DEIM2022 K21-5

リッチデバイスを用いたプライバシ保護に優れた分散機械学習モデルにおける顔画像認識

高野　紗輝†　　中尾　彰宏††　　山口　実靖†††　　小口　正人†

† お茶の水女子大学　〒 112–8610 東京都文京区大塚 2-1-1
†† 東京大学　〒 113–8654 東京都文京区本郷 7-3-1
††† 工学院大学　〒 163-8677 東京都新宿区西新宿 1-24-2
E-mail: †saki-t@ogl.is.ocha.ac.jp, ††nakao@nakao-lab.org, †††sane@cc.kogakuin.ac.jp,
††††oguchi@is.ocha.ac.jp

あらまし　近年，federated learning など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，プライバシ保護が十分であるとはいえず，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リッチデバイスに適し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をエッジデバイスで引き継ぎ，個人データも含めた機械学習を行う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．

キーワード　エッジコンピューティング, 分散機械学習, federated learning, 機械学習, IoT デバイス

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．さらに，おすすめ表示や画像認識などと様々な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に対して，プライバシを守りながら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 [16] 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データやエッジサーバでの学習結果を受け取り，機密性の高いデータも含めてデータ処理を行う．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．

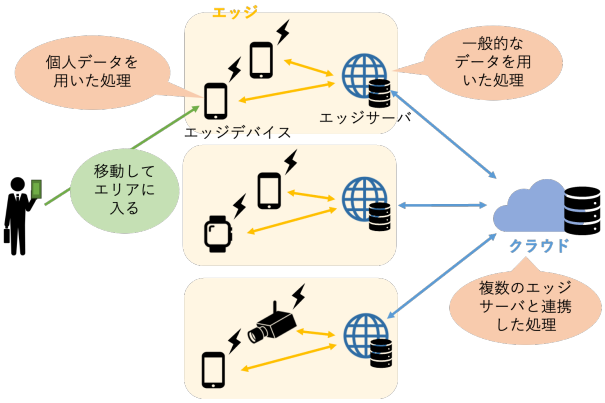

図 1　本研究の想定環境

我々は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し，エッジデバイス上で個人情報に関する学習を引き続き行う分散機械学習モデルを検討する．先行研究では，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，エッジサーバ上での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 [15]．本論文では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，新たにエッジデバイス上での学習の際に一般的なデータをどの程度含める必要があるのかについて，様々なケースのデータで評価を行い，より汎用性の高い知見を得た．

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 [10] [13]．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 [12]．

論文 [21] 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 [3] [17] 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 [6] やエッジデバイス上でのデータの圧縮・特徴量の抽出 [18] 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) [1] 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．

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，エッジサーバと連携しつつ，エッジデバイス上でも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の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 [20] [7] [24]．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

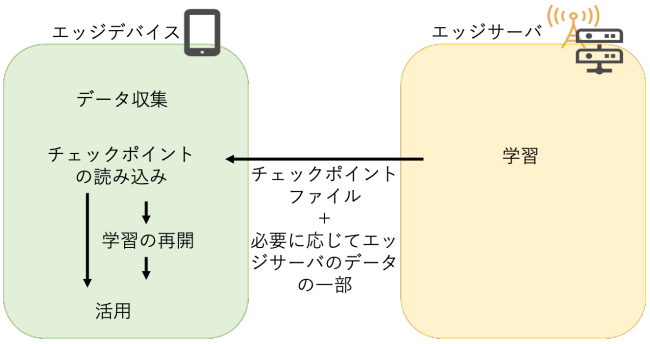

図 2　提案モデル

デルとなっている．

論文 [22] 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 [19] やヘルスケアアプリケーション [11]，自動車運転時における通信 [23]，スマートシティでのセンサから取得したデータの利用 [5] 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 [9]．

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 [14] [4]．

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．

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．また，必要に応じてエッジサーバのデータの一部を受け取る．このチェックポイントファイルを読み込み，即時的に活用することも可能だが，エッジデバイスに適した学習結果を使用したい場合にはエッジデバイスで収集した個人データを用いて学習を再開する．

