う．

5.1 先行研究との比較

先行研究 [23] では，エッジデバイスがエッジサーバ上における学習の結果とエッジサーバ上にある一般的なデータを一部受け取り，個人データを含むデータを用いて学習をエッジデバイス上で引き継き行うモデルを提案した．このエッジサーバと連携しつつ機密性の高い情報はエッジデバイスの外へと一切持ち出さない提案モデルを実装した結果，エッジサーバ上のデータのうち 3 割程度を受け取り，学習をエッジデバイス上で引き継ぐことで早くに精度の高い学習結果を得ることができ，機密性の高いデータも含めた学習が可能となることを示した．一方で，このモデルではエッジデバイスで一般的なデータと個人データの両方に対応する結果を得るためにエッジサーバ上のデータの一部をエッジデバイスに転送する必要があり，膨大なデータを用いた学習では通信コストの面において好ましくない．

そこで，アンサンブル学習のように複数のモデルを用いて全体の精度をあげる手法を提案する．アンサンブル学習とは，1 つ 1 つのモデルでは精度が低い複数のモデルを融合させることで 1 つの学習モデルを生成し，高い精度を得

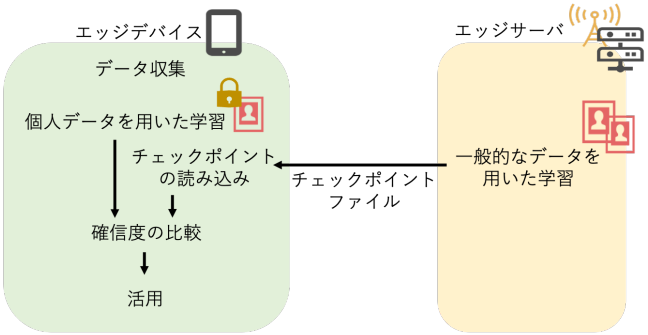

図 2 提案モデル

る手法である [24]．そのうちバギングという手法では，並列に学習した複数モデルの結果を回帰問題であれば平均値，分類問題であれば多数決の形で結果を集約する [25] [26]．

5.2 提案モデル

ここでは，モデルの確信度を用いることで，エッジサーバ上のデータを転送せずにエッジデバイス上で学習を行うモデルを提案する．提案モデルの概要図を図 2 に示す．

エッジサーバにおいて，あらかじめ一般的なデータを用いて学習を行い，学習の重みを保存したチェックポイントファイルを作成しておく．スマートフォンなどのエッジデバイスが移動し，エッジサーバに接続すると，エッジサーバ上で作成されたチェックポイントファイルを受け取る．エッジデバイスでは，収集した個人データを用いて学習を行う．このエッジデバイスで得た学習結果とエッジサーバより受け取ったチェックポイントファイルを読み込むことで得た結果を用いて確信度の比較を行う．具体的には，エッジサーバで一般的なデータを用いて学習を行ったモデルと，エッジデバイスで個人データを用いて学習を行ったモデルをエッジデバイス上でロードし，判定したいデータを両方のモデルに与える．予測がどのくらい確実であるかの統計的な尺度である確信度をそれぞれ算出し，確信度の高い方の判定結果を使用する．その結果，個人データに関する予測はエッジデバイス上での学習結果を使用し，一般的なデータに関する予測はエッジサーバ上での学習結果を使用することで，全体の学習精度が向上すると期待される．

このモデルは，エッジデバイスで収集した個人情報はエッジデバイス内のみで処理を行い，エッジサーバへ情報を一切渡さないという特徴を持つ．そのため，情報の一部をサーバへと送ることで生じている federated learning の問題を解決しつつ、個人情報を活用することが可能となる．

