Applied Intelligence
https://doi.org/10.1007/s10489-022-04065-3

Addressing modern and practical challenges in machine learning: a survey of online federated and transfer learning

Shuang Dai[1] · **Fanlin Meng**[1]

Accepted: 2 August 2022

Abstract
Online federated learning (OFL) and online transfer learning (OTL) are two collaborative paradigms for overcoming modern machine learning challenges such as data silos, streaming data, and data security. This survey explores OFL and OTL throughout their major evolutionary routes to enhance understanding of online federated and transfer learning. Practical aspects of popular datasets and cutting-edge applications for online federated and transfer learning are also highlighted in this work. Furthermore, this survey provides insight into potential future research areas and aims to serve as a resource for professionals developing online federated and transfer learning frameworks.

Keywords Online transfer learning · Online federated learning · Online learning · Federated transfer learning · Privacy-preserving

1 Introduction

Recent advancements in machine learning have propelled the broad utilization of smart technologies, particularly the Internet of Things (IoT). Worldwide, IoT devices are expected to nearly triple from 8.74 billion in 2020 to over 25 billion in 2030 [1]. On the one hand, massive data collected from IoT devices are critical for constructing robust machine learning models, and these have promoted the growth of innovations in the era of big data. Moreover, real-world machine learning advances rely on the availability of huge amounts of well-labelled data, such as ImgNet [2] and Alpha Zero [3]. On the other hand, big data, which are characterized by high volume, high velocity, and high diversity [4], cannot be utilized directly as high-quality ready inputs, posing significant challenges to the development of data-driven real-world machine learning systems.

In the era of big data, the challenges of developing data-driven machine learning systems differ radically from those of classic theoretical frameworks, owing to the inherent characteristics of big data and the restrictions imposed by data regulations and laws, such as the new General Data Protection Regulation (GDPR) [5]. These distinctions have important effects on the assumptions and performance indicators of data-driven machine learning systems and may stimulate the development of more innovative and practical machine learning algorithms. We begin this review by identifying the modern challenges in real-world machine learning and then present an overview of our survey. Finally, we consider how our survey contributes to the related fields.

1.1 Modern challenges in real-world machine learning

Machine learning has been widely applied to various real-world applications with satisfactory results. This survey identifies significant modern challenges in the era of big data and discusses their impact on developing real-world machine learning models at both the data and model levels, since a good machine learning model requires plentiful training data and a well-designed model.

From the data standpoint, high-quality datasets can provide more comprehensive information essential for building an effective machine learning model. However, in real-world machine learning applications, data may not be stored in a centralized location, and may exhibit statistical disparities [6] referred to as *data silos*. Medical

✉ Shuang Dai
sd19628@essex.ac.uk

Fanlin Meng
fanlin.meng@essex.ac.uk

[1] Department of Mathematical Sciences, University of Essex, Colchester, UK

Published online: 30 August 2022

records, for example, are private and are stored in isolated medical facilities; some facilities may only contain unlabelled data, whereas others may only hold a few labelled records. Moreover, *data labeling* is prohibitively expensive, particularly in fields requiring human skill and domain expertise, such as the medical sector. Therefore, the lack of labelled data is another obstacle for the development of real-world machine learning since the model performance is highly dependent on labelled data [2]. Also, data collection has become increasingly challenging from a legislative standpoint, which is referred to as *data governance*. For example, the GDPR [5] contains several provisions that protect user privacy and restrict companies from transferring data without explicit user consent. Moreover, the *real-time data* collected by IoTs enable more effective resource allocation and pose additional challenges to conventional offline machine learning frameworks that rely on pre-given training data. For instance, real-time traffic data on road conditions are collected and analysed to improve traffic management in smart cities, which necessitates a dynamic machine learning framework capable of handling streaming training samples [7].

From the model perspective, a well-designed model can make effective inferences and meet the needs of various tasks. However, the non-independent and identical distribution (non-IID) of data in the real world complicates the training of a single model that can be applied to all tasks. For instance, when the next-word prediction task is applied to a certain phrase, it should suggest a response tailored to each local user. Local users label the same data differently, necessitating the development of customized models [8]. As a result, *model personalization* is increasingly popular to meet the diverse needs of various users. Another current challenge in real-world machine learning is rapidly inferring a high-performance model for new users and effectively updating existing models, i.e. *constructing models effectively*. For instance, in a distributed system, conventional machine learning models that are based on a pre-given dataset must be retrained every time a new user joins, wasting both bandwidth and computing resources.

Various solutions have been proposed to address the aforementioned challenges, including online transfer learning (OTL) [9, 10], and online federated learning (OFL) [11]. OTL and OFL extend the concept of transfer learning (TL) [12] and federated learning (FL) [13] to the online context, allowing these advanced methods to process online big data efficiently. By leveraging knowledge from source domains, OTL aims to develop an online model for the target domain, addressing the challenges associated with predicting sequentially arriving data in the target domain due to a lack of well-labelled data. While OTL addresses the problem of data labeling in the online context, it still requires central access to data in both the source and target domains, which may violate data privacy and security standards in the era of big data. On the other hand, OFL focuses on training a central model using real-time data generated by multiple distributed local devices without violating data privacy regulations. During each training round, only the updated parameters of each local model are transmitted to the central model, ensuring performance of the centra model while maintaining data privacy.

1.2 Literature retrieval strategy and results

The selection of highly related sources and publications was based on standard criteria and protocols. The following three search engines and databases were chosen: (1) ScienceDirect (2) the Institute of Electronics Engineers (IEEE), and (3) Google Scholar. The literature searched ranges from work was conducted between January 2010 (OTL was first proposed in 2010 [9]) through to November 2021.

We selected the key phrases “online transfer learning” and “online federated learning” as our key phrases and included the synonyms as supplementary terms to expand our search results. As a result, the following key search strings are used for OTL during the literature retrieval stage:

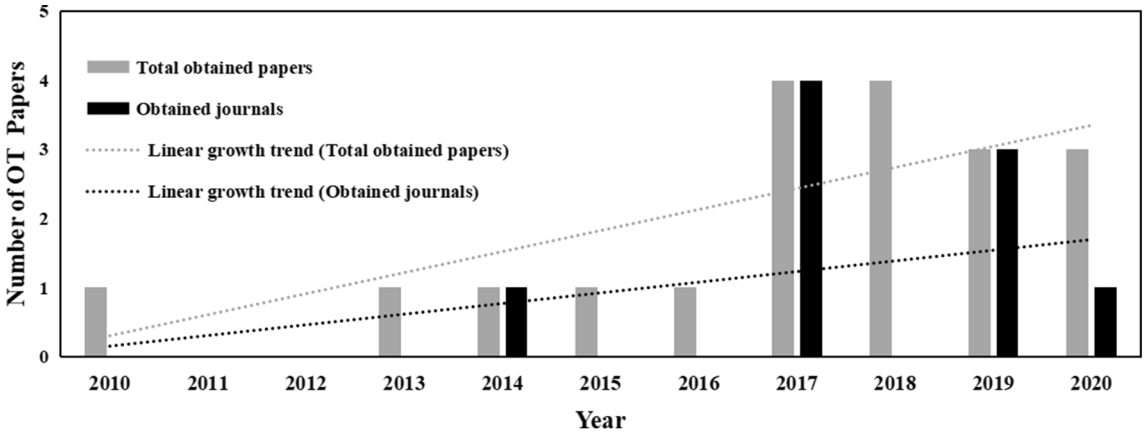
- Online/dynamic/adaptive transfer learning
- Online/dynamic/adaptive transformation learning

and the following key search strings are used for OFL during the literature retrieval stage:

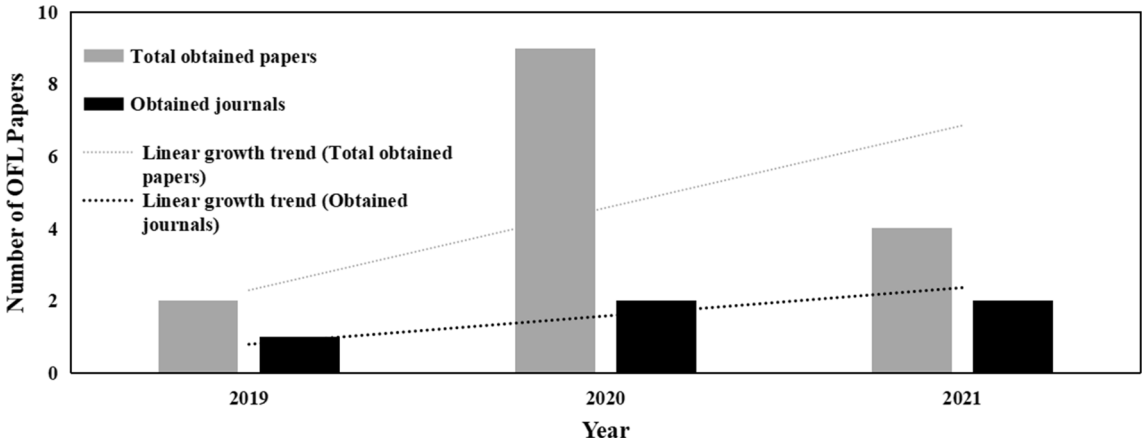
- Online/dynamic/adaptive federated learning
- Online/dynamic/adaptive federated machine learning

The keywords in each key search string utilize the Boolean operator ‘AND’; each key search string is connected with the Boolean operator ‘OR’. After excluding studies with incomplete titles and abstract information, 35 papers were included in this review. The obtained results consist of 20 OTL studies and 15 OFL papers. Figure 1(a) and (b) show the total retrieved articles and journals published from January 2010 to date (November 2021), along with their linear growth trends for OTL and OFL, respectively.

In summary, the development of OTL can be divided into two distinct phases based on the linear growth trend of the total papers obtained: the initial stage from 2010 to 2016, and the developing stage from 2017 to the present. While the field of OFL is still in its infancy, the lack of journal publications indicates that there is still a lot of potential for OFL research.



(a) The number of obtained papers for OTL.



(b) The number of obtained papers for OFL.

Fig. 1 Statistical results of the obtained papers

1.3 Overview of this survey

The purpose of this paper is to provide a detailed survey of various methods for addressing modern machine learning challenges, focusing on OFL and OTL.

Figure 2 illustrates the print of this survey. The green areas are our main emphasis, whereas existing surveys only concentrate on the yellow areas. The red section is one of the critical future paths we suggested for further investigation. We consider federated and transfer learning in online scenarios: OTL is not studied by traditional learning types of TL [12, 14], i.e. transductive, inductive, and unsupervised TL. Instead, we discuss OTL from two viewpoints: domain-based OTL and task-based OTL. Furthermore, we review OFL from three aspects: statistical heterogeneity, system heterogeneity, and privacy guarantees, highlighting the most significant challenges. The main contributions of our work are summarized as follows:

- To the best of our knowledge, this is the *first* survey to present recent advances in OTL and OFL studies, and to identify potential future research directions. It aims to serve as a resource for researchers and practitioners developing online federated and transfer learning frameworks.
- We provide definitions of OTL and OFL, as well as new viewpoints on them. Additionally, we describe recent advances in online federated and transfer learning, and highlight the connections between different methods.
- We summarize popular datasets and cutting-edge applications of OTL and OFL, discuss practical considerations and provide insights into potential future research directions.

The remainder of this survey is structured as follows. Section 2 reviews and reports on related works, which provides the necessary background on OTL and OFL. Then,

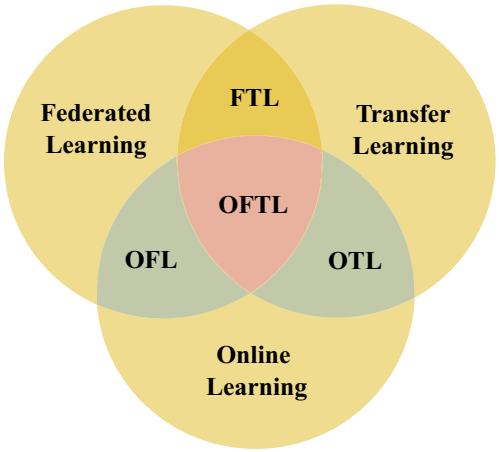


Fig. 2 Blueprint of our survey. FTL: Federated transfer learning; OFTL: Online federated transfer learning

recent advances in OTL and OFL are reviewed in Sections 3 and 4, respectively. Practical considerations in datasets and applications of OTL and OFL are summarized and presented in Section 5. In Section 6, we conclude this survey and discuss future research directions.

2 Related work

In this section, we review related work on OTL and OFL, including TL, FL, FTL, and OL. Moreover, we summarize the implementation scenarios of these methods and identify the existing challenges.

2.1 Transfer learning

Most of the traditional machine learning algorithms assume that the training and test data have similar distributions and feature spaces. However, this assumption does not hold in the majority of real-world scenarios. Furthermore, traditional machine learning has been hampered by a lack of adequately labelled training data and mismatched computing capability. TL [12] was proposed to address these challenges by leveraging knowledge from a single or multiple source domains to enhance a training task in the target domain (Fig. 3). The knowledge transferred could be instances from source domains [15], shared features from source domains and the target domain [16, 17], parameters from the trained learners of source domains [18], or relations between source domains and the target domain [19].

According to different implementation scenarios, TL can be categorized as single source TL and multiple sources TL. Single source TL refers to transferring knowledge from a single source domain [20] whereas the multiple sources TL utilizes several source domains to transfer the knowledge [21, 22]. Moreover, different TL techniques have been proposed to handle similar or different data structures between the source and target domains, i.e. homogeneous and heterogeneous TL [23, 24].

According to different label settings, a variety of TL methods have been proposed and can be classified into three major categories, i.e. transductive, inductive, and unsupervised TL [12].

Inductive TL is used when the target domain has well-labelled data, and there are different tasks in the source and target domains. TrAdaBoost [25] is a well-known inductive TL technique that extracts valuable information from the source domain by re-weighting predicted instances in both the source and target domains. However, this method only utilized a single source domain, and the extracted information may not be sufficient for the training task in the target domain. To address this challenge, [26, 27] combined the transfer task with multiple source domains, which enhanced the training performance of the target model. Moreover, unlike [25], which retained only one base learner and discarded the rest, [28] assumed that all base learners are useful, based on the theory that older learners can represent the major distributions of instances, while newer learners can provide accurate information about subsequent iterations.

Transductive TL is used when the source domain data is labelled, but the target domain data is unlabelled, and both the source and target domains have the same task. Domain adaptation is the most well-known subfield of transductive TL [29], which aims to minimize the marginal distribution gap between the source and the target domains. Xia et al. [30] proposed a method for selecting and weighting instances based on PU learning to identify examples from the source domain that are most likely to improve the training task. However, this method was limited by the difficulty of dealing with high-dimensional distributions. A solution was provided by [15], using the logistic approximation to adapt the high-dimensional data from the source domain to the target domain.

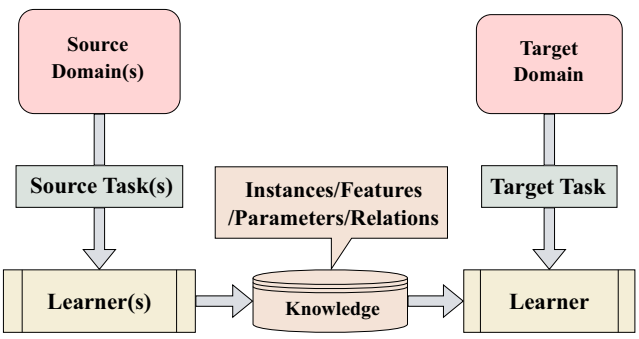


Fig. 3 Transfer learning process

In real-world situations, both the source and target domains may lack sufficient well-labelled data, which cannot be addressed by the TL techniques discussed so far. As a solution, unsupervised TL was introduced. Wang et al. [31] proposed transferred discriminative analysis (TDA), a method for generating class labels for unlabelled target data by leveraging knowledge from the source domain. Although unsupervised learning is a more practical solution in TL, it has received little attention from researchers over the last decade.