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．

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

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した．

インターネット上より集められた jpg 画像を人物ごとにフォルダ分けしてある Labeled Faces in the Wild (以下 lfw) [2] から，1 人あたりの写真が 30 枚を超える人物について 30 枚ずつ抜き出して使用した．33 人分のフォルダを作成し，内訳は男性 28 人，女性 5 人となった． 各写真について顔抽出を行い，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 データとした． 残りの写真を train データとし，ぼかし等により 9 倍にして使用した．その結果，train データは 1 人あたり約 23 枚を 9 倍に加工した約 207 枚，test データは 1 人あたり約 6 枚となった．

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，性能はサーバと比較すると劣り，GPU のコア数がサーバは 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した．

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．lfw を用いてエッジデバイス，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

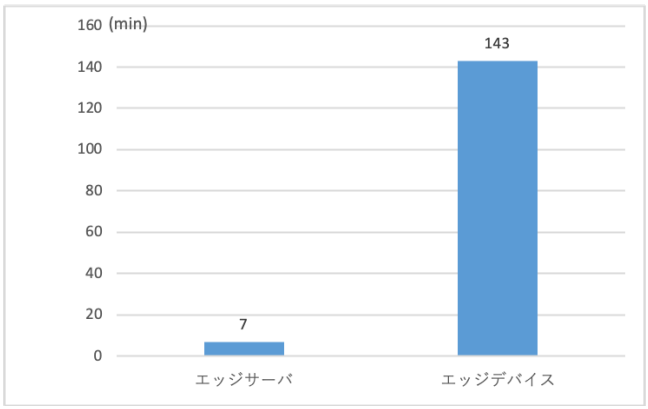

図 3　エッジサーバ，エッジデバイスによる lfw の実行時間

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（train データ：個人データ ＋ サーバの全てのデータ）

6.4.1　実験概要

エッジデバイスがエッジサーバの全てのデータに加え，機密性の高い情報として個人の顔写真を保持している状態で提案モデルを実行する．ここでは lfw より作成したデータセットのうち Tony Blair の写真を個人情報と見立てて実験を行う．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 枚の顔画像をぼかし等で 9 倍に加工したものに置き換え，test データは 21 枚となるように置き換える．train データ，test データ共に Tony Blair の画像が全体の約 10%を占めていることとなる．

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 40，各 epoch の steps 数を 212 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とエッジサーバの全てのデータをエッジデバイスへと送信する．エッジデバイスは受け取ったチェックポイントファイルを読み込み，個人の顔画像も含むデータを用いて epoch 数を 10，各 epoch の steps 数を 236 で学習を再開させる．

6.4.2　実験結果

エッジサーバ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)，エッジデバイスでさらに学習を行った後にエッジデバイス上で計測した精度 (③) を図 4 に示す．精度はエッジサーバ上では個人情報の含まれない test データで計測し，エッジデバイス上では全体の 10%を個人情報が占める test データで計測する．

