Federated Transfer Learning: concept and applications

Sudipan Saha and Tahir Ahmad

Abstract

Development of Artificial Intelligence (AI) is inherently tied to the development of data. However, in most industries data exists in form of isolated islands, with limited scope of sharing between different organizations. This is an hindrance to the further development of AI. Federated learning has emerged as a possible solution to this problem in the last few years without compromising user privacy. Among different variants of the federated learning, noteworthy is federated transfer learning (FTL) that allows knowledge to be transferred across domains that do not have many overlapping features and users. In this work we provide a comprehensive survey of the existing works on this topic. In more details, we study the background of FTL and its different existing applications. We further analyze FTL from privacy and machine learning perspective.

Index Terms

Federated Learning; Transfer Learning; Machine Learning; Privacy-preserving.

I. Introduction

Currently machine learning is playing an important role in many applications, including commodity recommendation, risk analysis, and image analysis. This is due to its excellent capability to extract insights from data. Machine learning and deep learning techniques strongly depend on the availability of data [1]. Abundance of data has led to the overwhelming success of machine learning in image analysis [2], remote sensing [3], and many other fields. Traditionally, such machine learning or deep learning models are trained over a centralized corpus of data. While it is easier to collect image data, they are also more intuitive to label without strong domain

Sudipan Saha is with Technical University of Munich, Taufkirchen, Germany and Tahir Ahmad is with Fondazione Bruno Kessler, Trento, Italy. E-mail: sudipan.saha@tum.de, ahmad@fbk.eu

knowledge. Moreover, image data is generally not sensitive and shared without restriction among different parties. However, training data is not easy to obtain in some industries, e.g., finance and healthcare [4]. Labeling data from such industries require strong professional expertise. Moreover, data in such industries are generally protected by different privacy and security related restrictions [5]. Additionally, there exists practical risks of data abuse once the data are shared to the third parties. In such industries, data exists in the form of isolated island [6]. Competition between different organizations also hinder the sharing and thus exposing one's user data to another. Thus, such industries own insufficient data to train reliable machine learning frameworks.

Federated Learning (FL) can be used to overcome the above-mentioned constraints by using data from different organizations to train machine learning model, however not violating the different data related regulations. FL system was first proposed by Google in 2016 [7], [8], [9]. Their method was proposed for mobile devices that enables users to train a centralized model while their data are stored locally. Thus, federated learning technique can be used prevent the leakage of private information. While centralized learning needs to collect data from users and store them in centralized server, federated learning can learn a global model while the data are distributed on the users' devices. Many other works adopted the federated learning framework [10], [11]. This led to emergence of sub-groups within federated learning, e.g., horizontal federated learning and vertical federated learning [6].

A constraint imposed by the traditional federated learning is that training data owned by different organizations need to share same feature space. In practice, this is never the case in industries like finance or healthcare. To mitigate this shortcoming, Federated Transfer Learning (FTL) was proposed [12]. Different participants in FTL can have their specific feature space, thus making it suitable for practical scenarios. FTL takes inspiration from transfer learning, a paradigm already popular in image analysis [13]. In this setting, machine learning models trained on a large dataset for one problem/domain is applied to a different but related problem/domain [14]. The performance of transfer learning is strongly dependent on interrelation between different domains. While talking about federated learning, stakeholders in the same data federation are usually organizations from the same industry. Thus, it is suitable to apply transfer learning in the federated learning framework.

Federated transfer learning lies at the intersection of two different but fast-evolving fields: machine learning and information privacy. Thus it is an imperative to bridge the gap between them to fully exploit the benefits of federated transfer learning. This motivated us to investigate

into the different aspects related to federated transfer learning. It is important to understand how transfer learning and federated learning intermarried to give rise to federated transfer learning. It is critical to understand about how FTL has been applied in different real-life applications. It is important to understand the different privacy and machine learning aspects related to FTL. Keeping them in mind, in this paper, we present a comprehensive survey of the federated transfer learning. Towards this we: 1) outline the definition of FTL, 2) present some case studies on FTL, 3) analyze FTL from privacy aspect, and 4) analyze FTL from machine learning aspects.

We briefly discuss about horizontal and vertical federated learning in Section II. Federated transfer learning is defined in Section III. We detail the case studies on FTL in Section IV. We analyze FTL's privacy aspects in Section V and machine learning aspects in Section VI. Datasets used in the FTL related works are briefly presented in Section VII. We conclude the work in Section VIII.

II. Related work

For the scenario where datasets held by different users differ mostly in samples, federated learning can be categorized into horizontal and vertical federated learning. In this section we briefly review them. We also briefly review transfer learning, considering its relation to the federated transfer learning.

