Selective Federated Transfer Learning using Representation Similarity

Tushar Semwal[1], **Haofan Wang**[2], **Chinnakotla Krishna Teja Reddy**[3]
{ [1]The University of Edinburgh, [2]Carnegie Mellon University, [3]GE Healthcare, [1,2,3]OpenMined}
{ [1]`tushar.semwal@ed.ac.uk`, [2] `haofanw@andrew.cmu.edu`, [3]`ck13@iitbbs.ac.in` }

Abstract

Transfer Learning (TL) has achieved significant developments in the past few years. However, the majority of work on TL assume implicit access to both the *target* and *source* datasets, which limits its application in the context of Federated Learning (FL), where target (client) datasets are usually not directly accessible. In this paper, we address the problem of source model selection in TL for federated scenarios. We propose a simple framework, called Selective Federated Transfer Learning (SFTL), to select the best pre-trained models which provide a positive performance gain when their parameters are transferred on to a new task. We leverage the concepts from representation similarity to compare the similarity of the client model and the source models and provide a method which could be augmented to existing FL algorithms to improve both the communication cost and the accuracy of client models.

1 Introduction

The conventional deep learning (DL) approaches involve first collecting data (for instance, images or text) from various sources such as smartphones, and storage devices, and then training a Deep Neural Network (DNN) using this centrally aggregated data. Though this centralised form of architecture is relatively practical to implement and provides full control, the act of gathering data in third-party private servers raises serious privacy concerns. Further, with the Internet of Things (IoT) entering a new era of development, International Data Corporation is estimating a total of 41.6 billion connected IoT devices generating 79.4 zettabytes of data in 2025 [1]. Transmitting such huge amount of data from these edge devices to the cloud could lead to bandwidth congestion, data-processing delays, and potential leakage of privacy. Recently, Federated Learning (FL) [2] was introduced as an alternative paradigm, which avoids a purely centralised form of training. In FL, the *clients* (edge devices), train a DNN locally on their data. Thus, instead of sending the raw data, the clients share the parameters or gradient updates of their respective *client model* with a central server. The server then aggregates the parameters received from the participating clients and share the averaged *global model* back to the clients. This cycle repeats for the desired number of communication rounds between clients and the server.

The DNNs are known to be data-hungry. With privacy and confidentiality constraints reducing access to the available data, it limits the performance of FL-based systems employing such parametric models. Since the inception of `FedAvg` [3], the first FL algorithm, researchers have adopted concepts from multi-task learning [4], resource-scheduling [5], asynchronous computing [6], cryptography [7], and reinforcement learning [8], to efficiently train DNNs in such distributed FL scenarios. One of the research areas is *Transfer Learning* (TL), which is widely applied in the conventional DL settings. TL is a technique in machine learning which leverages the knowledge stored in a *source* domain (or model) and transfers into a *target* domain [9]. The domain could be tasks, datasets, and games, to name a few. TL derives its concepts based on the fact that features learned by a model can be reused

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

or repurposed on similar tasks. For example, features learned to classify apples can also be used to recognise pears and peaches - fruits of the same family. TL becomes indispensable in scenarios where there is not enough data to train accurate models, especially in the case of FL, where access to the data is limited.

A transfer from a source model to a target model is said to be *positive* if it achieves an increase in the accuracy of the target task, otherwise the transfer is deemed *negative*. Selecting a suitable source model to achieve a *positive* transfer, is crucial for an efficient TL. Traditional DL has brought a substantial number of publications[1] describing methods on adapting or selecting source domains in TL. However, the majority of these work assume implicit access to both the target and source datasets. Such an assumption is not applicable in the context of FL where the information on target data distribution is either limited or inaccessible.

In this paper, we propose a simple framework to select the best pre-trained source models (feature extractors) which provide a performance gain on a new task, in a federated manner. To select the top-performing feature extractors, we first use the concepts of representation to find the most similar pre-trained source models to a client model and then transfer the knowledge from the selected source model. We propose the use of Centered Kernel Alignment (CKA) [10] as a metric to calculate scalar similarity scores between representation vectors of the source and client model. The available pool of source models are shared with the participating clients, and each client then calculates a similarity score using the CKA metric. We then propose a simple federated voting (`FedVote`) algorithm which elects the source models having the highest similarity scores and share their indexes back to the server. Since both the similarity calculation and election happens at the client-side, the data remains locally and never shared.