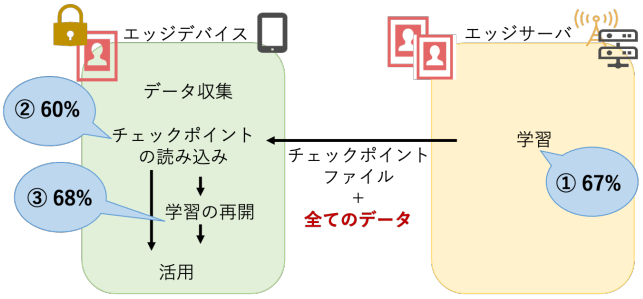

図 4　train データとして個人データとサーバの全てのデータを与えた際の学習精度

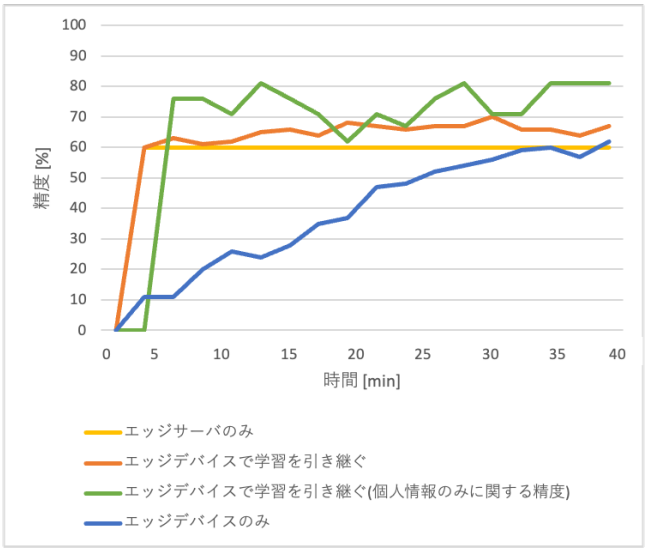

図 5　train データとして個人データとサーバの全てのデータを与えた際の学習精度の詳細

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 67%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情報が含まれる test データを用いて計測すると精度は②で示すように 60%となり，個人情報に対応することができない分，精度が下がる結果となる．エッジデバイス上で学習を再開させることで③で示すように 68%の精度を得ることができ，個人情報にも対応できる結果となる．

6.4.3　学習精度の詳細

エッジデバイスでチェクポイントファイルを読み込む直前からの時間を横軸として学習精度を図 5 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さらに学習を行わなかった際の精度，赤のグラフがエッジデバイス上で個人の顔画像も含めて学習を引き継いだ際の精度，緑のグラフがエッジデバイス上で個人の顔画像も含めて学習を引き継いだ際に個人情報のみに関して計測した精度，つまり個人の顔画像のみを test データとして与えてそれを正しく識別できるかを計測したものである．エッジデバイス上における個人データを含む学習によって赤の全体の精度および緑の個人情報に対する精度が上がる結果となる．

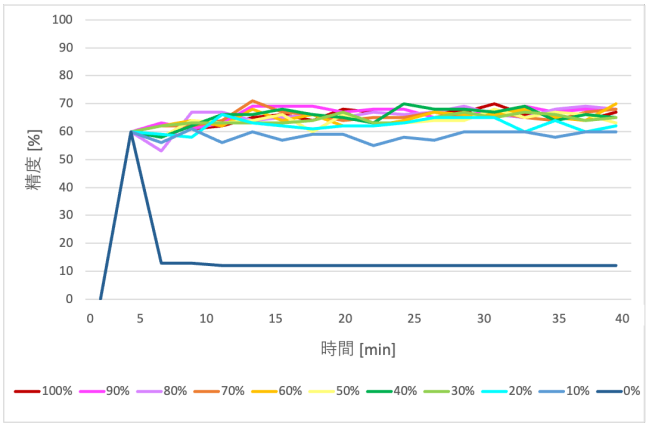

図 6　サーバから受け取る一般的なデータの割合を 0〜10 割まで変化させた際の全体の精度の比較

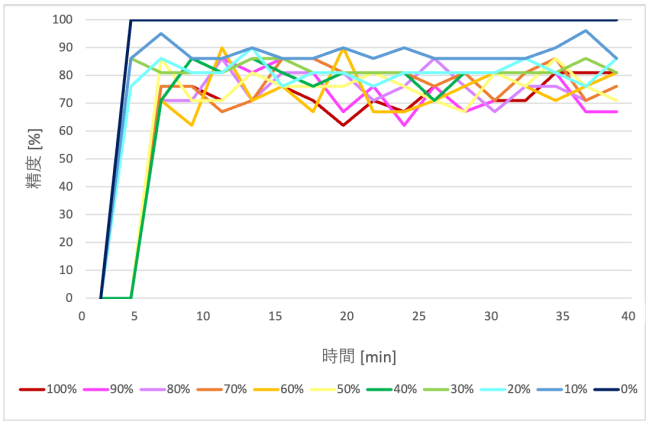

図 7　サーバから受け取る一般的なデータの割合を 0〜10 割まで変化させた際の個人情報のみに関する精度の比較

さらに，エッジサーバから学習を引き継がずにエッジデバイス上のみで学習を行った結果を青のグラフで示す．エッジデバイスの性能はエッジサーバに比べ低いため，精度が上がるまでにかなりの時間がかかる．赤で示しているエッジサーバから学習を引き継いだ際と同等の精度である 68%まで精度を上げるためには 100 分以上の学習が必要となり，エッジサーバの助けを借りることが有効だと分かる．