2.2 Federated learning

IoTs, such as smart healthcare devices and smart meters, continuously collect vast amount of data. Models trained from the aggregated data of these applications enable efficient management of smart city applications, however the process is complicated by a variety of legal constraints. In this context, FL has been proposed for training a global model from data distributed across multiple devices with only intermediate updates periodically being sent to a central server [13]. A typical FL paradigm is illustrated in Fig. 4, in which the central server distributes the initial model parameters to all local clients. Each client then trains the local model and uploads the updated parameters to the central server. After this, the global model will be updated and rebroadcast to local clients. The above processes are repeated continuously to ensure that the global model is updated and optimized across all local clients.

FL can be categorized into horizontal FL, vertical FL, and FTL, depending on how data are distributed among different devices in the sample and the feature space. Moreover, since FTL is known as a novel combination of TL and FL, we will discuss this technique in more detail in chapter (2.3).

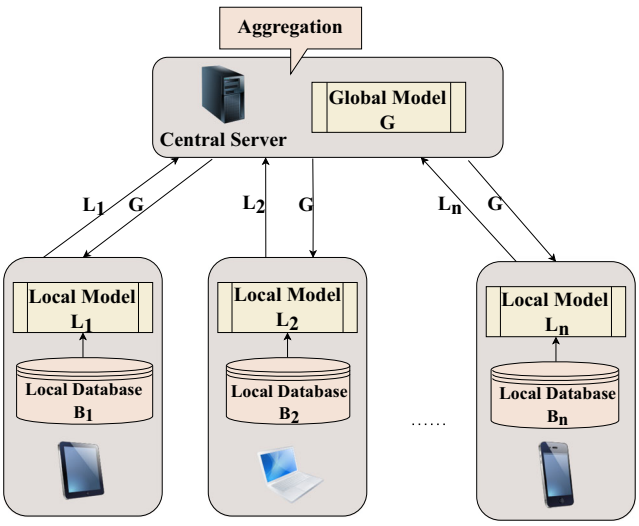


Fig. 4 Federated learning process

Horizontal federated learning (HFL) (Fig. 4) refers to the situation in which data from distributed devices share the same feature space but differ in samples. Google pioneered HFL by utilising data distributed across many local Android devices to forecast text input without violating privacy regulations [32]. Abad et al. [33] then developed a hierarchical heterogeneous HFL architecture for extending HFL to heterogeneous environments, thus optimizing the communication efficiency in local source devices with heterogeneous networks. Additionally, [34] designed a secure aggregation scheme based on [32] to further enhance the privacy of aggregated intermediate updates. Further research [35, 36] has been proposed to address the high cost of communication in the HFL framework.

Vertical federated learning (VFL) was proposed on the premise that heterogeneous data from various devices share common sample IDs but have distinct feature spaces, and thus VFL focuses on the correlation between devices from different sectors. In a typical VFL process, data with common sample IDs are retrieved and used to train a machine learning model (Fig. 5). VFL is more difficult to implement than HFL since it requires encrypted user-ID alignment algorithms [37] for common entities [13] and the authentication of a fully trusted third-party. To overcome these obstacles, [38] developed a framework that eliminates the need for a third-party coordinator, and this framework has proven to be efficient and scalable. Although VFL is capable of handling heterogeneous domains, the majority of VFL techniques rely on statistical models such as logistic regression rather than sophisticated machine learning frameworks, indicating that this field still demands enormous effort.

Apart from data distribution, FL can be categorized in a variety of ways. Based on the network topology, FL can be classified into centralized FL and peer-to-peer (P2P) FL [39, 40]. Centralized FL generally relies on a central server to aggregate and broadcast the updated parameters. In contrast to centralized FL, P2P FL does not rely on the central server for local model updates but instead exchanges parameters directly between neighbours. Based on data availability, FL can be classified into cross-silo FL and cross-device FL [41]. The cross-silo FL is suitable for scenarios involving a small number of local clients, in which siloed data are sourced from geo-distributed data centres (e.g. local banks or medical centres) instead of a large number of distributed edge nodes (e.g. smartphones or laptops). This is because almost every local client within the cross-silo FL is considered indexed and available for constant updating at any time. On the other hand, cross-device FL is used when there are a large number of participants and the local clients

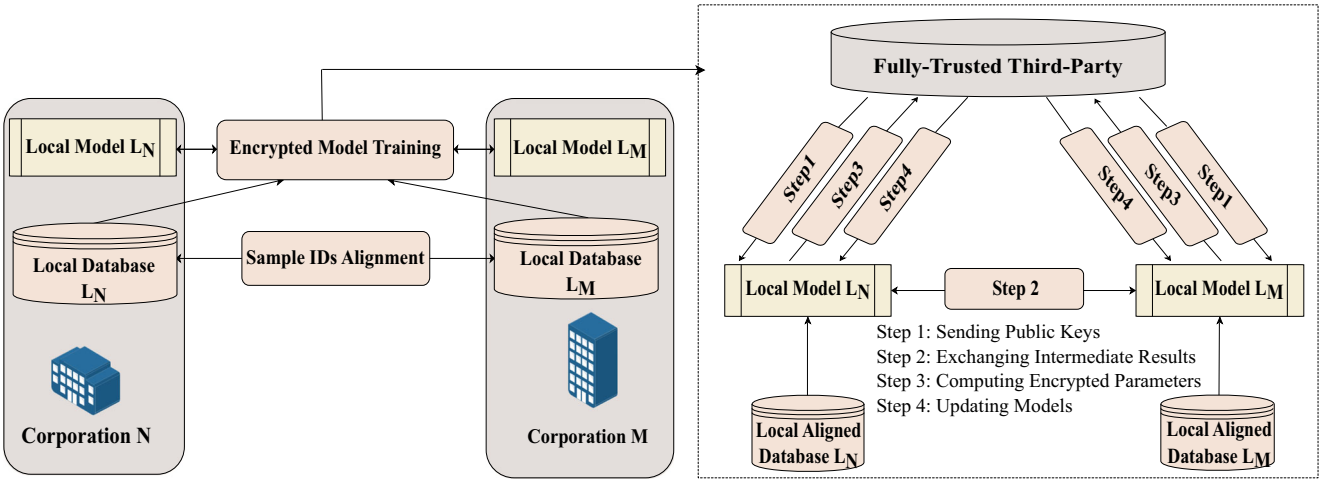


Fig. 5 Vertical Federated learning process

are not always available. To compensate for the unreliability of local clients, the cross-device FL often employs resource allocation techniques [42] and incentive mechanisms [43] to improve the overall performance of the FL framework.

2.3 Federated transfer learning

Distinguished from HFL and VFL, FTL [44] refers to situations where data across multiple devices differ in terms of both feature spaces and sample IDs and is regarded as a significant extension of traditional FL frameworks [13]. By enabling users to leverage large datasets with well-trained machine learning model parameters, FTL goes beyond simply allowing users to exploit only matching data (i.e. data with overlapped feature spaces or sample IDs) [45], and Fig. 6 depicts the general process of FTL. The use of TL in FL systems addresses the lack of well-labelled data in the source devices and enables various sectors to train more personalized local models in a secure and private manner. It is worth pointing out that while TL and FL are natural complements, there has been relatively little attention paid to the FTL framework.

Similar to conventional FL methods, the major impediment to FTL development is training data in heterogeneous settings, which is further complicated by the restrictive assumptions of FTL application scenarios. Gao et al. [46] developed a heterogeneous FTL framework to address the feature heterogeneity by mapping the feature spaces of common features to those of uncommon features. Moreover, to enable FTL in heterogeneous intelligent manufacturing applications, [47] utilized pre-built models from a variety of smart environments as the central source domain, and the central server then would select the best model to broadcast based on the similarity between the central source models and the local target models. Accordingly, each heterogeneous local device will conduct TL to acquire application-specific models. Furthermore, communication efficiency is another concern in FTL. In [48], secret sharing (SS) was adopted to improve the communication efficiency and to increase the privacy level of FTL.

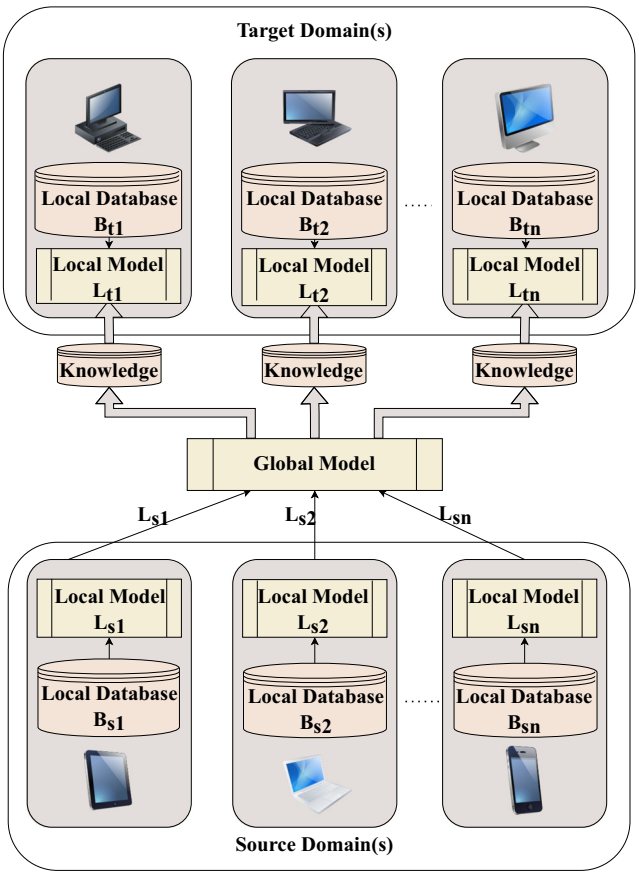


Fig. 6 Federated transfer learning process

FTL has received growing interests in real-world applications, such as smart healthcare [49], traffic monitoring [50], smart energy [51], and image analysis [52]. The majority of current FTL systems are based on deep learning architectures [47, 49, 51, 52] that usually freeze the base layers of the global model and retrain the fully-connected layer on local devices. Chen et al. [49] performed human activity recognition via FTL, which replaced one of the fully-connected layers with a correlation alignment layer to facilitate domain adaptation. FTL with deep learning architectures is efficient due to the highly transferable features in the low-level layers and the ability to capture specific features in the high-level layers of the deep network [53].

2.4 Online learning

OL is a machine learning paradigm for real-time data that uses feedback from sequence data to learn and update the best predictor for future data. In comparison to the optimal model in foresight, the primary goal of OL is to minimize cumulative error across the whole data sequence [54]. Compared to conventional batch learning algorithms, which require pre-given training data, OL is generally more effective and scalable when dealing with large-scale real-world machine learning problems involving data of varying quantity and velocity.

OL has been extensively investigated for many years [55, 56]. There are two fundamental types of OL algorithms: first-order OL and second-order OL. Hoi et al. [56]. The Perceptron [57, 58] is one of the earliest first-order OL algorithms, relying on gradient feedback to update a linear classifier whenever a new sample is misclassified. Passive-Aggressive (PA) [59] was introduced as a family of first-order OL algorithms based on margin-based learning. It updates the model when the classification confidence of a new sample falls below a predefined threshold. Moreover, online gradient descent [60–62] was proposed to model the OL as an online convex optimization problem.

The misclassified instances are retained as support vectors (SVs) in standard OL algorithms (e.g. Perceptron and PA). Despite their solid theoretical guarantees and efficient functioning, a fundamental issue is that the increasing number of SVs over time may result in an increased computational overhead. To overcome this challenge, [63] discarded the oldest SVs assuming that they were less representative of the data streams. Additionally, [64] presented bounded online gradient descent (BOGD) to constrain the amount of SVs that fall below a threshold.

Unlike first-order OL algorithms, which maximize convergence by utilizing only the first-order derivative information of the gradient, second-order OL algorithms maximize convergence by utilizing both the first-order and second-order information. The second-order Perceptron algorithm [65] was designed to examine the geometric properties of data. In order to capture second-order information about the confidence level of the features, the confidence weighted (CW) algorithm [66] was developed to manage the updating of the classifier. Furthermore, the second-order OL requires exponential space and time for updates, and the sketched online Newton (SON) [67] was introduced to address this issue. The SON is an enhanced version of the online Newton step with a linear running time in dimension and sketch size, allowing for dramatic improvements in second-order learning efficiency.

2.5 Frontier implementation scenarios and inter-connections of TL, FL, FTL, and OL

TL, FL, FTL, and OL are all innovative approaches built on standard machine learning techniques to address modern challenges in real-world applications. In this subsection, we will outline their implementation scenarios to investigate the underlying relationship between them and discuss the existing challenges. By doing this we hope to highlight the significance of our survey. Table 1 compares the implementation scenarios of traditional machine learning, TL, FL, FTL, and OL, which can be used as a guide to assist professionals in selecting the most appropriate method to apply to specific real-world problems.

Traditional machine learning relies on a massive amount of well-labelled centralized data and assumes that all data collected are homogeneous [29]. However, many real-world scenarios require a more scalable, private, and dynamic machine learning framework that can manage real-time data from a variety of IoT devices. TL, FL, and OL were therefore proposed as solutions to these modern challenges.

Although TL is rarely studied as a mechanism for knowledge transmission in a decentralized environment, when combined with FL, i.e. FTL, it is capable of transmitting knowledge across distributed devices. Additionally, TL in non-federated contexts typically involves instance transmission [15, 27, 68], posing a risk of privacy leakage. FL, on the other hand, preserves privacy [69, 70] by sharing local model update parameters instead of raw instances from local clients [71]. TL enhances target model performance by providing learners in target domains with a baseline performance rather than starting from scratch, thereby reducing computational overhead [72]. On the other hand, standard FL involves tens of millions or even billions of local devices [73], and all of these devices must meet eligibility computation power to participate in training, which is not practical, as demonstrated in [74]. As a result, it is logical to apply TL to this framework in order to enable FL with clients who have limited processing capabilities.

Real-world applications necessitate machine learning models to be resilient to heterogeneous data [41]. One of the

Table 1 Frontier implementation scenarios of different techniques

	Decentralization	Heterogeneity			Inadequate Well-labelled Data	Privacy-Preserving	Client-Side Personalization	Real-time Data
		Cross-Modality	Cross-Model	Cross-System				
Traditional Machine Learning	×	×	×	×	×	×	×	×
Transfer Learning	×	✓	✓	×	✓	×	✓	×
Federated Learning	✓	✓	✓	✓	✓	✓	×	×
Federated Transfer Learning	✓	✓	✓	✓	✓	✓	✓	×
Online Learning	×	×	×	×	×	×	×	✓

most challenging topics of heterogeneous scenarios is cross-modality [29], as it refers to situations in which the feature and/or label spaces of the source and target domains are completely different, which is one of the primary reasons for data heterogeneity in most real-world machine learning applications. The key idea in addressing this problem is to identify feature mapping functions that project the source and target feature spaces to a common latent space via matrix factorization [75] using labelled source data or co-occurrence data [76]. TL for cross-modality commonly transfers knowledge from easily labelled source domains to an expensively labelled target domain. An example is the well-known text-to-image TL [77], which leverages the semantic meaning of labelled text to improve model classification performance on sparsely annotated image data. Besides, VFL and FTL are also applicable to cross-modality scenarios. However, the former can only be used if certain conditions are met, i.e. having a large number of sample IDs that overlap between the source and target domains [13]. Additionally, while TL and FTL seek to leverage knowledge from source domains in enhancing the target model performance, the ultimate goal of VFL is to assist all source and target parties in developing a 'common wealth' strategy [13]. As shown in the table, TL, FL, and FTL can all be used in cross-modality scenarios, which explains why all of these strategies can help overcome challenges associated with the lack of well-labelled data.