2 Related Work

The majority of work on TL stresses the significance of the similarity between a source and target task on the transfer performance. An unwise transfer from a randomly selected source model could lead to a negative transfer which may further deteriorate the accuracy of the target model. The question on how to predict a suitable source model is a relatively under-explored study [11]. Some typical techniques include matching the conditional probability distribution of source and target datasets [12, 13], selecting relevant data instances from source data [14], and model adaptation methods [15]. Alternatively, instead of the data, ranking pre-trained convolutional neural networks using concepts from the mutual information theory has also been proposed [11]. These approaches provide an increase in the transfer performance for traditional DL tasks. However, such methods require access to the target data distribution, which may not be feasible in the FL scenarios.

We found that more common approaches for source domain selection use similarity metrics. Metrics such as $\mathcal{A}$-distance [16], Kullback-Leibler divergence [17], Jensen-Shannon divergence [18], and earth mover's distance [19] have found to be effective in selecting a suitable source domain and thereby enhancing the transfer performance [20]. However, the manner in which these approaches have been used in traditional DL problems may not be applicable in the context of FL. The primary reason being that these methods require a training process to adapt the loss based on a chosen similarity metric. Instead, this paper endeavours a computation-effective approach to improving the transfer performance and hence the accuracy of edge devices in an FL scenario.

To avoid any loose ends, we also provide a review of relevant work in FL. Since *FedAvg* by McMahan et al. [3], researcher have proposed various forms of FL frameworks describing approaches to improve accuracy [4, 21], communication efficiency [22, 23], and security [24, 25]. FL poses system and statistical challenges to train clients collecting data in a *non-IID* manner. Instead of training a single global model, Smith et al. [4] address this statistical issue of unbalanced data by learning separate models for each client node using multi-task learning. Though multi-task learning is also a form of TL, authors did not consider non-convex DNN models. Gao et al. [26] propose a federated transfer learning approach to train heterogeneous data from multiple clients by identifying and exploiting common feature spaces in a privacy-preserving manner. The authors present an improvement in the accuracy of the in-hospital mortality prediction task. The work presented above focus on adapting the accuracy of target client as per the source parties. A recent paper on federated transfer learning [27] looks into secured methods for transferring knowledge from rich labelled data, though the authors do

[1] Section 2 elaborates the related work.

not focus on the selection of the suitable source models. Thus, to the best of our knowledge, it is difficult to find work which focuses on selection in federated transfer learning on edge devices.

3 Selective Federated Transfer Learning

In this section, we first introduce the concepts and notations used in this paper. Since finding a similar source model forms a significant part of our work, we also present the similarity metric leveraged in this paper. We finally propose and discuss an algorithm for a successful Selective Federated Transfer Learning (SFTL). We focus our work on non-convex DNNs objectives; however, the proposed solution is equally applicable to finite-sum objectives.

3.1 Preliminaries

3.1.1 Federated Learning

The aim of a FL algorithm is to learn a model with parameters $\mathbf{w} = \{w^1, w^2, ..., w^l\}$, where w^l denotes the parameters (weights and biases) of l^{th} layer of a DNN. We are interested in those FL scenarios where the data is distributed in a non-IID (independent and identically distributed) fashion. We assume the data is partitioned over K clients where $\mathcal{P}_k$ is the set of indexes of data points at client k such that $n_k = |\mathcal{P}_k|$. Thus, the objective is of the form,

$$\min \sum_{k=1}^{K} \frac{n_k}{n} F_k(\mathbf{w}) \quad \text{where} \quad F_k(\mathbf{w}) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} f_i(\mathbf{w}) \tag{1}$$

For any ML problem, $f_i(\mathbf{w})$ = $\mathcal{L}(x_i, y_i : \mathbf{w})$, is the loss of the prediction on example (x_i, y_i) with model parameters $\mathbf{w}$.