A. Horizontal federated learning

Horizontal federated learning is a system in which all the parties share the same feature space. However, their userbase may be significantly different. Such parties can collaboratively learn a model with help of a server. Each party locally computes training gradient and masks them with some encryption or privacy-preservation technique [15]. All parties send encrypted gradient to the server. The server aggregates them and sends the aggregated result to all parties. Parties update their model with the decrypted aggregated gradient and this way all parties share final model parameters [7]. An example of horizontal federated learning is a set of banks located in the same city or region and thus sharing same set of features, however very few common users. Horizontal federated learning can be used to learn a common model by agglomerating models learnt in individual banks. The horizontal federal learning can be implemented with different machine learning algorithms without changing the main framework.

B. Vertical federated learning

In the vertical federated learning, participating parties do not expose users that do not overlap among the parties [16], [6]. Overlapping users are found by using an encryption-based user ID alignment. Since different parties have different features corresponding to the common users, vertical federated learning aggregates different features from different parties and computes the training loss and gradients in a privacy-preserving manner [16]. Subsequently computed gradients are used to train the model. Vertical federated learning assumes honest participants and there is no hard requirement of a third party, as illustrated here [17]. However, sometimes to secure computations between the participants, an additional party is introduced [6]. An example of the vertical federated learning is case of cooperation between the online retailers and the insurers. They own their own feature space (and labels). However, their have significant amount of common users. In such cases, vertical federated learning exploits the situation by merging the features together to create a larger feature space for machine learning tasks [18].

C. Transfer learning

Most machine learning algorithms assume that the training data and the test data have same distribution and they are in the same feature space. However, this assumption does not hold in most real life scenario. In most practical cases, we have intend to analyze one domain of interest, while we have sufficient training data in another domain. As an example, we can consider the problem of classification of a product review [19]. Using traditional machine learning approach, sufficient number of product review needs to be collected and annotated for training. The distribution of review data varies from product to product and hence this training data collection and annotation process needs to be performed separately for each product. Separate data collection for each product is time-consuming and challenging. Transfer learning emerged as a framework to address this problem. Transfer learning provides a mechanism of training model on one product and reusing it on another product.

Transfer learning allows the tasks, domains, and distributions in the training and testing to be different. Pan and Yang [19] noted that research on transfer learning attracted more attention since 1995 and was tackled under different names, e.g., knowledge transfer, inductive transfer, and multi-task learning. Different approaches were adopted for transfer learning [19]:

1) *Instance transfer* re-weights labeled data in the source domain to reuse it in the target domain [20].

2) *Parameter transfer* discovers shared parameters between the source and the target domain [21].
3) *Feature-representation transfer* finds a feature-representation such that it reduces the difference between source and target domains [22].
4) *Relational-knowledge transfer* builds mapping of relational knowledge between source and target domain [23].

In the last decade, deep learning emerged as a very successful paradigm in the machine learning research. However, deep learning is even more data dependent than the previous machine learning algorithms. To tackle this, researchers started adopting deep transfer learning, to utilize knowledge from other fields by deep neural networks [24]. In addition to the four approaches defined above, another approach that gained significant attention in the deep transfer learning is adversarial-based deep transfer learning that introduces adversarial techniques based on generative adversarial network (GAN) [25] and its variants to find transferable representations applicable to both the source domain and the target domain. Deep transfer learning is closely related to other topics in the deep learning, e.g., deep domain adaptation [3]. One important line of investigation in the deep transfer learning is that which networks are more suitable for transfer and which features are transferable in the deep network [26].

III. Federated Transfer Learning

Federated transfer learning is a special case of federated learning and different from both horizontal and vertical federated learning. In federated transfer learning, two datasets differ in the feature space. This applies to datasets collected from enterprises of different but similar nature. Due to the differences in the nature of business, such enterprises share only a small overlap in feature space. This is also applicable to the enterprises set up far in globe. Thus in such scenarios, datasets differ both in samples and in feature space. Transfer learning techniques aim to build effective model for the target domain while leveraging knowledge from the other (source) domains. A typical architecture of federated transfer learning is shown in Figure 1. Considering two parties A and B, where there is only a small overlap in feature space and sample space between A and B, a model learned on B is transferred to A by leveraging small overlapping data and features. We recall that horizontal federated learning is used when there is large overlap in the feature space between datasets and vertical federated learning is used when there is large overlap in user/sample space between datasets. In contrast to them, FTL

is used when there is small overlap in both feature space and sample space (shown by dotted box in Figure 1). FTL ingests a model trained on source domain samples and feature space. Subsequently FTL orients the model for reuse in target space such that model is used for non-overlapping samples leveraging the knowledge acquired from source domain non-overlapping features. Thus FTL covers the region in right upper corner of the Figure 1 by transferring knowledge from non-overlapping features from source domain to the new samples in the target domain. The ability to use the transferred on non-overlapping data in A makes FTL different from vertical transfer learning.