Aside from cross-modality heterogeneity, FL is well-suited for cross-model and cross-system scenarios due to its decentralized nature. In cross-model scenarios, which are also prevalent in fundamental machine learning applications, the structure of the locally trained models varies due to the diverse patterns of data usage by local clients [78]. FL prefers to use the global model with a predefined model paradigm as the referencing information in a cross-model scenario, and clients can update their local models based on different structures [79, 80]. Ensemble strategies are frequently used to enable TL in cross-model scenarios, which combines multiple learners from different source domains or learning algorithms with a weight assignment strategy to maximize the utility of candidate learners that have better performance in the target domain [27]. Furthermore, most TL paradigms require all learners to be trained in a centralized and consistent environment whereas real-world situations are more complicated. On the other hand, FL is applicable in situations where there is system heterogeneity due to differences in storage, computing, and battery capacities between individual client devices. Xie et al. [81] developed an asynchronous FL framework (FedAsync) for adaptively updating the weights of local models in response to stale information, thereby enhancing FL in a more effective, flexible, and scalable manner.

Moreover, TL can personalize models in non-federated environments by leveraging data from source tasks to improve performance in a related target domain. However, when TL is applied to domains that are extremely unrelated, the model performance of the target domain could be worse than that of the source domain without transferring the source data, which is known as *negative transfer* [82]. Similar to this concept, when local clients come from highly unrelated domains or system settings, training local models in FL for these clients using a consistent scheme may reduce the ability of each local model to depict unique client characteristics [83], resulting in a worse aggregated model than local models trained exclusively on their own datasets, which can be recognized as a *drift* problem [84]. One of the most widely used strategies for mitigating negative transfer is to use effective selection mechanisms to determine the relatedness (also known as transferability [85]) between the source and target domains prior to the transfer [25, 77]. On the other hand, the drift problem is more complicated and can be approached differently. Rather than avoiding it, the majority of researchers have turned this into a feature [41] by applying various techniques such as multi-task learning to the FL framework [86–88]: they create *personalized* or *device-specific* local models [71] for clients that are intended to behave better than the aggregated global model.

In the new era of big data, a prominent application scenarios is modelling real-time data, which typically become obsolete within hours or even minutes [89], such as recommendation systems for business websites [90] and

real-time non-intrusive load monitoring systems for elderly living alone [91]. Additionally, there is a *cold start* [71] problem in real-world machine learning applications, which refers to new clients or datasets incoming into the system from the source domain. Existing TL and FL methods are generally based on pre-given datasets, which wastes bandwidth and computational resources due to the need to retrain the framework to achieve optimal results in the scenarios above [92]. Thus, it is vital to incorporate TL and FL into the OL paradigm to overcome these constraints. However, as this field is still in its infancy, few solutions have been proposed in recent years, and no prior research has summarized the research area comprehensively. To fill this review gap, following the understanding of the relationship between related techniques, the following sections will provide detailed descriptions and summaries of current OTL and OFL studies for further consideration.

3 Online transfer learning

OTL enables the standard TL paradigms to transfer knowledge from source domains, thereby enhancing the online learning task on the target domain [9, 10].

It is worth noting that the organization of OTL in this survey differs from the aforementioned traditional TL categories, as OTL is a developing field with research focusing on a more fundamental and specific perspective. The following sections provide an interpretation of OTL approaches from a domain-task perspective. In general, domain-based interpretation is based on different settings within the source domain, including single source (SS) OTL and multiple sources (MS) OTL. On the other hand, the task-based interpretation is based on different task types within the target domain, including binary classification (BC) OTL and multi-class classification (MC) OTL. While the majority of OTL research has concentrated on classification tasks, similar techniques can be applied to other machine learning tasks such as regression and clustering [10, 93, 94].

Figure 7 gives a relation map of OTL studies, which includes all the obtained OTL papers in Section 1.2. Specifically, the relation map consists of a root representing the cornerstone literature [9, 10] in which the OTL was first proposed, and four stems representing the four sub-areas of OTL, namely, SS-BC, SS-MC, MS-SC, MS-BC, with each stem node containing leaf nodes that represent literature focused on various technical aspects of each area. According to the relation map, most existing OTL

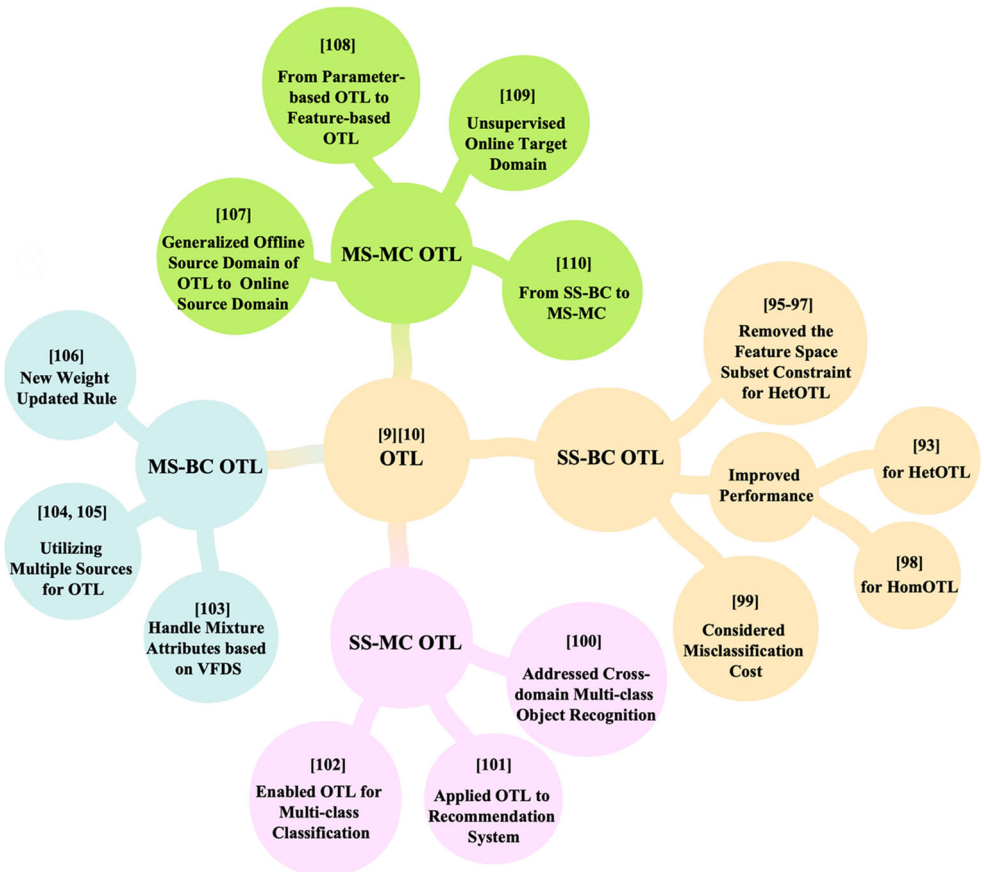


Fig. 7 Relation map for OTL. SS: Single source; BC: Binary classification; MS: Multiple sources; MC: Multi-class classification

studies have focused on SS-BC, MS-BC and MS-MC OTL, while studies for SS-MC OTL have been relatively scarce. Figure 8 summarizes the evolution timeline of OTL according to its sub-areas. The earliest research interests on OTL focused on SS-BC, and then several studies on SS-MC were proposed. After addressing the research difficulties in the single source domain, researchers started examining different approaches to OTL for utilizing information from multiple sources, i.e. MS-SC and MS-MC.

3.1 Notations and problem definition

Table 2 summarizes the frequently used mathematical notations in OTL, and we keep these notations consistent and similar to the majority of existing works [9, 10, 97, 104, 106, 108, 110] to facilitate comparisons between different OTL methods.

Given n source domains denoted by $D^S = \left\{D^{S_i}\right\}_{i=1}^{n}$, where each source domain D^{S_i} contains n^{S_i} labelled instances. The problem of OTL is formulated with single source (SS) task when $n = 1$, and with multiple sources (MS) task when $n > 1$. The source data space in the i-th source domain is denoted by $\mathcal{X}^{S_i} \times \mathcal{Y}^{S_i}$, where the feature space $\mathcal{X}^{S_i} = \mathbb{R}^{d_i}$. The target domain is denoted by D^T, with n^T instances. Similarly, we denote by $\mathcal{X}^T \times \mathcal{Y}^T$ the target data space in the target domain, where the feature space $\mathcal{X}^T = \mathbb{R}^{d_T}$. The problem of OTL is formulated with binary classification (BC) task when $k = 2$, and with multi-class classification (MC) task when $k > 2$. When $\mathcal{X}^{S_i} = \mathcal{X}^T$ and $\mathcal{Y}^{S_i} = \mathcal{Y}^T$, the problem is identified as homogeneous OTL (HomOTL). On the other hand, if the source and target domains have different feature spaces ($\mathcal{X}^{S_i} \neq \mathcal{X}^T$) or different label spaces ($\mathcal{Y}^{S_i} \neq \mathcal{Y}^T$), the problem is referred to as heterogeneous OTL (HetOTL) [9, 10, 111].

3.2 Single source-binary classification (SS-BC) OTL

SS-BC OTL was first proposed by [9, 10], which was considered in both homogeneous and heterogeneous scenarios (HomOTL and HetOTL). For HomOTL, as illustrated in Fig. 9, they first constructed the source model f^S using the offline source data by support vector machine (SVM) and utilized the Passive-Aggressive (PA) algorithm to build model f^T on the target domain. The PA formulated

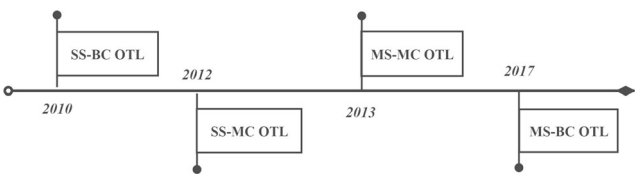


Fig. 8 Evolution timeline of OTL. SS: Single source; BC: Binary classification; MS: Multiple sources; MC: Multi-class classification

OL as a constrained convex optimization problem, and the weight ω of the online model on the target domain at a new time point $t+1$ was updated by the solution:

$$\omega_{t+1} = \omega_t + \tau_t y_t x_t \tag{1}$$

where $\tau_t = \min\left\{\mathcal{C}, \frac{\ell((x_t, y_t); \omega_t)}{\|x_t\|}\right\}^2$, and $\mathcal{C}$ is a positive regularization parameter. $\ell(\cdot)$ is the hinge loss, which can be written as $\ell((x, y); \omega) = \max\left\{1 - y(\omega^\top x), 0\right\}$. The resulting algorithm is *passive* and no update is needed when $\ell(\cdot) = 0$. Otherwise, when $\ell(\cdot)$ is positive, the algorithm is *aggressive* and the instance x_t will be selected as a support vector into the support vector set, which is then forced to learn ω_{t+1}. The PA standardized the trade-off between progress achieved at each new time point and information gathered in previous rounds [59].

After obtaining both the source and the target models, [10] proposed a weight updating scheme to adjust the weights μ of the source model and v of the target model, respectively:

$$\begin{cases} \mu_{t+1} = \frac{\mu_t s(f^S(x_t), y_t)}{\mu_t s(f^S(x_t), y_t) + v_t s(f^T(x_t), y_t)} \\ v_{t+1} = \frac{v_t s(f^T(x_t), y_t)}{\mu_t s(f^S(x_t), y_t) + v_t s(f^T(x_t), y_t)} \\ \mu_1 = v_1 = \frac{1}{2} \end{cases} \tag{2}$$

where μ_{t+1} and v_{t+1} are the weights of the source and target models, respectively, at time point $t+1$. $s(\cdot)$ is a weight decay function that increases the weights of models that contribute significantly to the final forecast.

Unlike [10], which only used a single source classifier, [98] proposed an AB-HomOTL inspired by the boosting algorithm to learn multiple weak source classifiers. As illustrated in Fig. 10, this paper focused on the learning strategy of the source model f^S in the homogeneous scenario for SS-BC OTL.

Specifically, AB-HomOTL selected PA as the primary learning algorithm for training m multiple weak source classifiers in the AdaBoost algorithm at the first stage. In the second stage, the source classifiers were integrated with the model f^T trained on the target domain. During this stage, a weight was assigned to each combination model based on its performance on the new instance (x^t, y^t). Finally, the ensemble models were integrated to produce the final robust target classifier f^t.

Rather than weighting classifiers dynamically according to their forecast accuracy, [99] emphasized that data in the real world are cost-sensitive and considered the misclassification cost to present an OTL algorithm with adaptive cost (OLAC). Specifically, they utilized the proportion of minority and majority samples to calculate the misclassification cost, enabling dynamic classifier adjustment for different samples. OLAC has been proven to be effective in improving the classification accuracy

Table 2 Summary of frequently used mathematical notations in OTL

Notation	Description
D^{S_i}	the i-th source domain
n^{S_i}	the number of instances in the i-th source domain
n, K	the number of the source domains/classes
D^S	the set of the source domains $D^S = \{D^{S_i}\}_{i=1}^n$
X^{S_i}	the feature space of the i-th source domain $X^{S_i} = \mathbb{R}^{d_i}$
$\mathcal{X}^T$	the feature space of the target domain $\mathcal{X}^T = \mathbb{R}^{d_T}$
$\mathcal{Y}^{S_i}$	the label space of the i-th source domain $\mathcal{Y}^{S_i} = \{1, 2, \ldots, k\}$
$\mathcal{Y}^T$	the label space of the target domain $\mathcal{Y}^T = \{1, 2, \ldots, k\}$
D^T	the set of target domain
n^T	the number of instances in D^T
(x^t, y^t)	the t-th arrived instance in the target domain
$f^{S_i}(\cdot)$	the model learned from the i-th source domain
$f^T(\cdot)$	the model learned from the target domain
$f^t(\cdot)$	the target model
$\mu_{t,i}$	the weight of the i-th source classifier at time point t
$v_{t,i}$	the weight of the i-th target classifier at time point t
n_c	the number of the co-occurrence instances
$(\widetilde{x}^{S_i}, \widetilde{x}^{T_i})$	the i-th unlabelled co-occurrence data

of minority samples, thereby increasing the overall model performance.

Zhao et al. [10] also considered the SS-BC OTL in the heterogeneous environment (HetOTL), which assumed that the feature spaces of the source domain are a subset of those of the target domain. Given a newly arrived instance (x^t, y^t), HetOTL divided it into two instances $(x^{t(1)}, y^t)$ and $(x^{t(2)}, y^t)$ where $x^{t(1)} \in \mathcal{X}^S$ and $x^{t(2)} \in \mathcal{X}^T/\mathcal{X}^S$. Then, inspired by the multi-view approach, HetOTL trained

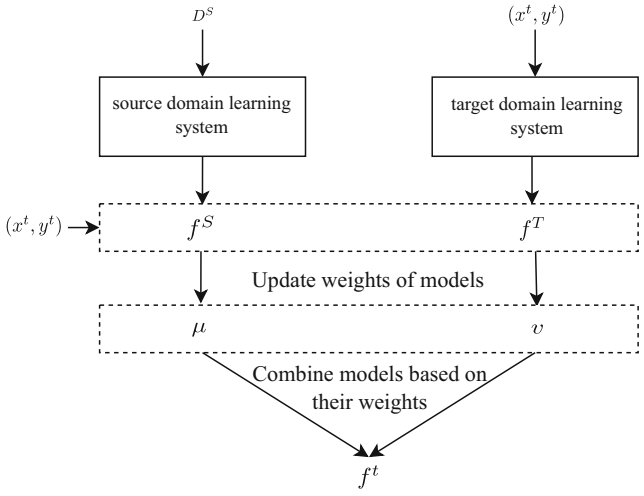


Fig. 9 SS-BC homogeneous OTL framework

and updated two classifiers $f^{T(1)}$ and $f^{T(2)}$ from two views simultaneously using the co-regularization optimization:

$$\left(f_{t+1}^{T(1)}, f_{t+1}^{T(2)}\right) = \underset{f^{T(1)}, f^{T(2)}}{\arg\min} \frac{\gamma_1}{2} \left\| f^{T(1)} - f_t^{T(1)} \right\|^2 + \frac{\gamma_2}{2} \left\| f^{T(2)} - f_t^{T(2)} \right\|^2 + \mathcal{C}\ell(f^{T(1)}, f^{T(2)}; t) \quad (3)$$

where γ_1, γ_2 and $\mathcal{C}$ are predefined positive regularization parameters, and $\ell(\cdot)$ is the loss function. During the updating, the classifier $f_1^{T(1)}$ was initialized by the trained source classifier f^S, and classifier $f_1^{T(2)}$ was initialized to 0. This updating rule ensured that the two-view classifiers did not deviate excessively from the previous updates (the first two regularization terms) while maintaining prediction performance (the last term).