3.1.2 Representation

In this paper, and as introduced in Raghu et al. [28], a network representation is derived from its each neuron's *activation vector*, which is the set of responses over a finite set of input samples chosen from the training or validation set. For a given private target dataset $\mathcal{X} = \{x_1, x_2, ..., x_m\}$, and a neuron i on layer l, we define the activation vector z_i^l as the vector of outputs over $\mathcal{X}$ in the form,

$$z_i^l = [z_i^l(x_1), z_i^l(x_2), z_i^l(x_3), ..., z_i^l(x_m)]$$

It is to be noted that the vector z_i^l is different from the output of a layer. Here z_i^l is the response of a *single* neuron over the full input dataset. This representation can be extended from a single neuron to the collection of neurons in a single layer. Thus, if a neuron's representation is a vector in $\mathbb{R}^m$, then a layer is the subspace of $\mathbb{R}^m$ spanned by its neurons' vectors. To generate the activation vector of different layers, we perform one forward pass of the set of input samples through a given DNN.

3.2 Similarity Metric

Quantifying the similarity between two DNNs is crucial for selecting the best source models. Common approaches based on the Canonical Correlation Analysis (CCA) such as Singular Vector CCA (SVCCA) [28] and Project Weighted CCA (PWCCA) [29] have been used to measure a scalar similarity value between layers of two neural networks. Regularised CCA is also ordinarily used in neuroscience [30]. However, these variants of CCA require a large number of data samples, which is not practical in the case of private ML where the data is scarce and has limited access [10].

In the FL scenario considered in this paper, the data cannot leave the client devices. Further, in contrast to the previous work [31, 32] where a single *target* (client) model is compared with given source models, we have K different local models and a single global model. Thus, this situation creates a need for a similarity metric which has four necessary features – (1) calculates similarity in a one-shot manner, i.e., no need to train or adapt to the source models (2) requires fewer input data samples (3) is not invariant to invertible linear transformations (such as batch normalisation) (4) can compare DNNs having different architectures. The variants of CCA and other similar approaches fail to satisfy conditions (2), (3) and (4).

Kornblith et al. [10] introduced CKA as a similarity index which is only invariant to orthogonal transformations and isotropic scaling (and not linear transformations). The similarity index used in this work is based on CKA. However, the regular CKA metric has an obvious drawback. It cannot be used for comparing activation vectors having different size, i.e., if the layers of the two DNNs being compared have the different number of neurons.

Inspired from work by Shuai et al. [33], we extend the CKA by augmenting it with the *sketching* techniques and use it as a similarity metric in this paper. Sketching is a method to project a higher dimensional matrix into a chosen dimension. Thus, we sketch the two activation vectors into a fixed dimension (say M) which is lower than the number of samples (i.e., $M \ll n_k$) and also preserves the spectral structure. CKA with sketching (s-CKA) makes calculating the similarity faster and thus results in satisfying all the four necessary conditions.

Similarity Metric	Accuracy
CCA (R^2_{CCA})	20
SVCCA	19
PWCCA	20.5
EMD	15
s-CKA	**79**

Table 1: Source model identification accuracy. A sanity check on the MNIST dataset.

We propose a simple test to evaluate the extended CKA metric used in this paper. We create a federated MNIST digit recognition task. We first divide the MNIST dataset into two halves in a balanced manner, i.e., each half receiving an equal number of data samples per label. We further divide the first half into five parts with each part containing data samples associated with two labels, i.e., (0,1), (2,3), (4,5), (6,7), and (8,9). For our experiment, we adopted a slight variant of a baseline Convolutional Neural Network (CNN) model (two fully connected layers after two 5x5 convolution layers). A CNN model was then trained on each part to produce five source models. From the remaining half of the MNIST dataset, we pathologically partition it in a non-IID manner. We sort the data by digit labels and divide it into 100 shards of 300 size. We then assign each of 50 clients two shards. Thus most of the clients (target models) will receive two labels and a total of 600 input samples.