6.5　サーバから受け取るデータの割合と精度の関係

上記の実験ではエッジサーバの全てのデータをエッジデバイスへと送信しているが，エッジサーバ上にある全てのデータをエッジデバイスが受け取ることはエッジデバイスの容量やデータ送受信時にかかる通信コストの面から現実的ではない．そこで，エッジサーバから全てのデータは受け取らずに一部のみを受け取ることを考える．

エッジサーバからエッジデバイスへと送信するデータ量をエッジサーバのデータの 0〜10 割と変化させた際の結果を図 6 で比較する．

一般的なデータを一切含めずに個人情報のみで学習した 0 割の際は一般的なデータに対応できずに低い精度となり，エッジデバイス上での学習にも一般的なデータを含める必要があることが分かる．一般的なデータを 1 割，2 割と多少含めることで

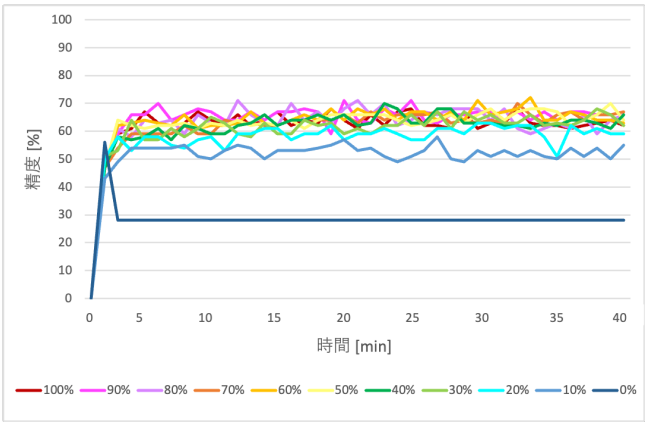

図 8　一般的なデータを 10 人とした際の全体の精度

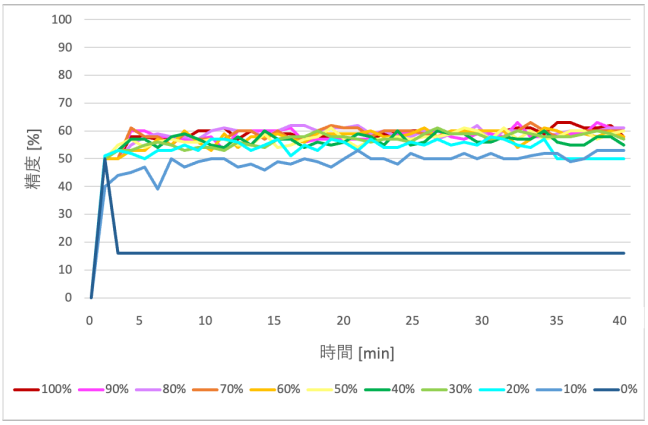

図 9　一般的なデータを 20 人とした際の全体の精度

精度がかなり上がり，3 割，4 割と一般的なデータを増やしていくとこのデータセットでの学習の上限である 68%近くまで精度が上がる．一方，9 割，10 割とデータを増やした場合であっても，精度は 3 割，4 割の場合とほぼ同じ結果となり，エッジデバイスの性能や画像を転送する通信コストを考慮すると全てのデータをエッジサーバからエッジデバイスへと送信することは好ましくないと考えられる．

また，一般的なデータの割合を 0〜10 割まで変化させて実験を行った際の個人情報のみに関する精度を図 7 に示すと，全ての割合において比較的良い精度となり，一般的なデータと個人データの両方に対応できていることが分かる．

6.6　データを変化させた際の結果

上記の実験では一般的な人物のデータを 32 人分，個人データを 1 人分用意しているが，ここではこのデータを変化させて実験を行い，汎用性の高い知見を得ることを目指す．

6.6.1　一般的なデータの人数を変化させる

一般的なデータを 10 人，20 人，30 人とし，それぞれエッジサーバから受け取るデータの割合を 0〜10 割まで変化させた際の結果を図 8〜10 に示す．ここで用いるデータセットは 6.4 節の実験で用いたデータセットから一部を抜き出したものであり，1 人物あたりの画像の枚数は 6.4 節と等しい．また，個人情報も 6.4 節と同じものを使用する．