Similar to [93, 98] proposed heterogeneous ensembled OTL (HetEOTL) based on AdaBoost to improve the performance of OTL models in the heterogeneous environment. The comparative experiment demonstrated that employing the ensemble strategy outperformed the previous HetOTL framework in [10]. Chen et al. [93] improved the performance of the OTL model, however it made the same assumption as [10], i.e. the feature spaces in the source domain are a subset of those in the target domain.

To relax the above assumption, studies based on co-occurrence data have been proposed [95–97]. Given a

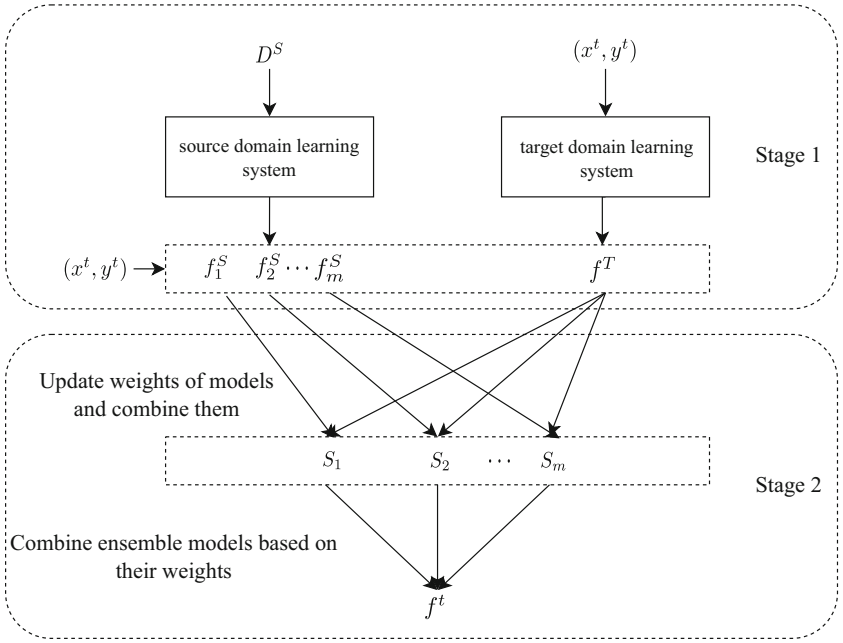


Fig. 10 AB-HomOTL framework

source domain D^S and a target domain D^T, whose feature spaces are totally diverse, i.e. $\mathcal{X}^S \cap \mathcal{X}^T = \emptyset$. The unlabelled co-occurrence data $\left\{\left(\widetilde{x}^{S_i}, \widetilde{x}^{T_i}\right)\right\}_{i=1}^{n_c} \in \mathcal{X}^S \times \mathcal{X}^T$ are collected from offline sources to bridge different feature spaces, in which $\widetilde{x}^{S_i} \in \mathcal{X}^S$ and $\widetilde{x}^{T_i} \in \mathcal{X}^T$. For example, the website Flickr[1] contains a massive collection of images with tags that can be used as co-occurrence data and are significantly less expensive to collect than labelled images (Fig. 11).

Yan et al. [96] proposed online heterogeneous transfer learning by hedge ensemble (OHTHE), which utilized co-occurrence data as auxiliary knowledge to build a correspondence map between the source and target domains, as illustrated in Fig. 12.

They first measured the heterogeneous similarity between the newly arrived instance x^t and the offline source instance x^s based on co-occurrence text-image data. The source model was then built by adding the weights of the k nearest neighbours of x^t in the source domain. Meanwhile, the target model was trained by PA. Following that, the OHTHE utilized the Hedge (β) strategy [113] to dynamically update the weights μ and v:

$$\begin{cases} \mu_{t+1} = \mu_t \beta^{\ell(y_t f^S(x_t))} \\ v_{t+1} = v_t \beta^{\ell(y_t f^T(x_t))} \\ \mu_1 + v_1 = 1 \end{cases} \tag{4}$$

where $\mu_1 \in (0, 1)$ and $v_1 \in (0, 1)$ are the initial weights. β is a weight decay factor that is used to identify models that contribute more to the final prediction and whose magnitude is determined by the loss function $\ell(\cdot)$.

[1] http://www.flickr.com

3.3 Multiple sources-binary classification (MS-BC) OTL

In real-world applications, it is difficult to extract sufficient knowledge from a single source domain, thus combining data from multiple source domains increases the reliability and robustness of source classifiers. However, combining all source domains directly may produce unsatisfactory forecasts since different source domains include information from different perspectives, and the data qualities within each source domain vary as well. As a result, OTL algorithms with multiple sources should be more sophisticated in order to distinguish critical source domains and thus construct a more robust source learner.

Wu et al. [105] trained a set of source classifiers using the kernel SVM, and each classifier was weighted according to its performance on the newly arrived instance of the target domain. The weighted source classifiers were then integrated to create an ensemble learner for the source domain. Simultaneously, PA was used to train the target classifier on the online data. The ensemble source and target classifiers were then integrated to generate an effective ensemble model in the second stage. The weight updating rule at the next round $t+1$ of the classifier from i-th source domain, the ensemble source classifier, and the target classifier can be described as follows:

$$\begin{cases} \mu_{t+1}^i = \mu_t^i \beta^{(f^{S_i}(x_t), y_t)} \\ \mu_{t+1} = \mu_t \beta^{(f^S(x_t), y_t)} \\ v_{t+1} = v_t \beta^{(f^T(x_t), y_t)} \\ \mu_1^i = \frac{1}{2n} \\ \mu_1 = v_1 = \frac{1}{2} \end{cases} \tag{5}$$

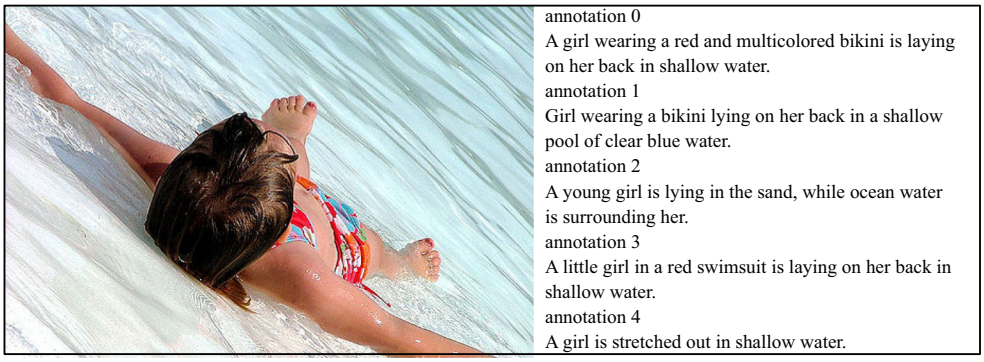


Fig. 11 An instance of co-occurrence text-image data from Flicker [112]

where $f^S = \sum_{i=1}^{n} \mu_t^i f^{S_i}(x_t)$. $\beta \in (0, 1)$ is a weight decay factor that is applied when the classifier suffers a loss value, and μ_t^i denotes the weight of the classifier from the i-th source domain at time point t.

In contrast to [105], which only investigated HomOTL, [104] adapted the OTL framework to a heterogeneous environment. Similar to the problem setting in [9, 10, 104] introduced heterogeneous OTL with multiple source domains (HetOTLMS), which was based on the premise that the feature spaces of the source domain are a subset of those of the target domain. Instead of training an ensemble source classifier, HetOTLMS combined the weak classifier from the i-th source domain with the target classifiers trained by PA to form n ensemble classifiers. In particular, for the i-th source domain in the t-th round, each newly arrived instance was divided into two parts, the first of which shared the same feature spaces as the source domain, and the second of which shared the remainder of the target feature spaces. Two classifiers in the target domain were generated and then integrated with the source classifier based on their weights to form an ensemble classifier.

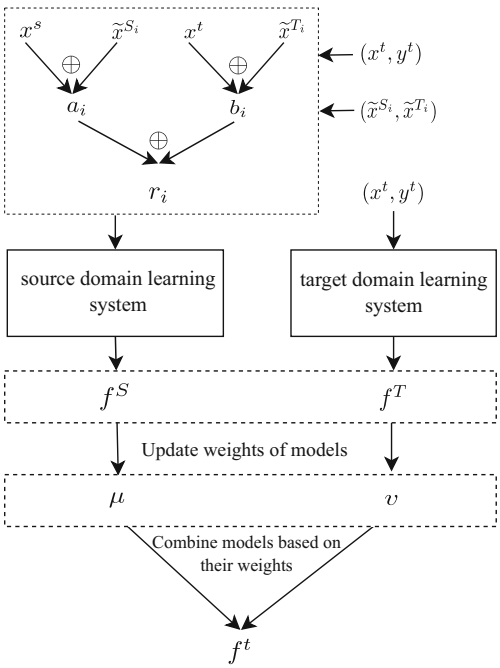


Fig. 12 OHTHE framework. The $\oplus$ maker denotes the measure of similarity between two instances

Most studies developed models based on PA that were limited to numerical attributes. Inspired by the very fast decision tree (VFDT), which incorporates Hoeffding bounds to guarantee the performance of an incremental decision tree, [103] modified VFDT as VFDT-D in the following ways to provide an OTL framework that handles mixed attributes:

- Cache a few instances to initialize the statistical information for newly constructed leaf nodes to satisfy the Hoeffding constraint and manage mixture attributes.
- Modify the output form of the VFDT to treat it as the posterior probability equal to the ratio of positive training instances in a leaf node with respect to the total number of training instances in that leaf node.

Then, using the VFDT-D, decision trees were induced from the source domains and the target domain. Following that, the tree path and posterior probability of the newly arrived instance x_t were then combined to determine the ideal source domain with the highest degree of similarity to x_t, which was then integrated with the target domain classifier to construct the final prediction decision function. Comparative experiments demonstrated that the proposed algorithm was capable of overcoming the cold start problem [71], which occurs when the model performance degrades in the early stage of the data stream due to the low number of instances arriving in the target domain.

It is worth noting that the target model performs worse than the source model as it lacks prior knowledge about the target domain. As more instances arrive, the target model will perform equally well or even better than the source

model. On the other hand, most studies [9, 10, 96, 105] updated model weights solely based on cumulative error, ignoring the intrinsic timescale of online data. To address this issue, [106] proposed a new weight updating rule which assigns greater weight to later occurrences. They assumed that the predictions made by the newer samples were more plausible than those made by the earlier samples and hence increased the weights over time to narrow the gap between the accuracy and the weights of the models. On the other hand, the traditional accumulating criteria ensure that the newly arrived outliers have a negligible effect on model updating, examining, therefore investigating whether the same scenario holds in this framework is necessary.

3.4 Single source multi-class classification (SS-MC) OTL and multiple sources multi-class classification (MS-MC) OTL

After reviewing binary classification OTL frameworks in the previous section, we will discuss multi-class classification OTL studies in this section. Multi-class tasks are common in the real world, such as document classification. Specifically, when an instance is relevant to a single subject, the classification problem is referred to as multi-class single-label classification; otherwise, the classification problem is referred to as multi-class multi-label classification [59], and the majority of existing OTL research has focused on multi-class single-label classification. Multi-class classification is more complicated than binary classification as it involves the development of offline and online models that consider multiple classes, necessitating the use of more sophisticated strategies to create a combined multi-class classifier with satisfactory performance [96].

Inspired by the online multi-class PA (MPA) algorithm [59, 102] presented an OTL algorithm for multi-class classification (OTLAMC) that adopted a novel loss function and weight updating mechanism to enable OTL in multi-class classification tasks. However, this paper only concentrated on knowledge transfer from a single source domain. Kang et al. [110] then developed the online multi-source transfer learning for multi-class classification (OMTL-MC) system, which incorporated data from multiple domains. While the OMTL-MC structure is similar to that of the HetOTLMS framework described in [104], there are two significant differences:

- The OMTL-MC framework examined OTL in a homogeneous environment, whereas the HetOTLMS framework investigated OTL in both homogeneous and heterogeneous settings.
- OMT-MC was developed with an extended Hinge loss (EHL) function to support multi-class classification tasks whereas HetOTLMS is only suitable for binary classification tasks.

Zhang et al. [100] proposed an online PA feature transformation (OPAFT) algorithm to calculate the similarity in a k nearest neighbour (k-NN) classifier. Furthermore, they extended this algorithm to the online multiple kernel feature transformation (OMKFT) algorithm to improve the performance of OPAFT for cross-domain and multi-class object recognition. Another feature-based OTL framework was proposed in [108], which investigated multi-class classification OTL with multiple source domains. Specifically, they constructed an initial transformation matrix for the i-th source domain by utilizing the source and target data. Then, the transformation matrix was used to project the original data into a new feature space. Meanwhile, the newly arrived instance was projected into its appropriate feature space using all of the transformation matrices, and a new source classifier was trained in this new space. The projected instance was then trained using the MPA algorithm to generate the associated classifiers for the target domain. Finally, the source and target classifiers were combined using the Hedge strategy. Rather than updating the transformation matrices at each time step, this paper used a time window to control the frequency of updates, thereby reducing computation costs.

In contrast to previous OTL architectures that required label revealing of target instances after each prediction, [109] introduced an online multiple source transfer learning (OMS-TL) architecture that requires only a few labelled data points in the target domain as a priori and does not require label revealing after each prediction. They employed a bipartite graph to represent the classification results from all the source domains and then estimated the likelihood of a sample belonging to each class using convex minimization. When a new instance is observed, the averaged probability from all source domain classes to which the sample belongs was combined with the target prediction based on the weighted average of previous predictions to generate the final result.

OTL aims to enhance the online learning task in the target domain by leveraging knowledge from source domains. By applying standard TL in the online context, real-time data generated by various edge devices can be efficiently processed. However, as with traditional TL, OTL is constrained by the assumption that all data from the source and the target domains must be processed centrally, which is impractical in the real world due to data

privacy regulations. As a result, the following section will introduce OFL, which enables real-time data processing in a distributed fashion while ensuring data privacy.

4 Online federated learning

FL holds significant promise for a variety of sophisticated applications, including smart traffic management [114], interactive social networks [115], and smart health monitoring [116], owing to the massive amounts of data generated by various edge devices (e.g. smartphones, wireless sensors, and wearable devices). On the other hand, Standard FL has been constrained to the premise that the training data at each local device is gathered offline and should be fully trained throughout each global round to deliver iteration round-efficient solutions [117]. It is impossible to assume that the training data at each local client remains constant throughout each round of training, as clients may have access to real-time data that will become obsolete in a matter of hours or even minutes [115, 118]. In this case, standard FL models will have difficulty capturing the fluctuations of real-time data, and their generalization performance is likely to decrease with an increase in training rounds. Therefore, enabling the standard FL architecture in online scenarios (i.e. OFL) is critical in the era of big data. Instead of delivering iteration round-efficient solutions by simply waiting for training results from all the local clients, OFL studies are increasingly focusing on the real-time data processing efficiency of local clients, i.e. on delivering iteration process-efficient solutions [117].