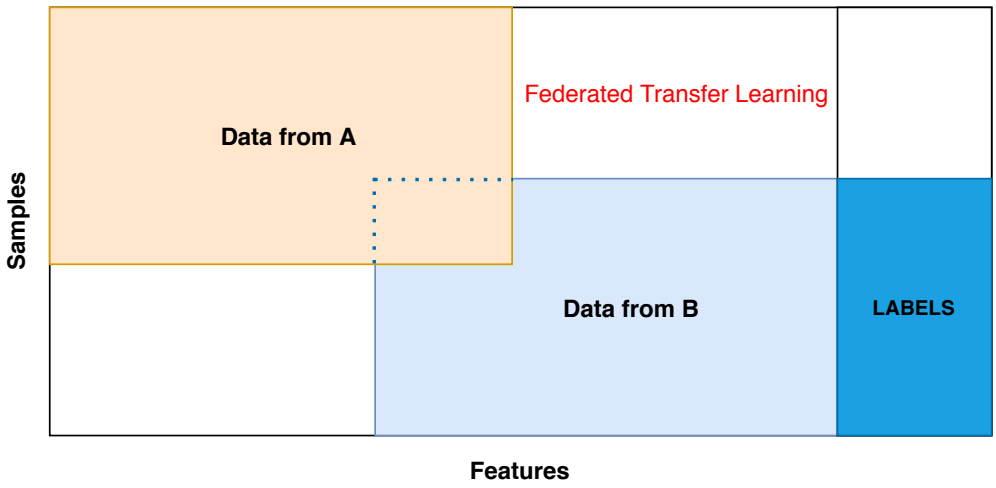

Fig. 1: Federated Transfer Learning [6]

IV. Use Cases of FTL

A. *Wearable Healthcare*

Wearable devices are fast becoming part of everyday life for patients and healthcare providers. Multiple features and functionalities of wearable devices include remote patient monitoring, tracking and collecting data, enhancing everyday health and lifestyle patterns, detecting chronic conditions, among others. Healthcare data, however, are usually fragmented and private making it difficult to generate robust results across populations. A typical architecture of wearable healthcare system is shown in Figure 2. It can be seen that the users use their wearable healthcare devices to measure various health related parameters they are concerned about. The user's data generated by healthcare devices often exist in the form of isolated islands. For further

assessment and analysis of results obtained the collected data is uploaded to the remote cloud based server [27].