6. 提案手法の実装と評価

6.1 データセット

実験には実際のアプリケーションなどで使用されることが想定される機密性が高く，容量の大きな顔画像を用いることとする．

表 1 エッジサーバの性能

OS	Ubuntu 18.04 LTS
CPU	Intel Core i7-8700
GPU	GeForce RTX 2080Ti
Memory	32Gbyte

表 2 エッジデバイス (Jetson Nano) の性能

OS	Ubuntu 18.04 LTS
CPU	Quad-core ARM A57 @ 1.43 GHz
GPU	128-core Maxwell
Memory	4 GB 64-bit LPDDR4 25.6 GB/s

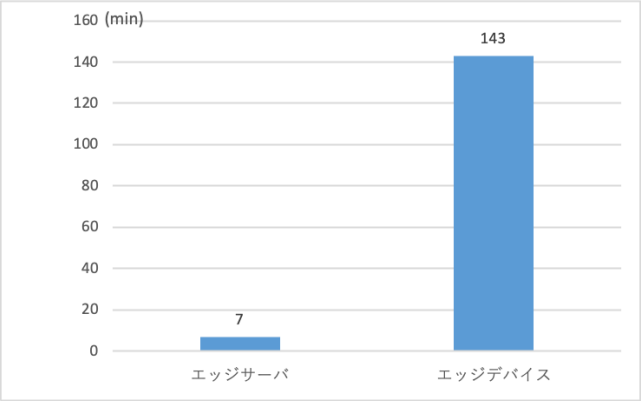

図 3 エッジサーバ，エッジデバイスによる lfw の実行時間

インターネット上より集められた jpg 画像を人物毎にフォルダ分けしてある Labeled Faces in the Wild (以下 lfw) [27] を使用する．各フォルダ毎に写真の顔抽出を行い，適切に抽出を行うことのできていない写真を取り除いた後，各フォルダの 2 割を test データとする．残りの写真を train データとし，ぼかし等により 9 倍にして使用する．

6.2 実験環境

実験で使用したエッジサーバの性能を表 1 に，エッジデバイスとして使用した Jetson Nano の性能を表 2 に示す．

Jetson Nano は GPU を搭載した小型 AI コンピュータボードであり，近い将来，スマートフォンや様々な IoT デバイスがこのような性能を持つことが期待される．しかし，性能はエッジサーバと比較すると劣り，GPU のコア数がエッジサーバは 4352 コアであるのに対し，Jetson Nano は 128 コアと大きな差がある．

本実験では分散処理に適している TensorFlow を機械学習に使用し，Jetson Nano －エッジサーバ間はイーサネットで接続する．

6.3 予備実験

エッジサーバとエッジデバイスにおいて機械学習処理を行った際の実行時間を比較する．ここでは lfw に含まれる 33 人について 30 枚ずつ画像を抜き出し，顔抽出および train データと test データに分ける処理を行なったデータを用いる．エッジデバイス，エッジサーバ共に精度が 65%

となるよう学習した結果を図 3 に示す．

エッジデバイス上でもエッジサーバと同等精度の学習を行うことができるものの，およそ 20 倍の時間を要し，lfw を用いた学習では 65%の精度を得るために 2 時間以上の学習が必要となる．このことから，エッジデバイスは低速ではあるが，エッジデバイス内のみでも学習可能であることが分かり，プライバシが非常に重要なデータもそのような形で学習に用いる事ができる．しかし，エッジデバイスのみでの学習には限界があり，エッジサーバとの連携が重要になると考えられる．

6.4 実験 1 (エッジデバイス上での学習に用いる train データ：個人データのみ)

6.4.1 実験概要 (実験 1)

エッジデバイスが収集した機密性の高い個人の顔写真のみを保持している状態で提案モデルを実行する．ここではエッジサーバでの学習には lfw に含まれる 30 人を一般的なデータとして用い，エッジデバイスの学習には lfw に含まれる Tony Blair の写真を個人情報と見立てて実験を行う．エッジサーバで用いる 30 人については，それぞれ train データを約 23 枚ずつ，test データを約 6 枚ずつ用意し，エッジデバイスではその持ち主の写真が多く収集されると考えられるため，Tony Blair の train データを 86 枚，test データを 21 枚用意する．