OFL considers that the data from each client is generated and collected in real-time, and it seeks to capture a high degree of temporal information from various distributions of data sources. Due to the time-varying nature of online data, several of the challenges associated with standard FL are becoming increasingly apparent in the online FL:

- Statistical heterogeneity: non-IID and unbalanced properties of online time-varying data cause model/concept drift [119] in OFL, and capturing the dynamic change of the rapidly generated online data poses a significant challenge to OFL.
- System heterogeneity: stragglers emerge due to device heterogeneity and network instabilities. Balancing the contribution of each local device to the local iteration against the communication cost of the global iteration is a critical challenge in OFL.
- Privacy guarantees: the vast amount of online data generated makes it more challenging to guarantee privacy for OFL. Various privacy protection strategies, such as differential privacy (DP) [120], have been implemented in FL in order to strike a balance between data utility and privacy, and these techniques should be optimized for the online environment to be more reasonable and practical for OFL.

Different OFL research focuses on different challenge priorities, and Table 3 summarizes current OFL studies on those three challenges.

The table demonstrated that the majority of OFL studies focused on statistical and system heterogeneity, while more research is required on privacy guarantees in OFL. Figure 13 summarizes the evolution timeline of OFL according to its sub-categories. It should be noted that OFL research is still in its infancy. In 2019, OFL began raising concerns about statistical heterogeneity and privacy issues, followed by studies exploring system heterogeneity in 2020.

4.1 Notations and problem definition

Consider we have a set of $\mathcal{K} = \{1 \dots K\}$ distributed devices. At each round t, the global server broadcasts its most recent parameters ω_g^t to the K devices. Each local device k receives the global parameters ω_g^t and a time-varying local instance (x_k^t, y_k^t) to update the parameters ω_k^{t+1} of its local model $f_k(\omega_k^{t+1})$. Finally, the local devices upload the updated parameters ω_k^{t+1} to the global server for dynamic aggregation:

$$\omega_g^{t+1} = h(\omega_1^{t+1} \dots \omega_K^{t+1}) \tag{6}$$

mapping h should be carefully selected in accordance with the model parameter structures [127], from which each device k can estimate the label y_k^{t+1} of a newly arrived data x_t^{k+1} in real-time. One of the most commonly used mappings in the standard FL system is *FedAvg* [32], which averages the aggregated local parameter sets:

$$\omega_g^{t+1} = \sum_{k=1}^{K} \frac{n_k}{N} \omega_k^{t+1} \tag{7}$$

where n_k is the number of the data samples taken on device k, and N is the total number of samples taken on K local devices. In OFL scenarios, the data is generated continuously on various local devices, increasing the uncertainty of the local model updating in comparison to the central model [122]. As a result, more plausible mappings should be established to constrain such variances and thus improve the generalization performance of the

Table 3 Summary of studies on OFL

Online Federated Learning	
Statistical Heterogeneity	[92, 119, 121–125]
System Heterogeneity	[11, 117, 122, 125–129]
Privacy Guarantees	[130, 131]

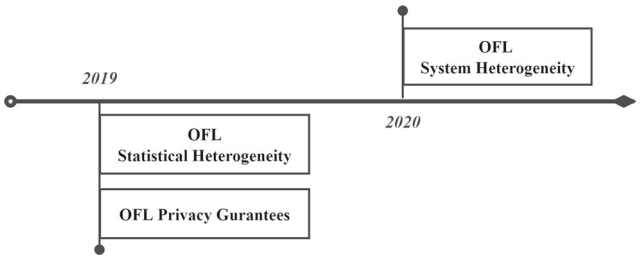


Fig. 13 Evolution timeline of OFL

model. Additionally, as not all devices are activated during each round t for a variety of reasons (e.g. due to network delays or device heterogeneity), strategies such as devices selection should be used to minimize the negative impact on overall communication efficiency.

4.2 OFL with statistical heterogeneity

Giorgas et al. [124] concentrated on the statistical heterogeneity associated with unbalanced data in OFL. Specifically, [124] assumed that the central server had been provided with pre-given data for training the initial central model. After initialization, the central model was broadcast to local devices for training with new samples from different classes. Then, the updated models on local devices were uploaded to the central server for integration. To ensure that the integrated model did not deviate substantially from the original central model, the integrated model may optionally be retrained using pre-given training data in the central server. This strategy effectively addressed common OL challenges, such as the catastrophic forgetting [132] that occurs due to the time-varying nature of online data.

To enable OFL framework in non-IID scenarios, [123] designed a non-linear regression OFL framework based on random Fourier feature-based kernel least-mean-square (RFF-KLMS). Specifically, they defined a non-linearly local model $f_k(\omega_k^t)$ for an arrived instance (x_k^t, y_k^t) of a local device k at time t. Then the local parameter updating function can be formulated as follows:

$$\omega_k^{t+1} = \mathbb{E}\left[\left|f_k(\omega_k^t) - \hat{f}_k(\omega_k^t)\right|\right] \tag{8}$$

where $\mathbb{E}[\cdot]$ is the expectation. $\hat{f}_k(\omega_k^t) = \omega_k^T z_k^t$, where ω_k is a linear representation of the non-linear model f_k in the random Fourier feature (RFF) space, and z_k^t is the mapping of x_k^t in the RFF space. Hence, the global parameter updating function can be further constructed in the RFF space by:

$$\omega_g^{t+1} = \frac{1}{K}\sum_{k=1}^{K}\omega_k^{t+1}. \tag{9}$$

Instead of training a global utility model for all local devices, some works have concentrated on improving the performance of personalized local models in the OFL framework. Based on the worker-leader-core network hierarchy, researchers designed hierarchical nested personalized federated learning (HN-PFL) for unmanned aerial vehicles (UAVs) [119]. The intra-UVA swarm is embedded inside an inter-UVA aggregation, which follows the worker-leader-core network structure to train high-level personalized models for local devices. To enhance the learning of HN-PFL, *model/concept drift* was introduced to quantify the dynamic changes of local online time-varying data. For a local device k with its local model $f(\omega_k^t)$, denote the online model drift at time t by $\Lambda_k^t \in \mathbb{R}^+$, which could capture the upper bound of the variation of parameters between two adjacent instances, we have:

$$\left\|\nabla f_k(\omega_k^{t+1}) - \nabla f_k(\omega_k^t)\right\|^2 \leq \Lambda_k^{t+1}. \tag{10}$$

Local models with a greater drift value are likely to become obsolete, necessitating a shorter learning period and more frequently revisiting. On the other hand, models with a small drift value have lower local parameter fluctuations, implying that they require less attention than models with greater drift values. The model drift value was then utilized to estimate the online gradient for each local model and a core network was created for each training sequence by storing the real-time properties of the network as reinforcement learning states. Additionally, to avoid the curse of dimensionality, they used a neural network to model the Q-table and determine the network states, rather than pre-building it using traditional reinforcement learning techniques [133].

Li et al. [92] emphasized the importance of developing individualized local models by combining multi-task learning with the OFL framework. Unlike previous works that analysed streaming data, [92] proposed an online federated multi-task learning framework (OFMTL) to address the problem of inferring effective local models for newly joined devices without affecting previous clients or the global server. The multi-task relationship learning [134] was used in the OFMTL to transfer the relationship between the local models of all the devices into a relationship precision matrix. The OFMTL formulated the learning of model parameters for the newly joined device as a convex optimization problem related to the weight matrix and the precision matrix, and an alternating optimization algorithm was proposed to alternatively optimize the model parameters and precision matrix of the new device by using information gained from previous devices. Additionally, to save computation resources while retaining the generalization performance of previous models, the model parameters were configured to be retrained only when the number of newly joined devices reached a fixed ratio with respect to the total number of previous devices.

4.3 OFL with system heterogeneity

Varying communication rates of heterogeneous local devices are also critical challenge for OFL, and the lagging devices with a lower communication rate in this system are known as *stragglers* [122]. Numerous solutions have since been proposed for standard FL systems. However, in real-world scenarios where the data on each local device fluctuates, the updated model for each global round may display more inherent dynamic features. Therefore, more sophisticated algorithms for FL in the online context (i.e. OFL) are required to minimize the negative impact of stragglers in a dynamic environment. Based on the reviewed papers, two types of protocols can be used to address the issue of stragglers: (1) synchronous protocol; (2) asynchronous protocol.

To deal with stragglers in the OFL system, [117] proposed an adaptive batch sizing (ABS) solution based on the synchronous protocol. Typical synchronous FL systems require the central server to wait until all local devices (including stragglers) have been updated before performing a global update [135, 136], or simply ignore and drop the stragglers [32]. Different from the above studies, ABS [117] limited the size of training data at each global round by allocating a batch size bound to each device based on their processing speed and real-time data generation speed, forcing all local devices, including stragglers, to be synchronous during each global communication round. Furthermore, ABS provided a buffer for each local device to retain or revisit local data depending on network settings to reduce volatility in the size of generated data during each training round. Despite a lack of mathematical definitions in [117], the proposed ABS structure in this paper is instructive.

Zhou et al. [125] proposed a cost-effective federated learning (CEFL) system capable of cooperatively reducing computation and communication overhead. Similar to [117], CEFL made dynamic decisions on local devices by limiting the entry of newly arrived training data, buffering, and scheduling the data according to the time-varying resource pricing of the local devices. Additionally, CEFL employed an additional optimization parameter to balance the computational overheads of local models with the overall communication cost.

Rather than leveraging all local devices for global model training, [128] highlighted that the key challenge for OFL is to distinguish between effective local models and to determine the appropriate number of local epochs using no prior knowledge. This study formulated the participant selection problem as an optimization problem based on system capacity (local device availability, data volume, and network bandwidth) and long-term convergence of both local and aggregated global models. To further extend this solution into online scenarios, an online schema was designed (Fig. 14) to dynamically select the participant and the number of local epochs for each device in the OFL system. Specifically, the online schema consists of two parts: online learning and online rounding. The first part produced fractional judgments solely based on prior knowledge, whereas the second part employed a compensation technique to randomly convert the fractional decisions to integers without violating any pre-defined constraints. The experiment results indicated that the proposed schema could dynamically adjust the upper bound on local convergence accuracy and select participants with superior local model performance and computation efficiency.

Asynchronous system design was also emphasized in studies addressing the heterogeneity of the OFL system.

Chen et al. [122] developed an asynchronous OFL framework (ASO-Fed) that enabled a wait-free OFL system and improved the prediction performance and computation efficiency of local devices when data arrived continuously. ASO-Fed learned the inter-client interaction on the global server using feature representation learning inspired by attention mechanisms [137, 138] and weight normalization

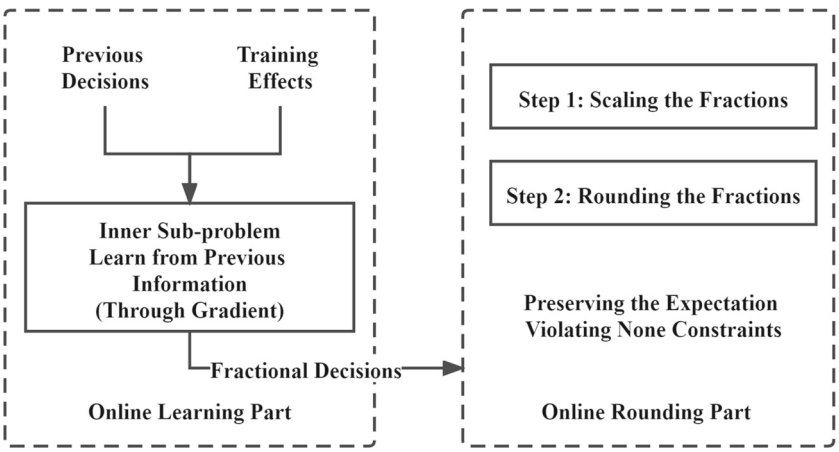


Fig. 14 Structure of the online schema proposed in [128]

[139, 140]. The decay coefficient was utilized on the client-side to balance the older and newer models when OL was performed on each local device. Additionally, ASO-Fed used a dynamic step size to minimize the negative impact of stragglers. The step size was determined by the data volume and communication capacity of each client, and a larger step size was assigned to local clients with a lower activation rate to compensate for the long latency and achieve higher performance. Experiments demonstrated that the proposed ASO-Fed framework converged faster than synchronized FL frameworks and significantly reduced the overall computational overheads.

On the other hand, [11] emphasized the importance of incorporating contributions from all local customers, even stragglers. They developed FLEET, which consists of two components, I-PROF and ADASGD. The first component aims to forecast and allocate computational overheads across all local devices. The latter is a novel stochastic gradient descent algorithm that employs weighted stale gradients determined by a stale-aware dampening factor and a similarity-based boosting value. The stragglers with longer delays were assigned a smaller stale-aware dampening factor, indicating that they contribute less to the overall updating process. In contrast, a lower similarity value indicates a gradient containing more significant new data features. The FLEET has been proven to be effective in minimizing the negative effects of stragglers while also capturing vital information to improve the generalization capability of the system.

4.4 OFL with privacy guarantees

Since the data are generated in an online fashion and the sequence of training data is unknown, ensuring privacy for the OFL framework requires a more sophisticated design of the privacy algorithms. Odeyomi and Zaruba [130] considered P2P FL in an online setting, and proposed an online mirror descent algorithm with long-term constraints on the sequential decisions made by each device. Additionally, a modified online version of local differential privacy was utilized to ensure the privacy of the OFL system. By using only the private version of loss gradients for real-time data sequence at each global round, the online local differential privacy method provides global privacy guarantees without relying upon loss information across the entire data sequence. In each global training round, each user received new data and updated its local model. After that, each updated local gradient was subjected to local differential privacy to ensure privacy in the online scenarios. When compared to the online gradient descent algorithm with differential privacy, the proposed algorithm was proved to be more accurate in the long run.

In large-scale online distributed network settings, the dynamic growth of the online dataset complicates the process of incorporating noise into each associated data sequence to ensure privacy. Zhou et al. [131] utilized a trusted third party to protect the privacy of OFL in a recommendation system based on adaptive binary tree-based noise aggregation. They constructed a binary item-cluster tree for each local device to reduce the scale of incoming online big data at each global round. Specifically, the item space was partitioned into refined child clusters, and the optimal recommendation for the corresponding client was searched from top to bottom of the constructed tree. Then, to ensure privacy, a trusted third party was proposed as a middleware to provide safe model aggregation over all agents using an exponential mechanism, and two forms of attacks from internal local devices and external adversaries were evaluated to demonstrate the usefulness of the approach.

OFL enables real-time distributed data training while maintaining data privacy, and much of the current literature on OFL focuses on addressing statistical and system heterogeneity rather than providing privacy guarantees. Compared to OTL studies, there are relatively few historical studies of OFL. In the following section, we will describe OTL and OFL from a practical perspective based on our discussion of the methodology in the previous two sections.

5 Practical aspects in online federated and transfer learning

Although studies of OTL and OFL have been conducted with promising results in a variety of fields in recent years, there are still practical concerns that need to be addressed. This section discusses the practical issues associated with online federated and transfer learning from two perspectives: datasets and applications.

5.1 Datasets

In this section, we first summarize all the datasets based on the obtained literature on OTL and OFL. Then we will discuss practical considerations and concerns around datasets for OTL and OFL.