全ての場合で，エッジデバイス上での学習において一般的な

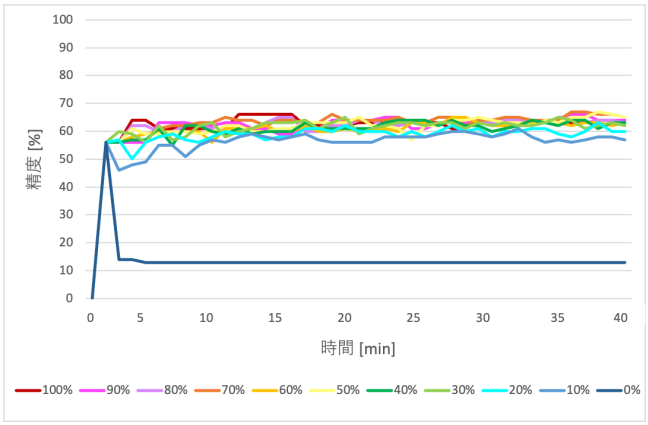

図 10　一般的なデータを 30 人とした際の全体の精度

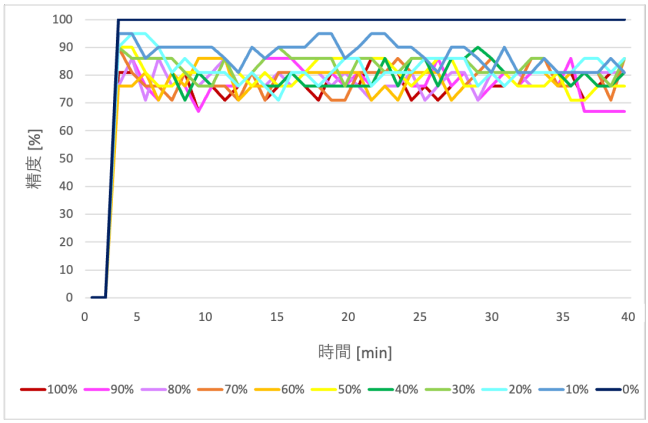

図 11　一般的なデータを 10 人とした際の個人情報のみに関する精度

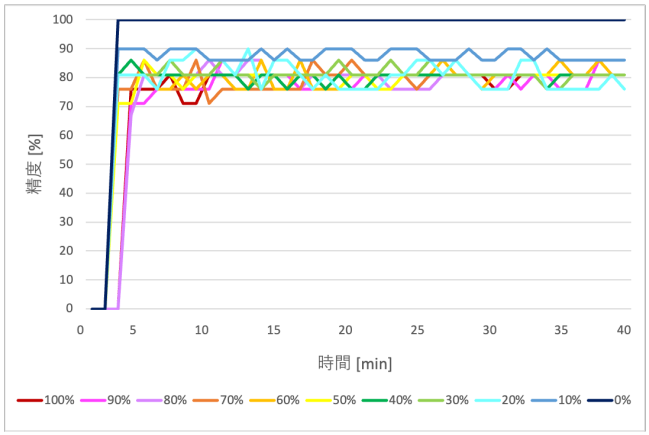

図 12　一般的なデータを 20 人とした際の個人情報のみに関する精度

データを全く含めなかった場合には一般的な test データに対応できずに低い精度となる．少し含めるだけでかなり精度が上がり，3 割以上ではそれぞれのデータセットの上限近くまで学習が可能となる．

また，個人情報のみに関する精度を図 11〜13 に示す．ここで精度が立ち上がるまでの時間に差があるのは，学習の際にデータ量が増加すると 1epoch あたりの step 数が増加し，学習結果が得られるまでに時間がかかるためである． 全ての場合において 70%以上と高い精度を得ることが可能である．