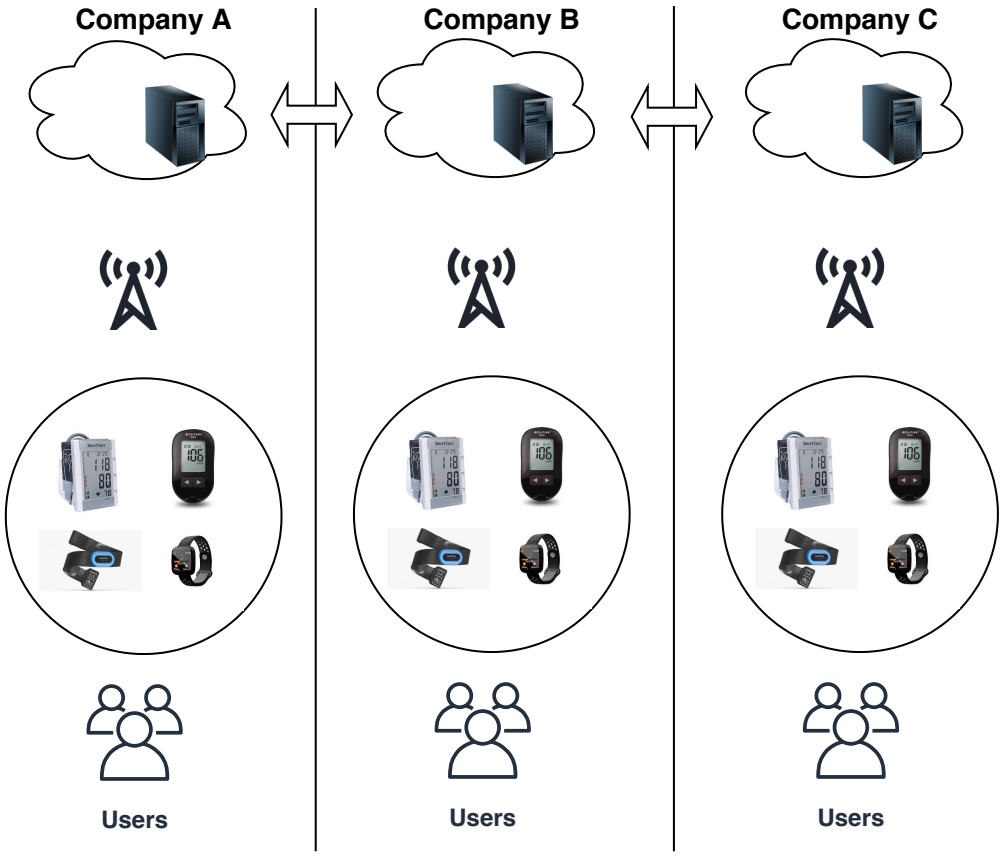

Fig. 2: Architecture of the wearable healthcare system

The architecture presents several critical challenges that hinders the development of effective analytical approaches for the generalizing of healthcare data. The adequate posture of healthcare data has several root causes including, (i) regulatory restrictions—lack of acquisition of massive user data, (ii) security and privacy—restricts sharing of data that exist in the form of isolated islands, and (iii) personalizing issue—the process of training machine learning model lacks personalization [28].

B. EEG signal classification

In addition to Section IV-A, another example of usage of federated transfer learning in healthcare domain is the electroencephalographic (EEG) signal classification [29]. Brain-Computer Interface (BCI) systems aim to decode participants' brain states. The success of deep learning

based BCI models for classification of EEG recordings is restricted by lack of large EEG datasets. Due to the privacy concern and high data collection expenses, EEG-BCI data is present in the form of multiple small datasets owned by different entities across the globe.

Towards this, Ju *et. al.* [29] proposes a method where the EEG data is represented as the spatial covariance matrix and is subsequently fed to a deep learning based federated transfer learning architecture. It is assumed that the architectures of each user's deep classifier is same. Federated averaging method [7] is adopted to aggregate models from different users. A server-client setting is used where a server acts as the model aggregator. In each round, the updated local models are sent to the server and server sends back the updated global model after aggregation. When a client receives the global model, it updates the model with its local data. This work [29] clearly demonstrates that use of domain adaptation in federated learning architecture boosts EEG classifier performance.

C. Autonomous Driving

Autonomous driving normally refers to self-driving vehicles or transport systems that move without the intervention of a human driver. As seen in Figure 3 autonomous driving technology is a complex integration of technologies including sensing, perception, and decision. The cloud platform (vehicular and Internet) provides data storage, simulation, high definition (HD) map generation, and deep learning model training functionalities. Autonomous vehicles are mobile systems, and autonomous driving clouds provide some basic infrastructure supports including distributed computing, distributed storage, and heterogeneous computing [30]. On top of this infrastructure, essential services can be implemented in the form of applications to support autonomous vehicles.

The dynamic nature of the autonomous driving environment and the uncertainty of real-life scenarios makes autonomous driving as a special use-case for FTL [31].

D. Image steganalysis

Image steganography is the technique of hiding information in the digital image without compromising its visual aesthetics. Image steganalysis is a counter technique to image steganography. It aims to detect the hidden information in the digital images. Towards this, it extracts and analyzes the steganographic features generated by image steganographic algorithms. There is a lack of data for training steganalysis methods due to the unwillingness of sharing data