The commonly used datasets for OTL and OFL are listed in Tables 4 and 5. In addition, there are special datasets used in some particular studies, which are described as follows. Five papers have adopted special datasets in the obtained OTL studies. [99] used various datasets in different scenarios: *landmine* dataset[2] for landmine detection, *wdbc*

[2] http://www.stevenhoi.org/OTL/

Table 4 Summary of the commonly used datasets in OTL studies

Dataset for OTL	References
Multi-language	[96, 97, 102, 104, 110]
20Newsgroups	[93, 98, 104–106]
Sentiment Analysis	[10, 93, 98, 104, 105]
Office-Caltech	[94, 100, 108, 110]
Text-image	[95–97]

Table 5 Summary of the commonly used datasets in OFL studies

Dataset for OFL	References
MNIST and its extensions	[11, 119, 122]
CIFAR	[11, 117]
Air quality	[122, 127]

dataset[3] for breast cancer diagnostic, *german* dataset for credit risk detection, *spambase* dataset for spam email filtering, *a9a* dataset for adult income prediction, and *w8a* dataset for text categorization [141]. Similarly, to explore OTL in different tasks, [96] utilized the *video* dataset from YouTube along with two popular used datasets in Table 4: *multi-language* and *text-image* datasets. Moreover, there are studies that use medical imaging datasets for disease diagnosis. For example, [109] used electrocardiogram (ECG) data for cardiac arrhythmia detection, and [107] used electrocorticogram (ECoG) data for epileptic seizure detection.

Nine papers have adopted special datasets in the obtained OFL studies. Compared to OTL, datasets used in OFL are more personalized, which implies that more publicly recognized datasets are required as benchmarks in this field. Li et al. [92] used *human activity recognition* dataset [142], *eating recognition via Google glass* dataset [143], and *eating habits monitoring* datasets [144] for various real-world tasks. In [125], an image dataset comparable to the average image size of the *CIFAR-10* dataset was used. Apart from using the listed *MNIST* and *air quality* datasets, [122] also adopted the *FitRec* dataset[4], *ExtraSensory* dataset[5] for human activity analysis. Similarly, [124] used a subset of *WISDM Smartphone activity and biometrics* dataset [145] for human activity recognition. To explore OFL in different application scenarios, apart from adopting the *air quality* dataset, [127] also used data from Twitter, *conductivity* dataset [146] in the OTL regression tasks, and used *power consumption*[6] dataset, *parking occupancy*[7] dataset, and *traffic* dataset in the OTL time-series forecasting tasks. Jin et al. [128] used data from US-centric population [73] as the FL workload trace. [131] utilized *YFCC100M*[8] dataset, which is the largest released public multimedia dataset. Instead of using the real-world datasets, two OFL papers [123, 130] chose to use synthetic datasets for experimental analysis.

5.1.1 Popular datasets for OTL

Five datasets have been commonly used in OTL: *Multi-language* dataset, *20Newsgroups* dataset, *sentiment analysis* dataset, *text-image* dataset, and *Office-Caltech* dataset.

The *multi-language* dataset [147] contains feature characteristics of documents written in five different languages (English, French, German, Spanish, and Italian) but sharing the same set of categories. Each language contains indexes of the documents written or translated in that language.

The *20Newsgroups* dataset[9] contains about 20,000 newsgroup documents organized by subject and subcategory. The *20Newsgroups* dataset has been mainly used to implement MS-BC OTL tasks [104–106]. Typically, researchers focus on two primary subjects, each of which has multiple subtopics. Then, to simulate multiple learning domains, a positive label is assigned to each subtopic, which corresponds to the negative label assigned to a subtopic within the other primary subject.

Another commonly used dataset is the *sentiment analysis* dataset[10], which consists of product reviews on Amazon for four different product categories (books, DVDs, electronics, and kitchen). Each review includes a human rating score (0-5 stars), a review caption, position, timestamp, an item description, a reviewer name, and the review content. This dataset has been used to perform SS-BC OTL [10, 93, 98] and MS-BC [104, 105] OTL tasks.

The *Office-Caltech* dataset [148] is made up of real-world object domains gathered from the Berkeley Office [149] and Caltech-256[11], which has been widely utilized in OTL tasks requiring multi-class classification. The Caltech-256 contains 30,607 pictures from 256 groups, and real-world object domains include Amazon, Webcam, and the digital single-lens reflex camera.

Different from the above datasets, the *text-image* dataset has been utilized in a wide range of cross-modality OTL scenarios, and it is sourced from the NUS-WIDE [150]

[3] http://archive.ics.uci.edu/ml/

[4] https://sites.google.com/eng.ucsd.edu/fitrec-project/home

[5] https://sites.google.com/eng.ucsd.edu/fitrec-project/home

[6] https://archive.ics.uci.edu/ml/datasets/

[7] https://data.birmingham.gov.uk/dataset/birmingham-parking

[8] http://bit.ly/yfcc100md

[9] http://qwone.com/~jason/20Newsgroups/

[10] http://www.cse.ust.hk/TL/index.html

[11] https://resolver.caltech.edu/CaltechAUTHORS:CNS-TR-2007-001

collection on Flickr. This dataset comprises photos and tags that have been published on the internet and is often used in heterogeneous SS-BC OTL. More precisely, the unlabelled text-image data pairs in this dataset are often utilized as co-occurrence data to bridge the text samples from the source domain and the images from the target domain.

5.1.2 Popular datasets for OFL

There are several datasets commonly used in OFL: *CIFAR-10 and CIFAR-100*, *MINIST*, and *air quality* dataset. *MINIST* and *CIFAR* are two public datasets that are often utilized in OFL tasks [11, 122], particularly in simulations of non-IID settings.

Both *CIFAR-10 and CIFAR-100* datasets[12] have 60,000 images, with the former having 10 classes with 6,000 photos each and the latter having 100 classes with 600 images each.

MINIST[13] is a database of handwritten digits that contains a training set of 60,000 instances and a testing set of 10,000 instances. This dataset is suitable for pattern recognition tasks as it requires minimal pre-processing.

Air quality datasets collected from weather sensors in different countries were used in [122] and [127] to predict the level of pollutants in the air.

5.1.3 Practical considerations

OTL tasks are generally conducted on public datasets such as *Office-Caltech*, which may have storage format restrictions and are liable to become obsolete. It is also challenging to update these existing datasets or to re-collect fresh datasets. On the other hand, real-world datasets are difficult to obtain due to privacy regulations. Furthermore, most datasets only include a limited number of labelled instances in the target domains, making it challenging to perform cross-validation to fine-tune the target model [100]. For example, OTL applications for the healthcare system are commonly based on publicly available hospital data, and these applications may be limited to patients in a particular geographical area, as people within various geographical regions may have varying physical conditions. Additionally, an OTL system may require target patients to upload their physical states in near real-time, which is highly unlikely in practice due to privacy concerns and system/ infrastructure limitations.

When designing a comparative experiment, different domain types of OTL tasks require different data settings and must comply with the same data dividing rule. On the other hand, data settings for OFL are relatively complex, which requires the stimulation of both the statistical heterogeneity generated by non-IID or imbalanced data and the system heterogeneity caused by the varying uploading rates across numerous local devices in an online scenario. To deal with statistical heterogeneity, researchers often use the standard data decentralization method [73] to classify the data and partition individual categories into multiple shards of varying sizes, after which each local client is allocated with different shards [11, 122]. To stimulate stragglers, a random delay timer may be used to reflect various network delays across local clients [122]. Furthermore, a data growth rate should be predetermined to imitate the growth of online data. Data settings for OFL involve a variety of parameters, and it is important to establish a unified standard for these parameters to facilitate comparative experiments.

5.2 Applications

It is important to note that, despite relatively few studies focusing purely on application-based scenarios, several major prospects for OTL and OFL can be drawn from the obtained studies and their datasets summarized in the previous section, which may in turn lead to future investigations and applications to real-world scenarios. This section will describe the identified cutting-edge applications and discuss relevant practical considerations. It covers the application scenarios and their compatibility of both OTL and OFL in different contexts, with the recognition that existing studies can be categorized into two sectors, namely industrial engineering and healthcare, based on an exhaustive summary from the obtained papers.

5.2.1 Applications in industrial engineering

Given the achievements of OTL in domain shift scenarios and OFL in data privacy protection, it is reasonable to apply these methods to industrial engineering tasks, and Table 6 summarizes the detailed sub-scenarios of OTL and OFL applications in industrial engineering.

OFL has been used in a variety of data-sensitive industrial domains, including environmental protection [122, 127], and unmanned aerial vehicle (UAV) control [119, 121]. OTL applications, on the other hand, are most commonly found in industrial situations that involve domain shift problems, such as sentiment analysis [10, 93, 98, 104, 105]. There are other situations in the industrial engineering where data is likely to be sensitive and therefore a cross-domain task is required, such as image recognition [11, 94, 100, 108, 110, 122, 126] and online recommendation systems [101, 131].

By combining data from multiple weather sensors located at nine separate locations, [122] develop a novel collaborative OFL model for predicting the pollutants in the air. Apart from the environmental protection, OFL has been

[12] https://www.cs.toronto.edu/~kriz/cifar.html

[13] http://yann.lecun.com/exdb/mnist/

Table 6 Summary of sub-scenarios of OTL and OFL in industrial engineering

Sub-Scenarios	OTL	OFL
Environmental Protection	\	[122, 127]
Unmanned Aerial Vehicle	\	[119, 121]
Sentiment Analysis	[10, 93, 98, 104, 105]	\
Image Recognition	[94, 100, 108, 110]	[11, 122, 126]
Online Recommendation Systems	[101]	[131]

used to control UAVs in real-time [119, 121] for mission-critical applications such as first-aid packet dispatching and firefighting [151, 152].

Sentiment analysis has arisen as a hot topic in OTL, with applications ranging from spam detection [109] to document categorization [93, 105, 106]. For example, [109] developed a spam email filter system by analyzing real-world emails from fifteen different users. Such a system can help reduce labor costs while safeguarding the property of people.

Image recognition is a trending topic in OTL. Transferring information from related domains makes it possible to conduct online image classification on the target domain. In addition, computer vision-based tasks [126] are a hot topic in industrial applications of OFL. Rather than uploading annotated personal visual data to a central database, participants in an object recognition task can train a local model on their personal site. Furthermore, by leveraging an online learning framework, OFL enables computer vision-based tasks to manage massive amounts of online image data that arrive sequentially from cameras.

Additionally, OFL and OTL have been implemented in online recommendation systems. [101] proposed Social-Transfer, a cross-domain OTL system for multimedia applications that learns from time-varying social stream data. To address the privacy concerns associated with information sharing, [131] developed a privacy-preserving recommendation system based on OFL that takes advantage of the privacy guarantees provided by the federated learning architecture while still capable of managing the streaming data.

5.2.2 Applications in healthcare

OFL and OTL are both promising solutions for healthcare. For OFL, the connection between real-time data monitoring from various edge devices and hospital records breaks down analysis barriers between various parties while maintaining data privacy. Furthermore, the information required to detect a disease differs from patient to patient. Given that the medical records of each patient constitute a unique domain, OTL is well-suited for disease diagnosis, as it can leverage multiple patient records to improve the diagnosis accuracy of the target patients. OTL has been used to diagnose a wide variety of diseases, including arrhythmias [99], breast cancer [99], and epileptic seizures [107].

Nowadays, with the rapid development in the storage capacity and computing power of edge devices such as smartphones and wearable devices (e.g. google glass), physical data about daily human life can be collected and analysed conveniently. These data, however, are sensitive and are at risk of being compromised by unauthorized access. On the other hand, real-time monitoring systems are required for special scenarios, such as remote health condition monitoring for the elderly living alone, as certain acute-onset diseases (e.g. heart attack, stroke) must be detected instantly. With privacy guarantees, OFL is an excellent candidate for the aforementioned application scenarios, and it has been used in a variety of healthcare applications, including human activity recognition [92, 122], and eating habits monitoring [92].

5.2.3 Practical considerations

Existing research on OTL has primarily concentrated on text/image-based applications, which may not be applicable to certain scenarios involving users who are unfamiliar with text/image input. There are studies on TL that have recommended the use of more forms of input, such as voices [153] and gestures [154]. Future OTL research should consider extending these advanced applications to online contexts, which would accommodate a variety of inputs and facilitate human-machine interaction.

While current application domains of OTL and OFL are primarily focused on industrial engineering and healthcare, there are many other areas worth exploring in TL and FL, such as smart transportation [50]. Traditional offline frameworks for smart transportation may benefit from an online environment; for example, establishing an online autonomous driving system may capture the dynamic nature of the vehicle system and the inherent uncertainty of the real-life environment, allowing drivers to make more accurate and timely decisions.

With the widespread use of edge devices, device owners can easily annotate their data by simply tagging or labeling the device, which has been frequently utilized in OFL research. On the other hand, malicious and false tagging

will become more prevalent as local users are able to tag on their own devices. As a result, OFL must concentrate on filtering out invalid tagging to ensure the accuracy of model inferences. Moreover, fewer OTL applications are utilizing smart edge devices due to privacy regulations regarding personal data. We anticipate that OTL models trained on real-time data generated by edge devices will perform significantly better. Therefore, it is anticipated that there will be future research opportunities to combine OTL and OFL to develop an online FTL framework that takes advantage of both OTL and OFL paradigms to accomplish this vision. After having investigated OTL and OFL from practical perspectives, we conclude this survey with a discussion of several areas of future work worthy of consideration. In particular, we present a vision for online FTL and describe the proposed framework in detail.

6 Discussion and conclusion

In this survey, we have provided a systematic and comprehensive overview of OTL and OFL. OTL employs knowledge from single or multiple source domains to train online target models for the target domain, while the OFL facilitates the traininig of online models at the edge of distributed networks. We discussed the unique properties of OTL from a domain-task perspective and described existing research on OFL addressing several major challenges. Moreover, popular datasets and cutting-edge online federated and transfer learning applications were summarized, and practical considerations were presented from both datasets and applications perspectives. In the following, we will identify open problems worthy of future research efforts, and also propose a vision of online federated transfer learning - a new concept we have developed with the aim of addressing the most significant challenges faced by existing studies.

rom the methodology perspective, existing OTL studies have mainly focused on SS-BC and MS-BC OTL, while studies for multi-class classification OTL tasks have been relatively scarce. Therefore, sophisticated OTL frameworks for various types of learning tasks should to be developed in future research. Moreover, most of the current OTL frameworks rely on the kernel method to build their online target classifiers, which has the distinct benefit of being more accurate than linear models. On the other hand, the kernel method also has an acknowledged disadvantage of being resource-intensive in terms of support vector storage. It is recommended that efficient solutions such as budget online kernel learning [54], which restricts the number of support vectors to a fixed budget, are included in the future OTL framework for their potential to reduce computing overhead significantly. On the other hand, studies in the field of OFL have frequently focused on developing effective models for a variety of asynchronous devices. Moreover, all current OFL frameworks, whether synchronous or asynchronous, assume that local devices are available during their allocated 'working period', which is impractical since unforeseen events may occasionally occur, rendering these local devices being unavailable. As a result, a feedback mechanism could be developed in the future OFL framework to confer sufficient authority on the local device to commence the communication process.

From the practical perspective, existing OTL studies often utilize public datasets, and the real-world datasets are difficult to obtain due to data privacy regulations, as OTL is based on the assumption that all models will be trained on a centralized platform. Therefore, there is a need for the collection of more state-of-the-art datasets for OTL tasks. On the other hand, OFL datasets are more diverse, since the local clients can retain the dataset on their own device. However, typical OFL tasks often require complex data settings for simulating the heterogeneous scenarios in the real world, and different settings of the datasets result in difficulties when comparing different OFL frameworks. Therefore, developing unified data setting protocols is also necessary for future research. Moreover, the most prevalent learning type in real-world applications is supervised learning for OTL and OFL, which involves label-revealing after each prediction. Although significant progress has been made in online federated and transfer learning for handling distributed time-varying data with few labels, applications for unsupervised learning remain a challenge in this field. Methods such as [155], which used a selective pseudo-labeling strategy to achieve high performance for unsupervised TL, and federated unsupervised representation learning [156], which pre-trained deep neural networks using unlabelled data in a federated setting, have shown promising outcomes recently. FL and TL, as two forms of collaborative training, hold tremendous potential in the filed of unsupervised learning. Given the dynamic requirements of real-world machine learning, it is reasonable to suggest that future research on FL and TL extensions for unsupervised learning in online contexts is necessary.