We used the original `FedAvg` algorithm for the training. First we train the clients for five rounds. After the clients are moderately trained, we calculate the similarity scores between each pair of client and source model. Now given a client model trained on labels (a, b), the most similar source model should be the one trained on the same labels. Thus, based on the labels of the client and selected most-similar source model, we calculate the accuracy of the chosen similarity metric.

The accuracy of identifying the most similar source model is shown in Table 1. As can be seen, while s-CKA achieves the highest accuracy, other similarity metrics performed worst. We repeat the experiment for each metric ten times, and report the average accuracies. We compare s-CKA with CCA, SVCCA, PWCCA, and Earth Movers Distance (EMD) [34]. Figure 1 shows the similarity scores between the different client models and each source model. The target models can be seen grouped by the respective similar source model. Even if one of the labels is common between the source and target models, the s-CKA gives a high similarity value in most of the cases.

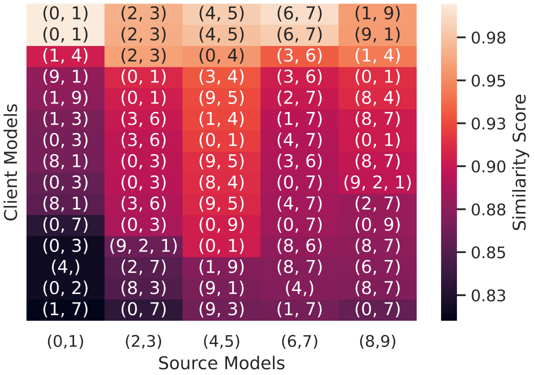


Figure 1: Targets grouped by the most similar target and source models. (a,b) are the labels of MNIST dataset.

3.3 `FedVote`: The Federated Voting Algorithm

Algorithm 1 presents the steps to select the best source models in a federated manner. The proposed framework works in two phases - selection and transfer. In the selection phase, we perform standard FL steps and keep updating the global model at each round. After a pre-defined R_{sel} number of rounds, we perform a similarity comparison between the available source models and local model at each of the chosen clients. The clients use their local training data to produce representations which are then used to output s-CKA scores between the local model and each of the source models. The representations are generated for a chosen layer of a model (generally the same layer across two

Algorithm 1: FedVote running on a client.

```
Input: Set of source models $\mathcal{M}_s$, global model $w$, $R_{sel}$
Output: Best $m$ models
for each round $R = 1, 2, \ldots$ do
    Choose $K$ clients
    for each client $k \in K$ do // in parallel
        $w \leftarrow$ LocalUpdate($k$,$w$) // client $k$ trains model on local data
        if $R == R_{sel}$ then
            for each $s \in \mathcal{M}_s$ do
                $z_w^{l_1}, z_s^{l_2} \leftarrow w(\mathcal{P}_k), s(\mathcal{P}_k)$ // create representations of layer $l_1$ of source
                 model and $l_2$ of client model; local data $\mathcal{P}_k$ is forward passed
                $votes \leftarrow$ s-CKA($z_w^{l_1}, z_s^{l_2}$) // calculate similarity score and increment
                 vote count for top $m \in \mathcal{M}_s$
            end
            return $votes$, $w$ to server
        else
    end
    Server node selects the highest voted $m$ models
end
```

models). The client then votes the most m similar models (higher s-CKA score). Each client sends the votes to the server, which are then aggregated to arrive at a consensus. Finally, the server selects the highest voted m models. In the transfer phase, we copy the parameters of the selected source model to the global model and continue the federated training process. For a positive transfer, we expect the same FL algorithm to achieve the same global test accuracy as the baseline in a lesser number of rounds. Effect of varying the R_{sel} on the accuracy of selecting the top model is further studied and presented in the appendix.