まず初めに，エッジサーバにおいて個人情報を含まない一般的なデータを用いて epoch 数を 100，各 epoch の steps 数を 198 として十分に学習を行う．エッジサーバの性能は高く，短時間で多くの学習を行うことが可能であるため，エッジサーバの持つデータにおいて学習の上限となる精度を得ることが可能な epoch 数を設定する．そして，学習の重みを保存したチェックポイントファイルをエッジデバイスへと送信する．エッジデバイスでは，個人データを用いた学習を epoch 数を 20，各 epoch の steps 数を 25 としてあらかじめ行なっておく．エッジデバイスがチェックポイントファイルを受け取ると，これを読み込み，エッジサーバで一般的なデータを用いて学習を行ったモデルとエッジデバイスで個人データを用いて学習を行ったモデルの確信度を比較する．

6.4.2 実験結果 (実験 1)

エッジサーバ上での学習後にエッジサーバ上で計測した精度 (①)，そこで得られた結果をエッジデバイス上で計測した精度 (②)，エッジデバイスで個人データを用いて学習を行った後にエッジデバイス上で計測した精度 (③)，確信度の比較を行ったのちにエッジデバイス上で計測した精度 (④) を図 4 に示す．精度はエッジサーバ上では個人データの含まれない test データで計測し，エッジデバイス上では最終的に判断できるようになりたいエッジサーバ上の一般的なデータと個人データを共に含む test データで計測する．

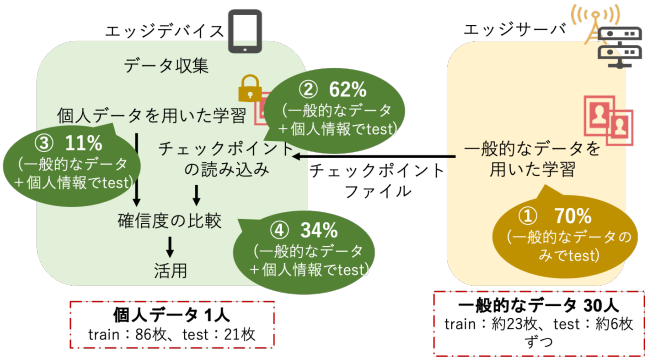

図 4 エッジデバイスに train データとして個人データのみを与えた際の学習精度 (実験 1)

エッジサーバ上では①で示すように一般的なデータに対して 70%まで学習することが可能である．得られたチェックポイントファイルをエッジデバイスへと渡し，エッジデバイス側で個人データが含まれる test データを用いて計測すると精度は②で示すように 62%となり，個人データに対応することができない分，精度が下がる結果となる．一方，エッジデバイス上の学習では③で示すように 11%と個人データには対応できるものの一般的なデータには全く対応のできない結果となる．そして，確信度の比較を行うと 34%となる．これは，個人データに関してはエッジデバイスで学習した結果の確信度がエッジサーバで学習した結果の確信度より高くなるため，個人データは全て当てることができる．しかし，一般的なデータに関してはエッジサーバでの学習によって得られた結果を用いて高い確信度で正解を導けるものがある一方，エッジデバイスでの学習によって得られた結果を用いて高い確信度で誤った人物だと認識することも多いため，全体として低い精度になっている．

6.4.3 考察 (実験 1)

エッジデバイス上での学習に個人データ (1 人分) しか用いなかった場合には，全ての test 画像に対して高い確信度でエッジデバイス上で学習した個人の画像だと認識してしまう．そのため，確信度の比較を行うとエッジサーバで学習した結果を用いて正しく判別するよりも高い確信度でエッジデバイス上の学習結果で誤った判別を行ってしまうことが多い．結果，個人データに対する学習精度は高いものの，一般的なデータも含めた全体の精度は低くなる．

そこで，エッジデバイス上での学習に個人データ以外を含めることを考える．エッジサーバから学習のたびにデータを転送するのは，エッジデバイスの容量や通信コストの面から好ましくない．そこで，エッジデバイス上での学習にはエッジサーバで学習した人物とは異なる人物の画像を含めることを考える．これは，エッジサーバの学習に使用するデータは最新のデータに次々と置き換わっていくが，エッジデバイスは初回アプリケーションダウンロード時に