The implementation scenarios of TL, FL, FTL, OTL, and OFL are summarized in Table 7, and the ideal implementation scenarios of online FTL are also given in the table. As can be seen from Table 7, OTL enables standard TL to handle real-time data efficiently. As with the standard TL, OTL is rarely studied in decentralized environments, and carries the risk of data privacy violations due to the instance transmission process. On the other hand, OFL is able to handle local data generated in real-time as well as provide privacy guarantees. However, similar to standard FL, OFL needs to utilize special techniques, such as TL, to create personalized local models. Since FTL

Table 7 Frontier implementation scenarios of different techniques

	Decentralization	Heterogeneity			Inadequate Well-labelled Data	Privacy-Preserving	Client-Side Personalization	Real-time Data
		Cross-Modality	Cross-Model	Cross-System				
Transfer Learning	X	✓	✓	X	✓	X	✓	X
Federated Learning	✓	✓	✓	✓	✓	✓	X	X
Federated Transfer Learning	✓	✓	✓	✓	✓	✓	✓	X
Online Transfer Learning	X	✓	✓	X	✓	X	✓	✓
Online Federated Learning	✓	✓	✓	✓	✓	✓	X	✓
Online Federated Transfer Learning	✓	✓	✓	✓	✓	✓	✓	✓

has gained increasing attention as numerous studies have demonstrated its efficiency [44], we envisage that extending FTL to online scenarios will enable the development of an advanced machine learning framework with dynamic natures that leverages both OTL and OFL paradigms for benefit to the general user.

Figure 15 illustrates the proposed online FTL framework that is described below. The data in the source domain

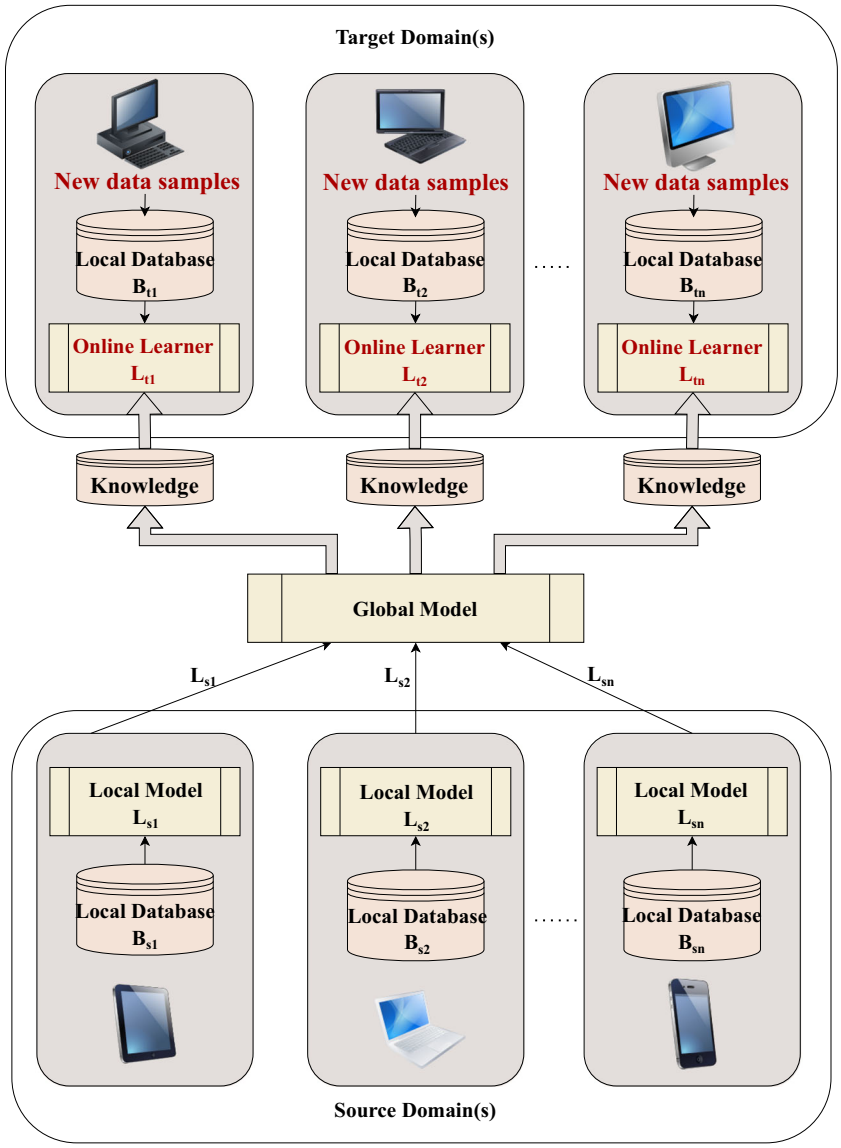


Fig. 15 A vision of online FTL framework

can be generated in real-time or from pre-given datasets. It should be noted that a scratch of the source data is essential to ensure the benchmark performance of the source models. Each local device in the target domain generates data in an online fashion, and the real-time data is analysed by online learners, who then attempt to formulate an optimal strategy for online updating during each training round [54]. The global model enables model aggregation, heterogeneous computing, updating, and broadcasting. Local devices, such as smartphones and laptops, provide essential infrastructure tools, including local online/offline training, uploading, and distributed storage.

Various applications may be developed on top of the proposed online FTL to provide critical human-machine interface services. By utilizing federated learning, machine learning models for multiple parties can be established without exporting local data, ensuring data security and privacy while providing users with tailored services. Meanwhile, TL enables FL to train models on a variety of different but related parties, which is practically important given that stakeholders within the same FL framework are usually from the same sector. Furthermore, classical batch/ offline learning has low efficiency in terms of computing costs, as well as limited scalability for large-scale applications due to the need for model retraining after online data sequences are generated. We envisage that extending FTL to online scenarios will help overcome the limitations of traditional batch learning by allowing online learners to update the local model safely and rapidly.

To summarize, this survey aims to serve as a resource for researchers and practitioners developing online federated and transfer learning frameworks. It provides a systematic and comprehensive description of OTL and OFL, and identifies open research questions worthy of future research efforts. Finding solutions to such new and arising research problems from methodologies to practical applications will necessitate collaborative and long-term efforts from various research communities.

Acknowledgements We would like to thank the Editor and two anonymous reviewers for their helpful and insightful comments. We are grateful to Dr. Dan Brawn at the University of Essex for proofreading our manuscript to improve the presentation.

References

1. Holst A (2021) Number of internet of things (iot) connected devices worldwide from 2019 to 2030. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/. Accessed 11 Dec 2021
2. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) IEEE conference on computer vision and pattern recognition (Ieee 2009), pp 248–255
3. Holcomb SD, Porter WK, Ault SV, Mao G, Wang J (2018) Proceedings of the 2018 international conference on big data and education, pp 67–71
4. Tiwari SR, Rana KK (2021) Feature selection in big data: trends and challenges. Data Sci Intell Appl:83–98
5. E.S. of Radiology (ESR) eu-affairs@ myesr. org (2017) The new eu general data protection regulation: what the radiologist should know. Insights Into Imaging 8:295–299
6. Li X, Huang K, Yang W, Wang S, Zhang Z (2019) International conference on learning representations
7. Mackenzie J, Roddick JF, Zito R (2019) An evaluation of htm and lstm for short-term arterial traffic flow prediction. IEEE Trans Intell Transp Syst 20(5):1847–1857. https://doi.org/10.1109/TITS.2018.2843349
8. Kulkarni V, Kulkarni M, Pant A (2020) 2020 Fourth world conference on smart trends in systems. In: Security and sustainability (worldS4) (IEEE, 2020), pp 794–797
9. Zhao P, Hoi S (2010) Proceedings of the 27th international conference on machine learning (ICML-10), pp 1231–1238
10. Zhao P, Hoi SC, Wang J, Li B (2014) Online transfer learning. Artif Intell 216:76–102
11. Damaskinos G, Guerraoui R, Kermarrec AM, Nitu V, Patra R, Taïani F (2020) Proceedings of the 21st international middleware conference, pp 163–177
12. Pan SJ, Yang Q (2009) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
13. Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: concept and applications. ACM Trans Intell Syst Technol (TIST) 10(2):1–19
14. Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. J Big Data 3(1):1–40
15. Xu F, Yu J, Xia R (2018) Instance-based domain adaptation via multiclustering logistic approximation. IEEE Intell Syst 33(1):78–88
16. Long M, Cao Y, Cao Z, Wang J, Jordan MI (2018) Transferable representation learning with deep adaptation networks. IEEE Trans Pattern Anal Mach Intell 41(12):3071–3085
17. Rozantsev A, Salzmann M, Fua P (2018) Beyond sharing weights for deep domain adaptation. IEEE Trans Pattern Anal Mach Intell 41(4):801–814
18. Yan K, Guo X, Ji Z, Zhou X (2021) Deep transfer learning for cross-species plant disease diagnosis adapting mixed subdomains. IEEE/ACM Trans Comput Bio Bioinform
19. Tang Y, Wei Y, Yu X, Lu J, Zhou J (2020) Graph interaction networks for relation transfer in human activity videos. IEEE Trans Circuits Syst Video Technol 30(9):2872–2886
20. Sun G, Liang L, Chen T, Xiao F, Lang F (2018) Network traffic classification based on transfer learning. Comput Electr Eng 69:920–927
21. Yang Y, Li X, Wang P, Xia Y, Ye Q (2020) Multi-source transfer learning via ensemble approach for initial diagnosis of alzheimer's disease. IEEE J Trans Eng Health Med 8:1–10
22. Gao P, Wu W, Li J (2021) Multi-source fast transfer learning algorithm based on support vector machine. Appl Intell 51(11):8451–8465
23. He X, Chen Y, Ghamisi P (2019) Heterogeneous transfer learning for hyperspectral image classification based on convolutional

neural network. IEEE Trans Geosci Remote Sens 58(5):3246–3263
24. Zhou JT, Pan SJ, Tsang IW (2019) A deep learning framework for hybrid heterogeneous transfer learning. Artif Intell 275:310–328
25. Dai W, Yang Q, Rong Xue G, Yu Y (2007) In ICML
26. Li Z, Liu B, Xiao Y (2017) 2017 13th International conference on natural computation. Fuzzy Syst Knowl Discovery (ICNC-FSKD) (IEEE), pp 2291–2295
27. Ye Y, Lin Q, Ma L, Wong KC, Gong M, Coello CAC (2022) Multiple source transfer learning for dynamic multiobjective optimization. Inf Sci
28. Eaton E, DesJardins M (2011) Proceedings of the twenty-fifth AAAI conference on artificial intelligence, pp 337–342
29. Niu S, Liu Y, Wang J, Song H (2020) A decade survey of transfer learning (2010–2020). IEEE Trans Artif Intell 1(2):151–166
30. Xia R, Hu X, Lu J, Yang J, Zong C (2013) Proceedings of the twenty-third international joint conference on Artificial Intelligence, pp 2176–2182
31. Wang Z, Song Y, Zhang C (2008) Joint European conference on machine learning and knowledge discovery in databases (Springer), pp 550–565
32. McMahan B, Moore E, Ramage D, Hampson S, Arcas BAY (2017) Artificial intelligence and statistics (PMLR), pp 1273–1282
33. Abad MSH, Ozfatura E, GUndUz D, Ercetin O (2020) ICASSP 2020 - 2020 IEEE international conference on acoustics. Speech Signal Process (ICASSP):8866–8870. https://doi.org/10.1109/ICASSP40776.2020.9054634
34. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K (2017) Proceedings of the 2017 ACM SIGSAC conference on computer and communications security
35. Sattler F, Wiedemann S, Müller KR, Samek W (2019) Robust and communication-efficient federated learning from non-iid data. IEEE Trans Neural Netw Learn Syst 31(9):3400–3413
36. Wang H, Kaplan Z, Niu D, Li B (2020) IEEE INFOCOM 2020-IEEE conference on computer communications. IEEE, pp 1698–1707
37. Yang Q, Liu Y, Cheng Y, Kang Y, Chen T, Yu H (2020) Federated learning. Springer, pp 69–81
38. Yang S, Ren B, Zhou X, Liu L (2019) Parallel distributed logistic regression for vertical federated learning without third-party coordinator. arXiv:1911.09824
39. Lian X, Zhang C, Zhang H, Hsieh CJ, Zhang W, Liu J (2017) Proceedings of the 31st international conference on neural information processing systems, pp 5336–5346
40. Wang H, Muñoz-gonzález L, Eklund D, Raza S (2021) Proceedings of the 14th ACM conference on security and privacy in wireless and mobile networks, pp 153–163
41. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, Bonawitz K, Charles Z, Cormode G, Cummings R et al (2021) Advances and open problems in federated learning. Foundations and Trends® Mach Learn 14(1–2):1–210
42. Chen M, Yang Z, Saad W, Yin C, Poor HV, Cui S (2019) IEEE global communications conference (GLOBECOM) (IEEE 2019):1–6
43. Sarikaya Y, Ercetin O (2019) Motivating workers in federated learning: a stackelberg game perspective. IEEE Netw Lett 2(1):23–27
44. Liu Y, Kang Y, Xing C, Chen T, Yang Q (2020) A secure federated transfer learning framework. IEEE Intell Syst 35(4):70–82
45. Mothukuri V, Parizi RM, Pouriyeh S, Huang Y, Dehghantanha A, Srivastava G (2021) A survey on security and privacy of federated learning. Futur Gener Comput Syst 115:619–640
46. Gao D, Liu Y, Huang A, Ju C, Yu H, Yang Q (2019) IEEE international conference on big data (big data) (IEEE) 2019, pp 2552–2559
47. Kevin I, Wang K, Zhou X, Liang W, Yan Z, She J (2021) Federated transfer learning based cross-domain prediction for smart manufacturing. IEEE Trans Industr Inform
48. Sharma S, Xing C, Liu Y, Kang Y (2019) IEEE international conference on big data (big data) (IEEE 2019), pp 2569–2576
49. Chen Y, Qin X, Wang J, Yu C, Gao W (2020) Fedhealth: a federated transfer learning framework for wearable healthcare. IEEE Intell Syst 35(4):83–93
50. Majeed U, Hassan SS, Hong CS (2021) International conference on information networking (ICOIN) (IEEE 2021), pp 588–593
51. Zhang Y, Tang G, Huang Q, Wang Y, Wu K, Yu K, Shao X (2022) Fednilm: applying federated learning to nilm applications at the edge. IEEE Trans Green Commun Netw
52. Yang H, He H, Zhang W, Cao X (2020) Fedsteg: a federated transfer learning framework for secure image steganalysis. IEEE Trans Netw Sci Eng
53. Han T, Liu C, Yang W, Jiang D (2019) Learning transferable features in deep convolutional neural networks for diagnosing unseen machine conditions. ISA Trans 93:341–353
54. Hoi SC, Sahoo D, Lu J, Zhao P (2021) Online learning: a comprehensive survey. Neurocomputing 459:249–289
55. Cesa-Bianchi N, Lugosi G (2006) Prediction, learning, and games (Cambridge University Press)
56. Hoi SC, Wang J, Zhao P (2014) Libol: a library for online learning algorithms. J Mach Learn Res 15(1):495
57. Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Rev 65(6):386
58. Novikoff AB (1963) On convergence proofs for perceptrons. Tech rep, STANFORD RESEARCH INST MENLO PARK CA
59. Crammer K, Dekel O, Keshet J, Shalev-Shwartz S, Singer Y (2006) Online passive-aggressive algorithms. J Mach Learn Res 7:551–585
60. Zinkevich M (2003) Proceedings of the 20th international conference on machine learning (icml-03), pp 928–936
61. Bartlett P, Hazan E, Rakhlin A (2009) Advances in neural information processing systems 20: proceedings of the 2007 conference (neural information processing systems (NIPS) foundation), pp 65–72
62. Dekel O, Gilad-Bachrach R, Shamir O, Xiao L (2012) Optimal distributed online prediction using mini-batches. J Mach Learn Res vol 13(1)
63. Dekel O, Shalev-Shwartz S, Singer Y (2008) The forgetron: a kernel-based perceptron on a budget. SIAM J Comput 37(5):1342–1372
64. Zhao P, Wang J, Wu P, Jin R, Hoi SC (2012) Proceedings of the 29th international coference on international conference on machine learning, pp 1075–1082
65. Cesa-Bianchi N, Conconi A, Gentile C (2005) A second-order perceptron algorithm. SIAM J Comput 34(3):640–668
66. Dredze M, Crammer K, Pereira F (2008) Proceedings of the 25th international conference on machine learning, pp 264–271
67. Luo H, Agarwal A, Cesa-Bianchi N, Langford J (2016) Efficient second order online learning by sketching. Adv Neural Inf Process Syst 29:902–910
68. Lu CH, Lin XZ (2020) Toward direct edge-to-edge transfer learning for iot-enabled edge cameras. IEEE Int Things J 8(6):4931–4943
69. McMahan HB, Ramage D, Talwar K, Zhang L (2018) International conference on learning representations
70. Truex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R, Zhou Y (2019) Proceedings of the 12th ACM workshop on artificial intelligence and security, pp 1–11