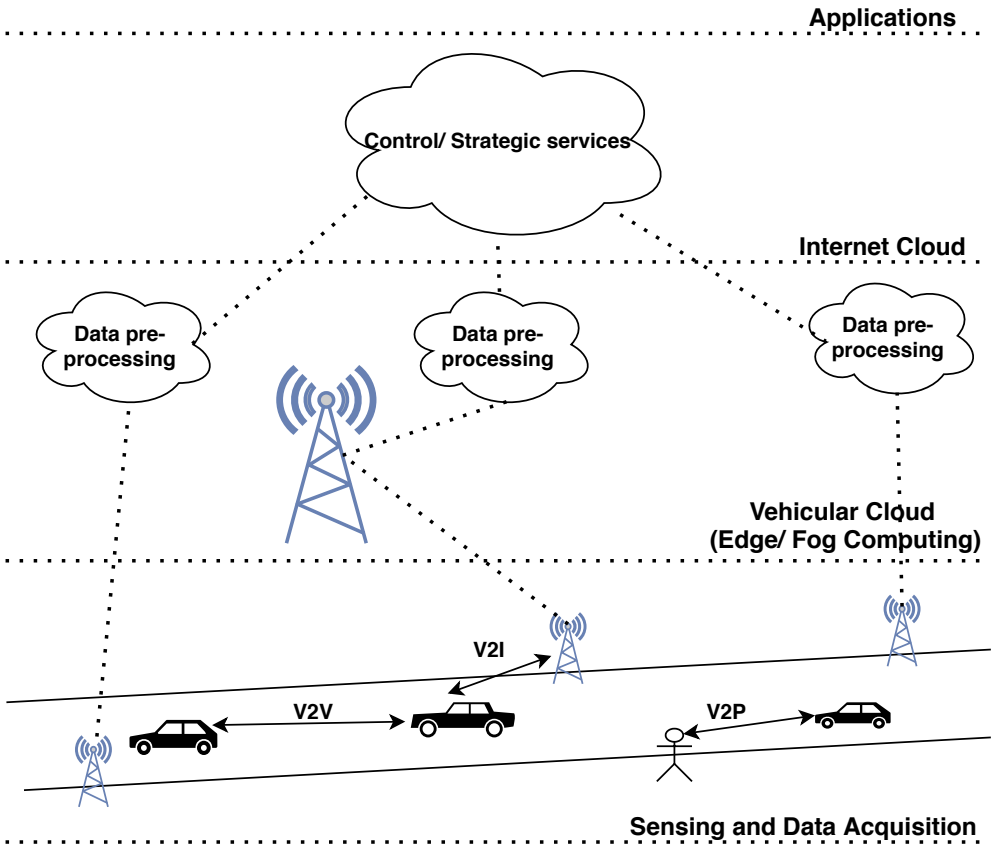

Fig. 3: Layered architecture of autonomous driving

among the steganographers. Furthermore, different data owners may have different preference for steganographic algorithms and cover images. The traditional image steganalysis algorithms cannot account for this personalized preference.

To overcome these challenges, Yang *et. al.* [32] proposed federated transfer learning framework for image steganalysis (FedSteg), as shown in Figure 4. In the proposed framework, different users have their own data (stego and cover images). The users do not leak the data to each other. Instead it is assumed that there is a cloud model and personalized local model for each user. The cloud model is trained with the data (cover and stego images) on the cloud-side. Then the cloud model is distributed to the users. Each user trains their local model with local data. The trained user models are sent back to the cloud side to help it train a new cloud model. This step only shares encrypted model parameters. In this fashion, each user can keep performing personalized training by consolidating the new cloud model with their previous model and its data [32]. However, there will be distribution discrepancy between the cloud and user data. To

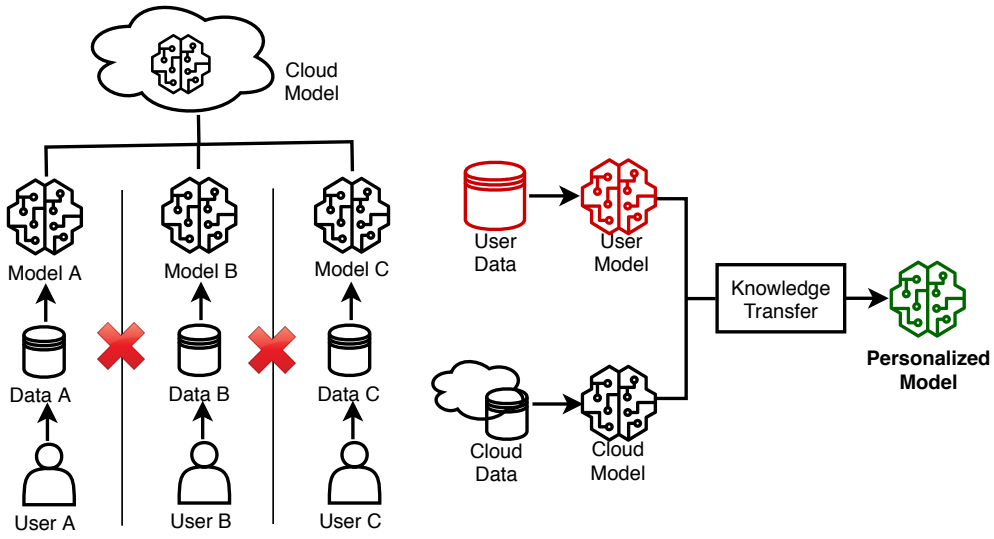

Fig. 4: Users A, B, and C denote steganographic images owner and they do not share any information (left). For personalized model training, each user individually interacts with the cloud model (right).

mitigate the distribution discrepancy, transfer learning is performed to make the model more suitable for user. Yang *et. al.* [32] uses a CNN based deep transfer learning to train personalized model for each user.

V. Privacy in FTL

Machine learning relies on the availability of vast amounts of data for training. However, in real life scenarios (as seen in Section IV), data are mostly scattered across different organizations and cannot be easily integrated due to many legal and practical constraints. Federated transfer learning (FTL) helps to improve statistical modeling under a data federation. The data federation as in case of FTL allows knowledge to be shared without compromising user privacy, and enables complimentary knowledge to be transferred in the network. Thus allowing the target domain party to develop a more flexible and powerful model by leveraging rich labels from source domain party [12].