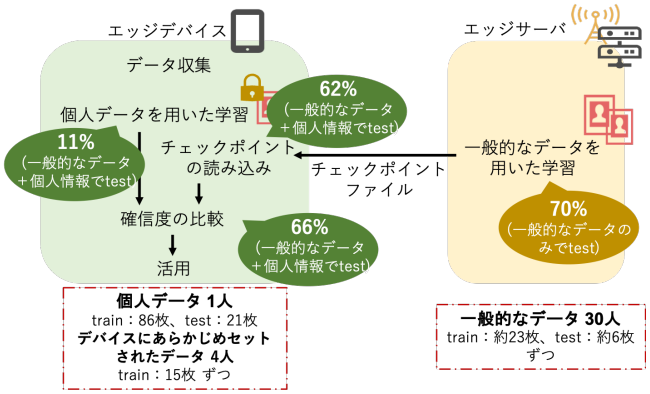

図 5 エッジデバイスに train データとして個人データとサーバとは異なる 4 人を与えた際の学習精度 (実験 2)

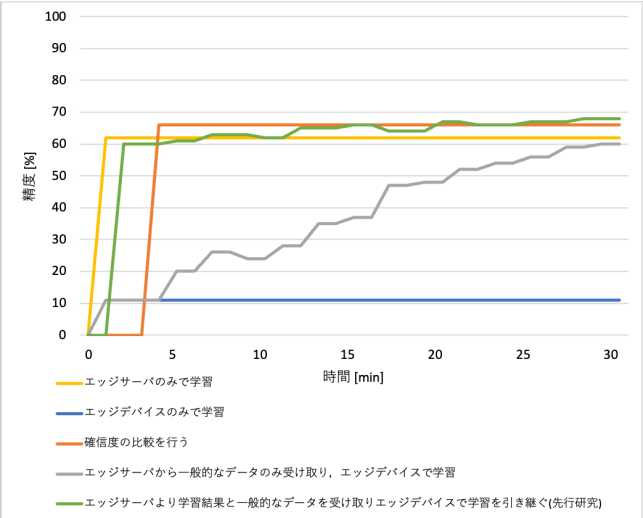

図 6 エッジデバイスに train データとして個人データとサーバとは異なる 4 人を与えた際の学習精度と時間の関係

取得した古い画像などをあらかじめセットしておくことを想定している．

6.5 実験 2 (エッジデバイス上での学習に用いる train データ：個人データ＋サーバとは異なる一般的な人物 4 人)

6.5.1 実験概要 (実験 2)

エッジサーバでは実験 1 と同じデータを使用し，エッジデバイスでは実験 1 で用いた train データに加えてサーバとは異なる人物を lfw より 4 人用意して使用する．この 4 人についてはそれぞれ 15 枚ずつ画像を用意する．test データは実験 1 と同じものを用いる．

エッジサーバ上では実験 1 と同様に epoch 数を 100，各 epoch の steps 数を 198 として十分に学習を行い，学習の重みを保存したチェックポイントファイルをエッジデバイスへと送信する．エッジデバイスでは，個人データとサーバとは異なる 4 人を用いた学習を epoch 数を 20，各 epoch の steps 数を 42 としてあらかじめ行なっておく．そして，エッジサーバで一般的なデータを用いて学習を行ったモデルとエッジデバイスで個人データとサーバとは異なる 4 人を用いて学習を行ったモデルの確信度を比較する．

6.5.2 実験結果 (実験 2)