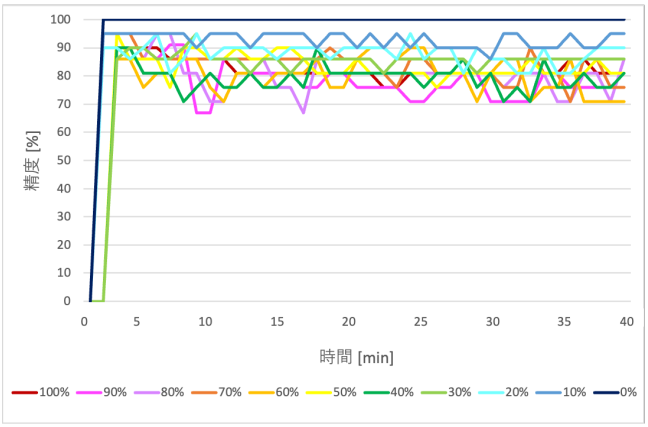

図 13　一般的なデータを 30 人とした際の個人情報のみに関する精度

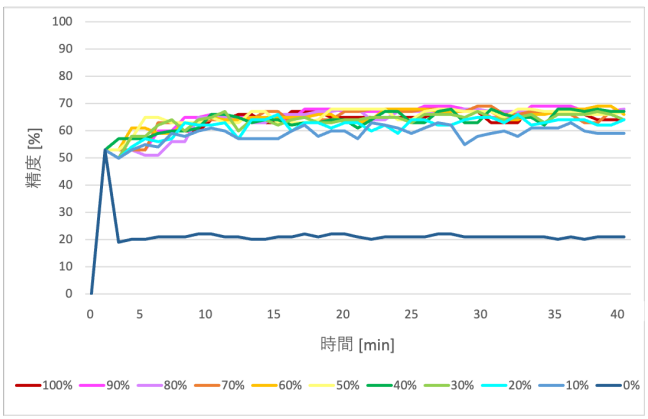

図 14　個人情報を 2 人, 一般的なデータを 31 人とした際の全体の精度

6.6.2　個人情報として扱う人数を変化させる

今までの実験では Tony Blair のみを個人情報として扱ったが，エッジデバイス上に複数人の個人情報が存在するケースも想定される．そこで，個人情報として扱う人数を 1 人，2 人，3 人とし，それぞれエッジサーバから受け取るデータの割合を 0〜10 割まで変化させる．この際，一般的な情報として扱う人数と個人情報として扱う人数の合計は全て 33 人となるようにする．また，使用するデータは 6.4 節に合わせ，個人情報として扱う人物の train データを 86 枚の顔画像をぼかし等で 9 倍に加工したものに置き換え，test データを 21 枚となるように置き換える．個人情報として扱う人数が 1 人の場合は図 6 に示すとおりであり，2 人及び 3 人の場合を図 14, 15 に示す．

この実験においても全ての場合で，エッジデバイス上での学習において一般的なデータを全く含めなかった場合には一般的なデータに対応できずに低い精度となるが，少し含めるだけでかなり精度が上がり，3 割以上ではそれぞれのデータセットの上限近くまで学習が可能となる．

また，個人情報のみに関する精度は，個人情報として扱う人数が 1 人の場合は図 7 に示すとおりであり，2 人及び 3 人の場合を図 16, 17 に示す．ここで精度が立ち上がるまでの時間に差があるのは，6.6.1 節同様，学習の際にデータ量が増加すると 1epoch あたりの step 数が増加し，学習結果が得られるまでに時間がかかるためである． 全ての場合において良い精度を得る