The broad application of FTL is currently hindered by limited dataset availability for algorithm training and validation, due to the absence of technical and legal approaches to protect user's privacy. FTL has strict privacy preservation requirements, therefore, to prevent user privacy compromise while promoting scientific research on large datasets, the implementation of technical

solutions and development/implementation of legal frameworks to simultaneously address the demands for data protection and utilization is mandatory.

Privacy-preserving FTL typically involves multiple parties with emphasis on security guarantees to perform machine learning. Here, we present an overview of the necessary privacy-preserving approaches and techniques in context of FTL [33], [34].

- **Privacy by Design** FTL designed from the ground up with privacy in mind. The idea is taking into account privacy, throughout the FTL development process. The principles of privacy by design may be applied to all types of sensitive data and the strength of the implemented privacy measure must be dependent on the sensitivity of subject data. For example, processing of only necessary data, storage of data for minimal period, and limited accessibility. Optimal privacy preservation requires implementations that are secure by default and require minimal or no data transfer and provide theoretical and/or technical guarantees of privacy [35], [33].
- **Anonymization and pseudonymization** The former refers to the removal of personally identifiable information from a dataset (e.g., removing information related to age and gender), whereas, the later refers to replacement of personally identifiable information in a dataset with a synthetic entry with separate storage of the linkage record. For example, in case of health insurance companies wishing to reduce financial risk, re-identification of patient records are a lucrative target. Recently, data mining companies are adopting large-scale re-identification attacks and the sale of re-identified medical records as a business model [36]. Keeping that into account, the use of naive anonymization or pseudonymization alone must therefore be viewed as a technically insufficient measure against identity inference [33].
- **Differential privacy** The alteration of a dataset to obfuscate individual data points while retaining the ability of interaction with a data within a certain scope (privacy budget) and of statistical analysis. The approach can also be applied to algorithms. For example, randomization of data to omit relationships between individuals and respective data entries. It provides privacy preservation against membership-inference attack in the model inference stage [37]. However, during model training FTL approaches based on differential privacy are vulnerable to privacy leakage among the participants [38], [39].
- **Homomorphic encryption** A cryptographic technique that preserves the ability to perform mathematical operations on data as if it was unencrypted i.e., plain text. For example, per-

forming neural network computations on encrypted data without the need of first decrypting it. Homomorphic encryption is studied for private federated logistic regression on vertically partitioned data [16]. More recently, [40] proposed a privacy-preserving linear regression on horizontally partitioned data using homomorphic encryption and Yao's Garbled Circuits [41].

- **Secure multi-party computation** The technique is based on splitting data among collaborating entities to perform joint computation but prevents any collaborating entity from gaining knowledge of the data. For example, identifying the common patients among two hospitals without disclosing the respective hospital patient's list. Earlier works [42], [43], [44] mostly focus on approaches based on multi-party computation. SecureML [45], a privacy-preserving protocol combining secret-sharing [46] and Yao's Garbled Circuit [41], is considered as the state-of-the-art protocol for linear regression, logistic regression, and neural networks. SecureNN [47] is also proposed using a multi-party protocol for efficient neural network training.
- **Hardware security implementations** The approach to assure data and algorithm privacy by utilizing specialized hardware. For example, in the form of secure processors or enclaves implemented in mobile device [48]. Due to the rising significance of hardware-level deep learning implementations, e.g., tensor processing units [49]). It is likely that such system-based privacy guarantees built into edge hardware will become more common, e.g., trusted execution environments [33].

VI. Machine learning in FTL

Most works on FTL [28], [29], [32] have adopted variants of deep learning as the architecture for FTL.

In [28], the deep model is formed using a series of 1D convolution layers and fully connected layers. The model also consists of other auxiliary layers like pooling. Softmax layer is used for classification. A simplified schema of the deep model is shown in Figure 5. During the model from user to server, the convolution layers are kept frozen and the fully connected layers are updated to learn user and task-specific parameters.

A similar framework as above is used in FedSteg [32]. The model consists of 9 convolution layers followed by fully connected layer. While transferring model from user to cloud/server, the convolution layers are kept frozen, while the fully connected layer is used for model transfer. For feature alignment between source and target, correlation alignment (CORAL) loss is used,

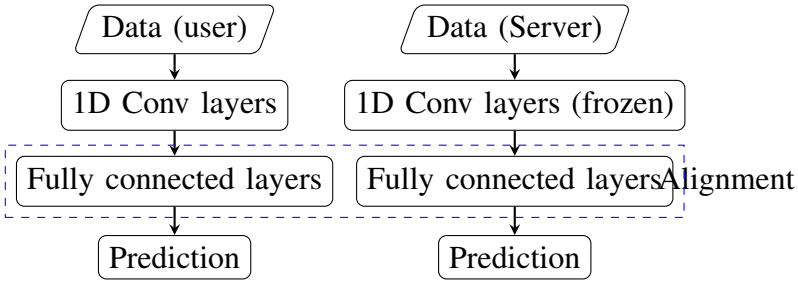

Fig. 5: The deep learning model for FTL [28]

which adapts the second order feature statistics [32]. The work in [29] uses both classification loss and domain loss that is implemented using maximum-mean discrepancy [50].