各ステップでの学習精度を図 5 に示す．

エッジサーバ上では一般的なデータに対して 70%まで学習することが可能であり，エッジデバイス側で個人データが含まれる test データを用いて計測すると精度は 62%となる．一方，エッジデバイス上の学習では 11%と個人データに関しては 95%の精度で正しく判別できるものの，一般的なデータには全く対応のできない結果となる．そして，確信度の比較を行うと 66%となる．これは，個人データに関しては 76%の精度で判別することができており，エッジデバイスで学習した結果の確信度がエッジサーバで学習した結果の確信度より高くなるため，個人データは高い確率で当てることができる．一般的なデータに関しても，エッジデバイス上で学習した結果を用いて誤った判別をするよりも高い確信度でエッジサーバによる学習結果を用いて正しい判別を行うことができるものが多いため，全体として良い精度になっている．

6.5.3 精度と時間の関係 (実験 2)

エッジデバイスでチェックポイントファイルを受け取った直後からの時間を横軸として学習精度を図 6 に示す．

黄色のグラフがエッジサーバでの学習を読み込んだだけで確信度の比較を行わなかった際の精度，青のグラフが確信度の比較を行わずにエッジデバイスのみで学習を行った際の精度，赤のグラフが確信度の比較を行った際の精度を表している．ここではエッジデバイス上での学習はあらかじめ行われていたものとしている．エッジデバイスの性能が低く，実験 2 においてエッジデバイス上で 20 epoch 分の学習を行うには 9 分必要であるが，エッジデバイス上での学習にはエッジサーバのデータを使用しないため，エッジサーバとの接続を待たずにエッジデバイスが使われていない時間にあらかじめ学習を行っておくことができる．

エッジサーバのみでの学習では個人データに対応することができずに 62%の精度，エッジデバイスのみでの学習では一般的なデータに対応することができずに 11%の精度しか得ることができない．確信度の比較を行うことで，エッジサーバでの学習とエッジデバイスでの学習を融合した結果が得られる 3 分後には個人データと一般的なデータ両方に対してバランスよく学習した結果が得られる．

さらに，エッジサーバの助けを借りずに，エッジサーバから一般的なデータのみを受け取り，エッジデバイス上のみで学習を行った結果を灰色のグラフ，先行研究 [23] で扱った，エッジサーバより学習結果と一般的なデータを受け取りエッジデバイスで学習を引き継ぐモデルを緑のグラフで示し比較する．エッジデバイスの性能の低さから，エッジデバイス上で機械学習を動かすにはかなりの時間がかかる．そのため，エッジサーバの助けを借りることが有

効であり，確信度の比較を行うことでより早くに精度の良い結果を得ることができる．

6.5.4 考察 (実験 2)

エッジデバイス上での学習に個人データ以外を含めることで一般的なデータにも個人データにも対応した結果を得ることが可能である．この際に使用する一般的なデータは判別したい人物とは異なる人物で良い．そのため，エッジサーバ上のデータを転送する必要がなく，エッジデバイスの容量や通信コストの面で利点がある．さらに，エッジデバイスが使用されていない間にあらかじめエッジデバイス上で学習を行っておくことで，早くに一般的なデータにも個人データにも対応した結果を得ることができる．

6.6 データ数・学習量の検討

実験 2 では一例としてエッジデバイスにあらかじめセットされたデータとしてエッジサーバとは異なる人物を 4 人与え，学習をエッジサーバにおいて 100 epoch，エッジデバイスにおいて 20 epoch で行なっている．ここでは，データ数・学習量と精度の関係を明らかにするため，あらかじめセットされたデータとして扱う人数および学習の epoch 数を変化さて実験を行う．

(1) あらかじめセットされたデータとして扱う人数：4 人，エッジサーバでの学習：100 epoch

実験 2 と同様に，エッジサーバにおいて epoch 100 で学習を行い，エッジデバイスにおいてエッジサーバで保持しているデータとは異なる人物 4 人と個人データ 1 人を用いて学習を行う．エッジデバイスでの学習時における epoch 数を変化させた結果を表 3 にまとめる．エッジデバイス上での学習時における各 epoch の step 数は 42 である．この際，エッジサーバ上における一般的な人物 30 人について学習した結果をエッジデバイス上で一般的なデータと個人データを共に含む test データを用いて精度を計測すると 62%である．上段より，エッジデバイス上においてエッジサーバの持つデータとは異なる 4 人と個人データを含むデータを用いて学習を行った際の精度，その内，個人データに対して判別できる割合，エッジサーバとエッジデバイスで得た学習結果から確信度の比較を行った際の精度，その内，個人データに対して判別できる割合を計測した結果を示す．