71. Li T, Sahu AK, Talwalkar A, Smith V (2020) Federated learning: challenges, methods, and future directions. IEEE Signal Proc Mag 37(3):50–60
72. Liu K, Zhang H, Ng JKY, Xia Y, Feng L, Lee VC, Son SH (2017) Toward low-overhead fingerprint-based indoor localization via transfer learning: design, implementation, and evaluation. IEEE Trans Industr Inform 14(3):898–908
73. Bonawitz K, Eichner H, Grieskamp W, Huba D, Ingerman A, Ivanov V, Kiddon C, Konečnỳ J, Mazzocchi S, McMahan B et al (2019) Towards federated learning at scale: system design. Proc Mach Learn Syst 1:374–388
74. Yang T, Andrew G, Eichner H, Sun H, Li W, Kong N, Ramage D, Beaufays F (2018) Applied federated learning: improving google keyboard query suggestions. arXiv:1812.02903
75. Singh AP, Gordon GJ (2008) Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 650–658
76. Zhao L, Chen Z, Yang LT, Deen MJ, Wang ZJ (2019) Deep semantic mapping for heterogeneous multimedia transfer learning using co-occurrence data. ACM Trans Multimed Comput Commun Appl (TOMM) 15(1s):1–21
77. Yang L, Jing L, Yu J, Ng MK (2015) Learning transferred weights from co-occurrence data for heterogeneous transfer learning. IEEE Trans Neural Netw Learn Syst 27(11):2187–2200
78. Li D, Wang J (2019) Fedmd: heterogenous federated learning via model distillation. arXiv:1910.03581
79. Dinh CT, Tran N, Nguyen TD (2020) Personalized federated learning with moreau envelopes. Adv Neural Inf Process Syst, vol 33
80. Lin T, Kong L, Stich SU, Jaggi M (2020) NeurIPS
81. Xie C, Koyejo O, Gupta I (2019) Asynchronous federated optimization. J Environ Sci (China) English Ed
82. Gui L, Xu R, Lu Q, Du J, Zhou Y (2018) Negative transfer detection in transductive transfer learning. Int J Mach Learn Cybern 9(2):185–197
83. Tan AZ, Yu H, Cui L, Yang Q (2022) Towards personalized federated learning. IEEE Trans Neural Netw Learn Syst
84. Karimireddy SP, Kale S, Mohri M, Reddi S, Stich S, Suresh AT (2020) International conference on machine learning (PMLR), pp 5132–5143
85. Duong-Trung N, Quach LD, Nguyen CN (2019) Learning deep transferability for several agricultural classification problems. Int J Adv Comput Sci Appl, vol 10(1)
86. Zeng T, Guo J, Kim KJ, Parsons K, Orlik P, Di Cairano S, Saad W (2021) IEEE intelligent vehicles symposium (IV) (IEEE 2021), pp 451–457
87. Smith V, Chiang CK, Sanjabi M, Talwalkar A (2017) Proceedings of the 31st international conference on neural information processing systems, pp 4427–4437
88. Zhao Y, Chen J, Wu D, Teng J, Yu S (2019) Proceedings of the tenth international symposium on information and communication technology, pp 273–279
89. Zheng T, Chen G, Wang X, Chen C, Wang X, Luo S (2019) Real-time intelligent big data processing: technology, platform, and applications. Sci China Inf Sci 62(8):1–12
90. Xiao J, Wang M, Jiang B, Li J (2018) A personalized recommendation system with combinational algorithm for online learning. J Ambient Intell Human Comput 9(3):667–677
91. Alcalá JM, Ureñ a J, Hernández ÁD, Gualda D (2017) Assessing human activity in elderly people using non-intrusive load monitoring. Sensors 17(2):351
92. Li R, Ma F, Jiang W, Gao J (2019) IEEE international conference on big data (big data) (IEEE 2019), pp 215–220
93. Chen Q, Du YT, Xu M, Wang CJ (2018) IEEE 30th international conference on tools with artificial intelligence (ICTAI) (IEEE 2018), pp 350–357
94. Luo Y, Liu T, Wen Y, Tao D (2018) IJCAI, pp 2525–2531
95. Yan Y, Wu Q, Tan M, Min H (2016) European conference on computer vision (Springer), pp 467–474
96. Yan Y, Wu Q, Tan M, Ng MK, Min H, Tsang IW (2017) Online heterogeneous transfer by hedge ensemble of offline and online decisions. IEEE Trans Neural Netw Learn Syst 29(7):3252–3263
97. Wu H, Yan Y, Ye Y, Min H, Ng MK, Wu Q (2019) Online heterogeneous transfer learning by knowledge transition. ACM Trans Intell Syst Technol (TIST) 10(3):1–19
98. Du YT, Qian C, Lu HY, Wang CJ (2018) IEEE 30th international conference on tools with artificial intelligence (ICTAI) (IEEE 2018), PP 344–349
99. Zhang Y, Wu M, Hu X, Zhu Y (2018) Proceedings of the 2018 international conference on signal processing and machine learning, pp 135–141
100. Zhang X, Zhuang Y, Wang W, Pedrycz W (2017) Online feature transformation learning for cross-domain object category recognition. IEEE Trans Neural Netw Learn Syst 29(7):2857–2871
101. Roy SD, Mei T, Zeng W, Li S (2012) Proceedings of the 20th ACM international conference on multimedia, pp 649–658
102. Kang Z, Yang B, Li Z, Wang P (2019) Otlamc: an online transfer learning algorithm for multi-class classification. Knowl-Based Syst 176:133–146
103. Wen Y, Qin Y, Qin K, Lu X, Liu P (2019) Online transfer learning with multiple decision trees. Int J Mach Learn Cybern 10(10):2941–2962
104. Wu Q, Wu H, Zhou X, Tan M, Xu Y, Yan Y, Hao T (2017) Online transfer learning with multiple homogeneous or heterogeneous sources. IEEE Trans Knowl Data Eng 29(7):1494–1507
105. Wu Q, Zhou X, Yan Y, Wu H, Min H (2017) Online transfer learning by leveraging multiple source domains. Knowl Inf Syst 52(3):687–707
106. Wang X, Wang X, Zeng Z (2020) 2020 12th International conference on advanced computational intelligence (ICACI) (IEEE), pp 349–355
107. Wang B, Pineau J (2015) Proceedings of the twenty-ninth AAAI conference on artificial intelligence, pp 3038–3044
108. Du Y, Tan Z, Chen Q, Zhang Y, Wang C (2020) ECAI 2020 (IOS Press), pp 1111–1118
109. Ge L, Gao J, Zhang A (2013) Proceedings of the 22nd ACM international conference on information & knowledge management, pp 2423–2428
110. Kang Z, Yang B, Yang S, Fang X, Zhao C (2020) Online transfer learning with multiple source domains for multi-class classification. Knowl-Based Syst 190(105):149
111. Pan W, Yang Q (2013) Transfer learning in heterogeneous collaborative filtering domains. Artif Intell 197:39–55
112. Young P, Lai A, Hodosh M, Hockenmaier J (2014) From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions. Trans Association Comput Linguistics 2:67–78
113. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Comput Syst Sci 55(1):119–139
114. Liu Y, James J, Kang J, Niyato D, Zhang S (2020) Privacy-preserving traffic flow prediction: a federated learning approach. IEEE Int Things J 7(8):7751–7763
115. Mishne G, Dalton J, Li Z, Sharma A, Lin J (2013) Proceedings of the 2013 ACM SIGMOD international conference on management of data, pp 1147–1158
116. Brisimi TS, Chen R, Mela T, Olshevsky A, Paschalidis IC, Shi W (2018) Federated learning of predictive models from federated electronic health records. Int J Med Inform 112:59–67
117. Wang J, Guo Z, Liu S, Xia Y (2020) IEEE 28th international conference on network protocols (ICNP) (IEEE 2020), pp 1–2

118. Bajwa WU, Cevher V, Papailiopoulos D, Scaglione A (2020) Machine learning from distributed, streaming data [from the guest editors]. IEEE Signal Proc Mag 37(3):11–13
119. Wang S, Hosseinalipour S, Gorlatova M, Brinton CG, Chiang M (2021) Uav-assisted online machine learning over multi-tiered networks: a hierarchical nested personalized federated learning approach. arXiv:2106.15734
120. Dwork C, Roth A (2014) The algorithmic foundations of differential privacy. Foundations and trends®, in theoretical computer science 9(3–4):211–407
121. Shiri H, Park J, Bennis M (2020) Communication-efficient massive uav online path control: federated learning meets mean-field game theory. IEEE Trans Commun 68(11):6840–6857
122. Chen Y, Ning Y, Slawski M, Rangwala H (2020) IEEE international conference on big data (big data) (IEEE 2020), pp 15–24
123. Gogineni VC, Werner S, Huang YF, kuh A (2022) ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP) (IEEE), pp 5228–5232
124. Giorgas K, Varlamis I (2020) 24th Pan-hellenic conference on informatics, pp 91–95
125. Zhou Z, Yang S, Pu L, Yu S (2020) Cefl: online admission control, data scheduling, and accuracy tuning for cost-efficient federated learning across edge nodes. IEEE Int Things J 7(10):9341–9356
126. Liu Y, Huang A, Luo Y, Huang H, Liu Y, Chen Y, Feng L, Chen T, Yu H, Yang Q (2020) Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 13,172–13,179
127. Hong S, Chae J (2021) Communication-efficient randomized algorithm for multi-kernel online federated learning. IEEE Trans Pattern Anal Mach Intell
128. Jin Y, Jiao L, Qian Z, Zhang S, Lu S, Wang X (2020) IEEE 40th international conference on distributed computing systems (ICDCS) (IEEE 2020), pp 606–616
129. Zhai S, Jin X, Wei L, Luo H, Cao M (2021) Dynamic federated learning for gmec with time-varying wireless link. IEEE Access 9:10,400–10,412
130. Odeyomi O, Zaruba G (2021) IEEE international symposium on information theory (ISIT) (IEEE 2021), pp 1308–1313
131. Zhou P, Wang K, Guo L, Gong S, Zheng B (2019) A privacy-preserving distributed contextual federated online learning framework with big data support in social recommender systems. IEEE Trans Knowl Data Eng
132. Liu S, Feng X, Zheng H (2022) Pacific-asia conference on knowledge discovery and data mining (Springer), pp 613–625
133. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G et al (2015) Human-level control through deep reinforcement learning. nature 518(7540):529–533
134. Zhang Y, Yeung DY (2010) Proceedings of the twenty-sixth conference on uncertainty in artificial intelligence, pp 733–742
135. Jiang J, Cui B, Zhang C, Yu L (2017) Proceedings of the 2017 ACM international conference on management of data, pp 463–478
136. Zhang W, Gupta S, Lian X, Liu J (2016) Proceedings of the twenty-fifth international joint conference on artificial intelligence, pp 2350–2356
137. Firat O, Cho K, Sankaran B, Vural FTY, Bengio Y (2017) Multi-way, multilingual neural machine translation. Comput Speech Language 45:236–252
138. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Proceedings of the 31st international conference on neural information processing systems, pp 6000–6010
139. Wang YM, Elhag TM (2006) On the normalization of interval and fuzzy weights. Fuzzy Sets Syst 157(18):2456–2471
140. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv:1607.06450
141. Zhao P, Zhuang F, Wu M, Li XL, Hoi SC (2015) IEEE international conference on data mining (IEEE 2015), pp 649–658
142. Anguita D, Ghio A, Oneto L, Parra Perez X, Reyes Ortiz JL (2013) Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning, pp 437–442
143. Rahman SA, Merck C, Huang Y, Kleinberg S (2015) 2015 9th International conference on pervasive computing technologies for healthcare (PervasiveHealth) (IEEE), pp 108–111
144. Bi Y, Xu W, Guan N, Wei Y, Yi W (2014) Proceedings of the 8th international conference on pervasive computing technologies for healthcare, pp 174–177
145. Weiss GM (2019) Wisdm smartphone and smartwatch activity and biometrics dataset. UCI Mach Learn Repo WISDM Smartphone Smartwatch Activity Bio Dataset Data Set 7:133,190–133,202
146. Hamidieh K (2018) A data-driven statistical model for predicting the critical temperature of a superconductor. Comput Mater Sci 154:346–354
147. Amini MR, Usunier N, Goutte C (2009) Learning from multiple partially observed views-an application to multilingual text categorization. Adv Neural Inf Process Syst, vol 22
148. Gong B, Shi Y, Sha F, Grauman K (2012) IEEE conference on computer vision and pattern recognition (IEEE 2012), pp 2066–2073
149. Saenko K, Kulis B, Fritz M, Darrell T (2010) European conference on computer vision (Springer), pp 213–226
150. Chua TS, Tang J, Hong R, Li H, Luo Z, Zheng Y (2009) Proceedings of the ACM international conference on image and video retrieval, pp 1–9
151. Ackerman E, Strickland E (2018) Medical delivery drones take flight in east africa. IEEE Spectr 55(1):34–35
152. Tisdale J, Kim Z, Hedrick JK (2009) Autonomous uav path planning and estimation. IEEE Robot Autom Magazine 16(2):35–42
153. Devlin J, Chang MW, Lee K, Toutanova K (2019) Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies vol 1(Long and Short Papers), pp 4171–4186
154. Côté-Allard U, Fall CL, Drouin A, Campeau-Lecours A, Gosselin C, Glette K, Laviolette F, Gosselin B (2019) Deep learning for electromyographic hand gesture signal classification using transfer learning. IEEE Trans Neural Syst Rehab Eng 27(4):760–771
155. Wang Q, Breckon T (2020) Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 6243–6250
156. van Berlo B, Saeed A, Ozcelebi T (2020) Proceedings of the third ACM international workshop on edge systems, analytics and networking, pp 31–36

Shuang Dai achieved her Master Degree in advanced computer science (Artificial Intelligence) at the Department of Computer Science of University of Manchester, and now is a Ph.D. Student starts at the Department of Mathematical Sciences of University of Essex since October 2019.

Currently, Shuang is developing a real-time federated/distributed privacy-preserving machine learning framework for smart meter big data. Her research interests cover federated learning, distributed machine learning, privacy-preserving data mining, real-time smart meter big data clustering and forecasting.

In addition she is an research assistant at the University of Essex working with several orgnisations including the Institute of Social and Economic Research and Essex Partnership University NHS Foundation Trust and Essex County Council.