4 Experimental Results

We test the proposed framework on two tasks – Digit Recognition (DR), and Object Classification (OC). To carry out transfer experiments, each task consists of multiple source DNN models ($\mathcal{M}_s$) trained on publicly available datasets, and a single or a set of private target dataset for which the best models are to be selected and transferred. The DR task included five $\mathcal{M}_s$ each trained on MNIST [35], Fashion-mnist (FMNIST) [36], Kuzushiji-mnist (KMNIST) [37], Extended-mnist (EMNIST) [38], and STL10 [39] datasets. The client dataset was chosen to be USPS [40] dataset. In the OC task, we split the CIFAR100 [41] dataset into ten 10-class sub-datasets. One of the sub-dataset was chosen as a client dataset while others as $\mathcal{M}_s$. We used the same notations as in the original FL paper presented in [3] – K (total number of clients), E (local epochs), C (fraction of clients selected at each round), B (local batch size). Further, we define a performance metric R_{th} as the number of communication rounds to reach the desired test accuracy by an FL algorithm. Thus, we compare our framework against the original FedAvg and report our results in terms of R_{th} rather than FLOPS or wall-clock time. We implemented a standard CNN as in [3] for the DR task while for the more complex OC task, we used a deeper VGG-16 [42] model. For each task, we used the following hyperparameters as described in [3]: $K = 100$, $B = 50$, $C = 10$, and $E = 5$. The target dataset was split into clients in two ways - **IID** setting where each client was given a fixed number of random samples from the same distribution and **non-IID** in which the clients only received the training samples from a subset of labels.

4.1 Results

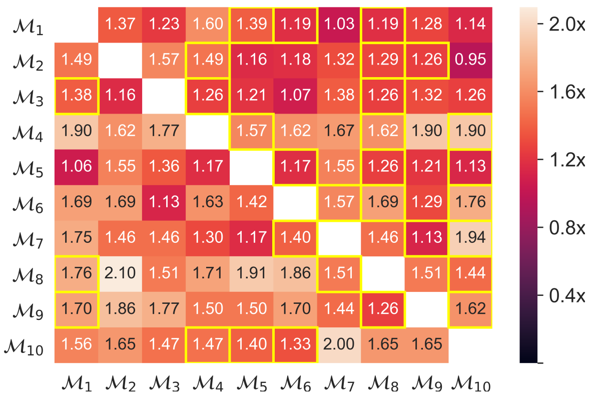
Table 2 presents the R_{th} obtained to achieve a test-set accuracy of 90% for different $\mathcal{M}_s \twoheadrightarrow$ USPS pairs of the DR task. Each of the source model is ranked by our proposed algorithm based on the s-CKA values. Except KMNIST, all the other source models provided a positive transfer. KMNIST dataset includes Kanji letters and is thus completely different from the USPS

$\mathcal{M}_s$	R_{th} (Non-IID)	R_{th} (IID)
Baseline	332	170
SVHN	66 (5.03×)	31 (5.45×)
MNIST	93 (3.56×)	51 (3.33×)
EMNIST	161 (2.06×)	94 (1.80×)
FMNIST	256 (1.29×)	111 (1.53×)
STL10	287 (1.15×)	120 (1.41×)
KMNIST	571 (0.58×)	225 (0.75×)

Table 2: Transfer performance on the DR task to reach a 90% test-set accuracy.

input images of digits. As can be seen from Table 2, the top three models selected by our algorithm - `SVHN`, `MNIST`, and `EMNIST`, provided the least R_{th} values. Source models reported in Table 2 are entirely different. However, it may happen that the available source models are closely related. Selecting the best source models in such a scenario raises the difficulty of the problem presented in this paper.

Figure 2 shows a transfer performance matrix for each of the ten models produced from the CIFAR100 splits. One of the datasets was chosen as a target while remaining as source models. The same was repeated for all the ten CIFAR100 splits. As can be seen from the figure, the top-3 models selected by our algorithm (cells with yellow borders) provide a sub-optimal performance for most of the $\mathcal{M}_{s_i} \twoheadrightarrow \mathcal{M}_{s_j}$ pairs. Since the models are still part of the same CIFAR100 dataset, differentiating their performance and avoiding a negative transfer shows the efficacy of the proposed algorithm.