エッジデバイスでの学習量が増えるにつれ，個人データに対する知見は増えるものの，エッジサーバで学習した一般的なデータが判別しにくくなる．

(2) あらかじめセットされたデータとして扱う人数：8 人，エッジサーバでの学習：100 epoch

(1) のケースにおいて，エッジデバイスでの学習の際に用いるサーバと異なる人物のデータを 4 人から 8 人に増やす．1 人あたりの人数は (1) と同様である．結果を表 4 にまとめる．エッジデバイス上での学習時における各 epoch の step 数は 58 である．

先ほどと同様にエッジデバイスでの学習量が増えるにつれ，個人データに対する知見は増えるものの，エッジサーバで学習した一般的なデータが判別しにくくなる．一方で，エッジデバイスでの学習の際に使用するデータ数を増やしたことによりエッジデバイス上での学習結果から得られる確信度が (1) の際と比べ全体的に低くなるため，epoch 数を増やした場合であっても全体の精度が大幅に減少していない．

(3) あらかじめセットされたデータとして扱う人数：4 人，エッジサーバでの学習：40 epoch

次にエッジサーバでの学習量を 40 epoch と減らし，エッジデバイスでの学習にはエッジサーバで保持しているデータとは異なる人物 4 人を用いる．エッジサーバ上における学習結果をエッジデバイス上で一般的なデータと個人データを共に含む test データを用いて精度を計測すると 62%である．結果を表 5 にまとめる．エッジデバイス上での学習時における各 epoch の step 数は 42 である．

エッジサーバでの学習量が (1) の際と比較して少ないため，エッジサーバ上での学習から得られる確信度が全体的に低くなる．一方で，学習量が増えるとエッ

表 3 あらかじめセットされたデータとして扱う人数：4 人，エッジサーバでの学習：100 epoch の際にエッジデバイスでの学習量を変化させた結果

epoch 数	20	40	100
エッジデバイス上での学習精度	11%	11%	11%
(内，個人データに対する精度)	95%	95%	95%
確信度の比較を行った際の精度	66%	62%	49%
(内，個人データに対する精度)	76%	81%	86%

表 4 あらかじめセットされたデータとして扱う人数：8 人，エッジサーバでの学習：100 epoch の際にエッジデバイスでの学習量を変化させた結果

epoch 数	20	40	100
エッジデバイス上での学習精度	10%	10%	11%
(内，個人データに対する精度)	86%	90%	95%
確信度の比較を行った際の精度	66%	64%	61%
(内，個人データに対する精度)	62%	76%	90%

表 5 あらかじめセットされたデータとして扱う人数：4 人，エッジサーバでの学習：40 epoch の際にエッジデバイスでの学習量を変化させた結果

epoch 数	20	40	100
エッジデバイス上での学習精度	11%	11%	11%
(内，個人データに対する精度)	95%	95%	95%
確信度の比較を行った際の精度	60%	56%	48%
(内，個人データに対する精度)	81%	86%	90%

表 6 あらかじめセットされたデータとして扱う人数：8 人，エッジサーバでの学習：40 epoch の際にエッジデバイスでの学習量を変化させた結果

epoch 数	20	40	100
エッジデバイス上での学習精度	10%	10%	11%
(内，個人データに対する精度)	86%	90%	95%
確信度の比較を行った際の精度	66%	62%	60%
(内，個人データに対する精度)	67%	71%	76%

ジデバイス上での学習から得られる確信度が全体的に高くなる傾向があり，エッジデバイスで学習を行った個人データに対しては高い確率で判別できるが，エッジサーバで行った一般的なデータの学習による結果が反映されにくくなる．