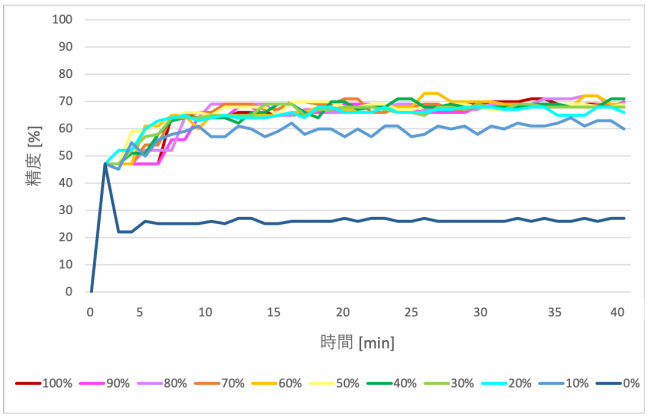

図 15　個人情報を 3 人, 一般的なデータを 30 人とした際の全体の精度

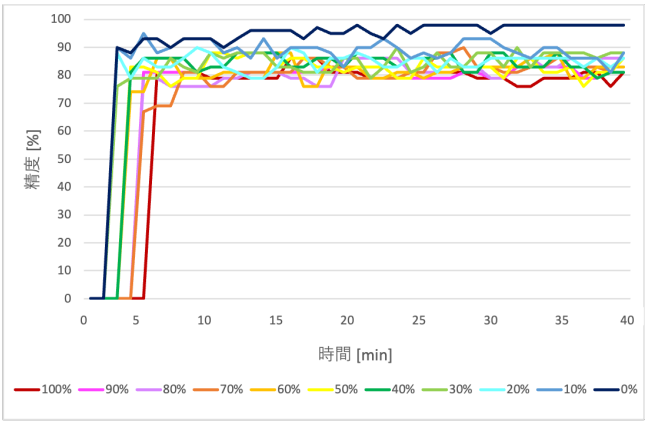

図 16　個人情報を 2 人, 一般的なデータを 31 人とした際の個人情報のみに関する精度

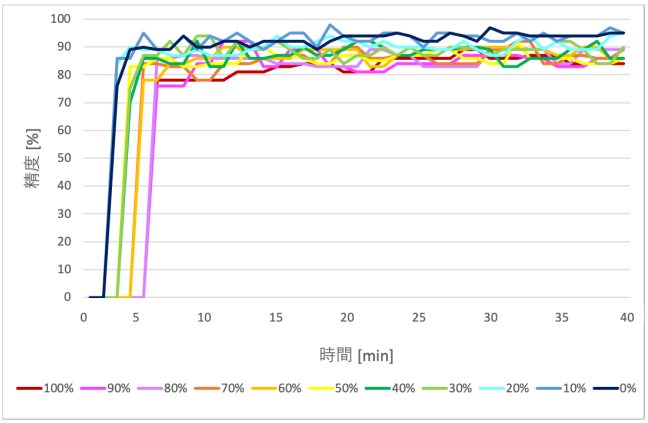

図 17　個人情報を 3 人, 一般的なデータを 30 人とした際の個人情報のみに関する精度

ことが可能である.

7　結　　論

以上の実験より，エッジサーバにおいて一般的なデータで学習を行い，エッジデバイスで学習を引き継ぐ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．さらに，エッジデバイス上での学習には一般的なデータを含める必要があり，このデータ構成においては，エッジサーバのデータのうち 1 割与えるだけでかなり良い精度となり，3 割与えると上限近くまで学習可能となる．性能の低いエッジデバイスのメモリ容量やデータ送受信時にかかる通信コストを考慮すると，エッジサーバより受け取るデータ量は 3 割程度で十分であると言える.

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.

エッジサーバ上での学習を引き継いでエッジデバイス上でも機械学習を動かす提案モデルにおいて，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り，エッジサーバの助けを借りることにより短時間で良い精度を得ることが可能であることが示された．また，今回のデータ構成においてはサーバのデータのうち 3 割程度を受け取るだけで良い精度を得ることが可能であることが示され，エッジサーバから一部のデータのみを受け取ることでエッジデバイス上で必要なデータ容量やデータ送受信時にかかる通信コストの削減が可能であると考えられる.

今回は画像データとして lfw から取得した顔画像を使用したが，今後はデータセットを変化させて実験を行い，様々な個人情報に対応したより良いモデル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.

謝　　辞

本研究は一部，JST CREST JPMJCR1503 の支援を受けたものである.

文　　献

[1] General Data Protection Regulation. https://eur-lex.europa.eu/eli/reg/2016/679/oj. (2021/04 閲覧).
[2] Labeled Faces in the Wild. http://vis-www.cs.umass.edu/lfw/. (2021/04 閲覧).
[3] 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.
[4] 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.
[5] 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.
[6] 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.

[7] 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.

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

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

[23] 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.

[24] 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.