In federated autonomous driving [31], reinforcement learning is used. Reinforcement learning [51], different from other machine learning techniques, can be considered as a collection of observation and action spaces. In [31], each RL agent (user) is trained individually. Following this, knowledge of the distributed RL agents is distributed. Following that an online transfer process is performed to make alignments on the observations and actions.

In FTL, training with heterogeneous data may present additional challenge, e.g., not all client data distributions may be adequately captured by the model. Furthermore, quality of a particular local data partition may be significantly different from the rest. To overcome these challenges, Dimitriadis *et. al.* [52] presented a dynamic gradient aggregation (DGA) method which weights the local gradients during aggregation step.

Inspite of success of the methods discussed above, most of them do not provide an in-depth analysis of how different feature spaces can be handled in different users. Moreover, most of the above works are restricted to limited number of users. They are based on simplistic assumptions that convolution layers learn shared feature while fully connected layers learn domain-specific/ task-specific features. While correct, such assumptions do not fully exploit the tremendous progress made by deep domain adaptation community [53]. Furthermore, there are very few works that investigates the complicated machine learning issues that can arise in the heterogeneous FTL setting [52].

VII. FTL DATASETS

Most works on FTL have adopted existing machine learning datasets and modified them as per the requirement of FTL.

Fedhealth [28] used human activity recognition dataset - UCI Smartphone [54], consisting of 6 activities collected from 30 users within an age bracket of 19-48 years. To adapt the dataset for FedHealth, 5 subjects were regarded as isolated users (who cannot share data) and the remaining were used to train cloud model.

Ju *et. al.* [29] used PhysioNet EEG Motor Imagery (MI) Dataset [55]. The MI dataset is recorded from 109 subjects. Their experiments [29] use 5-fold cross-validation settings with 4 folds being used for training. Fedsteg [32] used two different image datasets. Gao *et. al.* [34] conducted experiments from several public datasets from UCI repository [56].

Differently from [28], [34] and [29], Liang *et. al.* [31] conducted real-life experiments on RC cars and Airsim.

VIII. Conclusions

Data isolation and data privacy concerns pose significant challenge for applying artificial intelligence techniques in real life applications. Moreover, features and users generally vary from organization to organization. In the last few years, federated learning, especially FTL has emerged as a potential solution to overcome the above challenges. In this work, we analyzed the concept of FTL and its applications in several practical domains. Furthermore, we did a detailed review of the machine learning techniques used in FTL. Our analysis shows that while FTL has explored different machine learning techniques, there is still potential of using more sophisticated machine learning techniques along with FTL. Adversarial techniques may provide a promising direction for further strengthen FTL. While FTL has been used in some practical applications, number of such applications are still few. Moreover, most of the datasets used in the FTL works are basic machine learning datasets and not tailored to real life scenarios. FTL holds the promise to break the barrier between different enterprises. However, there is scope of further advancement, by incorporating advanced machine learning techniques in FTL and applying FTL to more practical applications.

References

[1] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[3] S. Saha, F. Bovolo, and L. Bruzzone, "Unsupervised multiple-change detection in vhr multisensor images via deep-learning based adaptation," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 5033–5036.

[4] Q. Jing, W. Wang, J. Zhang, H. Tian, and K. Chen, "Quantifying the performance of federated transfer learning," *arXiv preprint arXiv:1912.12795*, 2019.

[5] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.

[6] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[7] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," 2016.

[8] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[9] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[10] H. Zhu and Y. Jin, "Multi-objective evolutionary federated learning," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 4, pp. 1310–1322, 2019.

[11] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2020.

[12] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intelligent Systems*, 2020.

[13] S. Saha, Y. T. Solano-Correa, F. Bovolo, and L. Bruzzone, "Unsupervised deep transfer learning-based change detection for hr multispectral images," *IEEE Geoscience and Remote Sensing Letters*, 2020.

[14] S. Saha, F. Bovolo, and L. Bruzzone, "Unsupervised deep change vector analysis for multiple-change detection in vhr images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 3677–3693, 2019.

[15] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[16] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," *arXiv preprint arXiv:1711.10677*, 2017.