(4) あらかじめセットされたデータとして扱う人数：8 人，エッジサーバでの学習：40 epoch

(3) のケースにおいて，エッジデバイスでの学習の際に用いるサーバと異なる人物のデータを (1) と同じく 8 人に増やす．結果を表 6 にまとめる．エッジデバイス上での学習時における各 epoch の step 数は 58 である．

(3) の際と比較してエッジデバイスで学習する際に用いる人数が増えるため，エッジデバイス上での学習から得られる確信度が全体的に低くなる．そのため，エッジサーバとエッジデバイスから得られる学習結果の反映されるバランスが改善され，(3) に比べ全体の精度が上がる結果となっている．

(1)〜(4) より，エッジサーバで学習した一般的なデータに関する精度の向上とエッジデバイスで学習した個人データに関する精度の向上はトレードオフの関係にある．一般的なデータと個人データの両方にバランスよく対応するためには，データ量および学習量が重要となる．

7. 結論

以上の実験より，エッジサーバの助けを借りつつエッジデバイス上で個人情報を用いた学習を行うことで，早い段階において精度の高い学習結果を得ることができ，本提案モデルを用いることで機密性の高いデータも含めた学習が可能となる．

エッジデバイス上の学習にはある程度のデータの種類を用意する必要があるが，これらのデータは最終的に判定を行いたいデータと同じである必要はない．そのため，エッジサーバとは異なるデータをエッジデバイスの学習に使用する本提案モデルは，学習時における画像データの転送が一切必要なく，通信コストの面で優れている．また，一般的なデータに関する精度の向上とエッジデバイスで学習した個人データに関する精度の向上はトレードオフの関係にあるため，エッジサーバおよびエッジデバイスでの学習の際に用いるデータ数と学習量が最終的に得られる精度に影響を与える．学習の際に用いるデータ数と学習量を適切に定めることで，一般的なデータ，個人データ共に良い精度を得ることが可能である．

8. まとめと今後の課題

従来のエッジコンピューティングで課題となっている，エッジデバイスの外へと一切持ち出したくない個人データを含めた学習を可能とすることを目的として，リッチクライアントに適した分散機械学習モデルの検討を行った．

エッジデバイス上においても機械学習を動かし，その学習結果とエッジサーバ上での学習結果の確信度を比較するモデルを提案した．そして，エッジデバイスとして Jetson Nano，学習データとして顔画像を用いて実装を行った．エッジデバイスで収集した個人情報はエッジデバイス内のみで処理を行い，エッジサーバへ情報を一切渡さないため，プライバシ保護が可能となった．さらに，エッジサーバの助けを借りることにより，短時間で一般的な情報と個人情報の両方に対応できる良い精度を得ることが可能であることが示された．また，一般的な情報に関する精度の向上とエッジデバイスで学習した個人情報に関する精度の向上はトレードオフの関係にあるため，エッジサーバおよびエッジデバイスでの学習の際に用いるデータ数と学習量を適切に定めることで良い精度を得ることが可能であることが示された．

今後はデータセットを変化させて実験を行い，エッジサーバとエッジデバイスでの学習における適切なデータ数と学習量の検討を予定している．また，プライバシを保護した上でエッジデバイスの情報をエッジサーバやクラウドにフィードバックすることも検討していきたい．

参考文献

[1] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella. On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. *IEEE Communications Surveys Tutorials*, Vol. 19, No. 3, pp. 1657–1681, 2017.

[2] MG Sarwar Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hussain. Machine learning at the network edge: A survey. *ACM Computing Surveys (CSUR)*, Vol. 54, No. 8, pp. 1–37, 2021.

[3] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, Vol. 3, No. 5, pp. 637–646, 2016.

[4] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, Vol. 50, No. 1, pp. 30–39, 2017.

[5] S. Yang. Iot stream processing and analytics in the fog. *IEEE Communications Magazine*, Vol. 55, No. 8, pp. 21–27, 2017.

[6] N. Chen, Y. Chen, S. Song, C. Huang, and X. Ye. Poster abstract: Smart urban surveillance using fog computing.