	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$	$\mathcal{M}_8$	$\mathcal{M}_9$	$\mathcal{M}_{10}$
$\mathcal{M}_1$		1.37	1.23	1.60	1.39	1.19	1.03	1.19	1.28	1.14
$\mathcal{M}_2$	1.49		1.57	1.49	1.16	1.18	1.32	1.29	1.26	0.95
$\mathcal{M}_3$	1.38	1.16		1.26	1.21	1.07	1.38	1.26	1.32	1.26
$\mathcal{M}_4$	1.90	1.62	1.77		1.57	1.62	1.67	1.62	1.90	1.90
$\mathcal{M}_5$	1.06	1.55	1.36	1.17		1.17	1.55	1.26	1.21	1.13
$\mathcal{M}_6$	1.69	1.69	1.13	1.63	1.42		1.57	1.69	1.29	1.76
$\mathcal{M}_7$	1.75	1.46	1.46	1.30	1.17	1.40		1.46	1.13	1.94
$\mathcal{M}_8$	1.76	2.10	1.51	1.71	1.91	1.86	1.51		1.51	1.44
$\mathcal{M}_9$	1.70	1.86	1.77	1.50	1.50	1.70	1.44	1.26		1.62
$\mathcal{M}_{10}$	1.56	1.65	1.47	1.47	1.40	1.33	2.00	1.65	1.65	

Figure 2: Transfer performance matrix obtained from the ten 10-class CIFAR100 splits. Performance gain of top-3 selected models are highlighted with yellow coloured border.

In Fig. 3, we compare the proposed method with a random selection strategy on the OC task. For each client model, we average the transfer performance achieved over five random selections and compare the values with the best performing model selected by our algorithm. Even when the source models share a high similarity (as shown in Fig. 2), our work improves over a randomly selected $\mathcal{M}_s$.

One may argue that instead of selecting the best among the source models, it could be better to always choose generic models with large size such as ResNet [43] which are pre-trained on ImageNet [44] data. We experimented using a pre-trained generic ResNet model on the DR task and found the R_{th} to be 56. As reported in Table 2, the best R_{th} obtained using our method is 66, which is comparable to that of the generic model. Edge devices have limited resources. In addition, more computations increase the depletion of the battery. Hence, selecting smaller size expert models in contrast to a generic model justifies our work. Further, a better source model may achieve superior performance than a pre-trained ResNet. A similar form of conclusions is presented in [45].

5 Conclusions

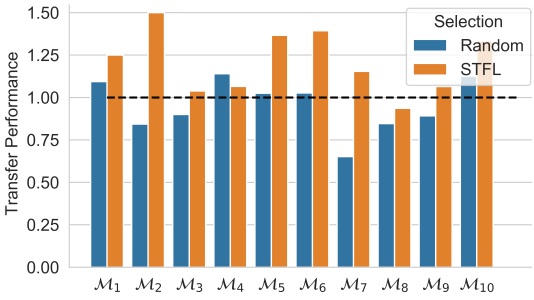


Figure 3: Comparison with the random selection strategy on OC task.

To address the problem of source model selection in Transfer Learning for federated scenarios, we propose SFTL, a framework which provides both selection and transfer of model parameters on the edge devices. We leverage the concepts from representation learning to compare the similarity of the client model and the source models. We provide a method which could be augmented to existing FL algorithms to improve both the communication cost and the accuracy of client models. Since this is an early work, we did not perform an extensive hyper-parameter search, which could further improve the transfer performance. The proposed work supports the principles of privacy and thus provides a method to perform resource-friendly computations at the edge. The raw data both from the source domains and clients are never shared. However, sharing model parameters can still result in privacy leaks. We intend to perform a privacy study and hyper-parameter search as the part of our future work.

References

[1] IDC. The growth in connected iot devices is expected to generate 79.4 zb of data in 2025, according to a new idc forecast. 2019.

[2] B McMahan, E Moore, D Ramage, and S Hampson. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[3] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[4] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[5] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

[6] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*, 2019.

[7] Yunlong Lu, Xiaohong Huang, Yueyue Dai, Sabita Maharjan, and Yan Zhang. Differentially private asynchronous federated learning for mobile edge computing in urban informatics. *IEEE Transactions on Industrial Informatics*, 2019.