[17] S. Yang, B. Ren, X. Zhou, and L. Liu, "Parallel distributed logistic regression for vertical federated learning without third-party coordinator," *arXiv preprint arXiv:1911.09824*, 2019.

[18] G. Wang, "Interpret federated learning with shapley values," *arXiv preprint arXiv:1905.04519*, 2019.

[19] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[20] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in nlp," in *Proceedings of the 45th annual meeting of the association of computational linguistics*, 2007, pp. 264–271.

[21] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge transfer via multiple model local structure mapping," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 283–291.

[22] A. Argyriou, M. Pontil, Y. Ying, and C. A. Micchelli, "A spectral regularization framework for multi-task structure learning," in *Advances in neural information processing systems*, 2008, pp. 25–32.

[23] L. Mihalkova, T. Huynh, and R. J. Mooney, "Mapping and revising markov logic networks for transfer learning," in *Aaai*, vol. 7, 2007, pp. 608–614.

[24] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International conference on artificial neural networks*. Springer, 2018, pp. 270–279.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[26] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[27] F. Sun, W. Zang, R. Gravina, G. Fortino, and Y. Li, "Gait-based identification for elderly users in wearable healthcare systems," *Information Fusion*, vol. 53, pp. 134–144, 2020.

[28] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, 2020.

[29] C. Ju, D. Gao, R. Mane, B. Tan, Y. Liu, and C. Guan, "Federated transfer learning for eeg signal classification," *arXiv preprint arXiv:2004.12321*, 2020.

[30] S. Liu, J. Tang, C. Wang, Q. Wang, and J.-L. Gaudiot, "Implementing a cloud platform for autonomous driving," *arXiv preprint arXiv:1704.02696*, 2017.

[31] X. Liang, Y. Liu, T. Chen, M. Liu, and Q. Yang, "Federated transfer reinforcement learning for autonomous driving," *arXiv preprint arXiv:1910.06001*, 2019.

[32] H. Yang, H. He, W. Zhang, and X. Cao, "Fedsteg: A federated transfer learning framework for secure image steganalysis," *IEEE Transactions on Network Science and Engineering*, 2020.

[33] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, pp. 1–7, 2020.

[34] D. Gao, Y. Liu, A. Huang, C. Ju, H. Yu, and Q. Yang, "Privacy-preserving heterogeneous federated transfer learning," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 2552–2559.

[35] A. Cavoukian and M. Prosch, "Privacy by redesign: Building a better legacy," *Information Privacy Commissioner Ontario*, pp. 1–8, 2011.

[36] A. Tanner, *Our bodies, our data: how companies make billions selling our medical records*. Beacon Press, 2017.

[37] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[38] Y. Wang, Q. Gu, and D. Brown, "Differentially private hypothesis transfer learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 811–826.

[39] Q. Yao, X. Guo, J. T. Kwok, W. Tu, Y. Chen, W. Dai, and Q. Yang, "Differential private stack generalization with an application to diabetes prediction," *arXiv preprint arXiv:1811.09491*, 2018.

[40] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *2013 IEEE Symposium on Security and Privacy*. IEEE, 2013, pp. 334–348.

[41] S. Yakoubov, "A gentle introduction to yao's garbled circuits," 2019.

[42] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *IEEE transactions on knowledge and data engineering*, vol. 16, no. 9, pp. 1026–1037, 2004.

[43] L. Wan, W. K. Ng, S. Han, and V. C. Lee, "Privacy-preservation for gradient descent methods," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 775–783.

[44] A. C. Yao, "Protocols for secure computations," in *23rd annual symposium on foundations of computer science (sfcs 1982)*. IEEE, 1982, pp. 160–164.

[45] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 19–38.

[46] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[47] S. Wagh, D. Gupta, and N. Chandran, "Securenn: Efficient and private neural network training." *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 442, 2018.

[48] A. P. Security, "Secure enclave overview," https://support.apple.com/guide/security/secure-enclave-overview-sec59b0b31f/web, 2020, accessed: 24-Sep-2020.

[49] Google, "Cloud TPU," https://cloud.google.com/tpu/, 2020, accessed: 24-Sep-2020.

[50] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Advances in neural information processing systems*, 2007, pp. 513–520.

[51] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017.

[52] D. Dimitriadis, K. Kumatani, R. Gmyr, Y. Gaur, and S. E. Eskimez, "Federated transfer learning with dynamic gradient aggregation," *arXiv preprint arXiv:2008.02452*, 2020.

[53] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 5, pp. 1–46, 2020.

[54] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*. Springer, 2012, pp. 216–223.

[55] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "Bci2000: a general-purpose brain-computer interface (bci) system," *IEEE Transactions on biomedical engineering*, vol. 51, no. 6, pp. 1034–1043, 2004.

[56] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.