In *2016 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 95–96, 2016.

[7] Bo Tang, Zhen Chen, Gerald Hefferman, Shuyi Pei, Tao Wei, Haibo He, and Qing Yang. Incorporating intelligence in fog computing for big data analysis in smart cities. *IEEE Transactions on Industrial Informatics*, Vol. 13, No. 5, pp. 2140–2150, 2017.

[8] J. Lin, W. Yu, X. Yang, Q. Yang, X. Fu, and W. Zhao. A real-time en-route route guidance decision scheme for transportation-based cyberphysical systems. *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 3, pp. 2551–2566, 2017.

[9] Junho Lee, Dongwook Kim, Jinhyun Park, and Hyungweon Park. A multi-server authentication protocol achieving privacy protection and traceability for 5g mobile edge computing. *Proc. of the 39th IEEE International Conference on Consumer Electronics (ICCE 2021)*, January 2021.

[10] Shangguang Wang, Chuntao Ding, Ning Zhang, Xiulong Liu, Ao Zhou, Jiannong Cao, and Xuemin Shen. A cloud-guided feature extraction approach for image retrieval in mobile edge computing. *IEEE Transactions on Mobile Computing*, Vol. 20, No. 2, pp. 292–305, 2021.

[11] General Data Protection Regulation. `https://eur-lex.europa.eu/eli/reg/2016/679/oj`. (2021/04 閲覧).

[12] 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, No. 2, pp. 1–19, 2019.

[13] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, Vol. 149, p. 106854, 2020.

[14] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, Vol. 216, p. 106775, 2021.

[15] T. Yang, G. Andrew, Hubert Eichner, Haicheng Sun, W. Li, Nicholas Kong, D. Ramage, and F. Beaufays. Applied federated learning: Improving google keyboard query suggestions. *ArXiv*, Vol. abs/1812.02903, , 2018.

[16] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, Vol. 5, No. 1, pp. 1–19, 2021.

[17] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, Vol. 3, No. 1, pp. 1–7, 2020.

[18] Dongdong Ye, Rong Yu, Miao Pan, and Zhu Han. Federated learning in vehicular edge computing: A selective model aggregation approach. *IEEE Access*, Vol. 8, pp. 23920–23935, 2020.

[19] Ji Chu Jiang, Burak Kantarci, Sema Oktug, and Tolga Soyata. Federated learning in smart city sensing: Challenges and opportunities. *Sensors*, Vol. 20, No. 21, p. 6230, 2020.

[20] M. Mohri, Gary Sivek, and A. T. Suresh. Agnostic federated learning. *ArXiv*, Vol. abs/1902.00146, , 2019.

[21] Mengkai Song, Zhibo Wang, Zhifei Zhang, Yang Song, Qian Wang, Ju Ren, and Hairong Qi. Analyzing user-level privacy attack against federated learning. *IEEE Journal on Selected Areas in Communications*, Vol. 38, No. 10, pp. 2430–2444, 2020.

[22] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients–how easy is it to break privacy in federated learning? *arXiv preprint arXiv:2003.14053*, 2020.

[23] Saki Takano, Akihiro Nakao, Saneyasu Yamaguchi, and Masato Oguchi. Privacy-protective distributed machine learning using rich clients. *2021 International Conference on Emerging Technologies for Communications (ICETC 2021), IEICE Proceedings Series*, Vol. 68, No. C1-4, 2021.

[24] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, Vol. 11, pp. 169–198, 1999.

[25] Leo Breiman. Bagging predictors. *Machine learning*, Vol. 24, No. 2, pp. 123–140, 1996.

[26] Goksu Tuysuzoglu and Derya Birant. Enhanced bagging (ebagging): A novel approach for ensemble learning. *Int. Arab. J. Inf. Technol*, Vol. 17, No. 4, pp. 515–528, 2020.

[27] Labeled Faces in the Wild. `http://vis-www.cs.umass.edu/lfw/`. (2021/04 閲覧).