[8] Xiaofei Wang, Chenyang Wang, Xiuhua Li, Victor CM Leung, and Tarik Taleb. Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching. *IEEE Internet of Things Journal*, 2020.

[9] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[10] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[11] Muhammad Jamal Afridi, Arun Ross, and Erik M Shapiro. On automated source selection for transfer learning in convolutional neural networks. *Pattern recognition*, 73:65–75, 2018.

[12] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

[13] Soheil Kolouri, Akif B Tosun, John A Ozolek, and Gustavo K Rohde. A continuous linear optimal transport approach for pattern analysis in image datasets. *Pattern recognition*, 51:453–462, 2016.

[14] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200, 2007.

[15] Yusuf Aytar and Andrew Zisserman. Tabula rasa: Model transfer for object category detection. In *2011 international conference on computer vision*, pages 2252–2259. IEEE, 2011.

[16] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[17] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[18] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[19] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[20] Barbara Plank and Gertjan Van Noord. Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1566–1576. Association for Computational Linguistics, 2011.

[21] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems 2020*, pages 429–450. 2020.

[22] Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas. Federated learning based proactive content caching in edge computing. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2018.

[23] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[24] K. A. Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[25] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948, Online, 26–28 Aug 2020. PMLR.

[26] Dashan Gao, Yang Liu, Anbu Huang, Ce Ju, Han Yu, and Qiang Yang. Privacy-preserving heterogeneous federated transfer learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2552–2559. IEEE, 2019.

[27] Yang Liu, Yan Kang, Chaoping Xing, Tianjian Chen, and Qiang Yang. A secure federated transfer learning framework. *IEEE Intelligent Systems*, 2020.

[28] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, pages 6076–6085, 2017.

[29] Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, pages 5727–5736, 2018.

[30] Natalia Y Bilenko and Jack L Gallant. Pyrcca: regularized kernel canonical correlation analysis in python and its applications to neuroimaging. *Frontiers in neuroinformatics*, 10:49, 2016.

[31] Ben Tan, Yu Zhang, Sinno Jialin Pan, and Qiang Yang. Distant domain transfer learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[32] Jie Song, Yixin Chen, Xinchao Wang, Chengchao Shen, and Mingli Song. Deep model transferability from attribution maps. In *Advances in Neural Information Processing Systems*, pages 6182–6192, 2019.

[33] Shuai Tang, Wesley J Maddox, Charlie Dickens, Tom Diethe, and Andreas Damianou. Similarity of neural networks with gradients. *arXiv preprint arXiv:2003.11498*, 2020.

[34] Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. Earth mover distance over high-dimensional spaces. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, page 343–352, USA, 2008. Society for Industrial and Applied Mathematics.

[35] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[36] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[37] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

[38] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[39] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[40] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[41] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[44] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[45] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018.

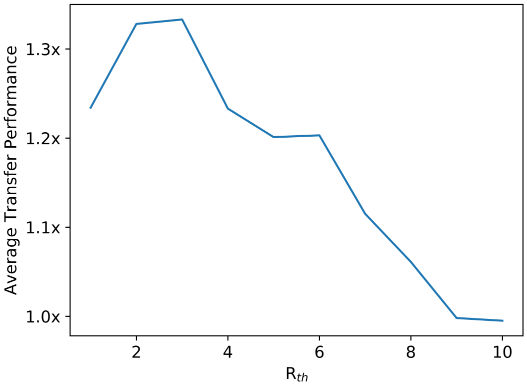


Figure 4: Effect of R$_{sel}$ on the transfer performance for the OC task.

Appendix A Effect of R_{sel} on the selection accuracy

Fig. 4 shows the variation in the transfer performance for different values of R_{sel} on the OC task. As can be seen from the figure, lower and very high values of R_{sel} may degrade the selection accuracy, which in effect reduces the transfer performance. Hence, selecting the best source models during the initial rounds of communication is not a wise choice. We found that training the client models for at least a few rounds lets the client model learn features which can be compared reliably.