ASCEND
November 15-17, 2021, Las Vegas, Nevada & Virtual
ASCEND 2021

10.2514/6.2021-4007

Forecasting Internet Demand in a Satellite Communication Network Using Transfer Learning

Skylar Eiskowitz*, Edward F. Crawley †, Bruce G. Cameron ‡
Massachusetts Institute of Technology, Cambridge, MA, 02139

Time series forecasting is increasingly important due to the massive production of data with new advances like the internet of things and smart cities. Time series forecasting models are typically developed and evaluated with the assumption that sufficient training data under the same distribution as the input data is available. In this paper, we use transfer learning to forecast for a new satellite user terminal that has limited training data. We pre-train an LSTM model on other user terminals that do have sufficient data, and then transfer the weights learned from this combined model in order to produce more accurate forecasts for a new target user terminal without sufficient training data. We use minute and hour level internet demand data of seven user terminals and propose a clustering framework to help decide which source user(s) to use to train the source model. Results suggest that transfer learning approaches can greatly benefit a target terminal with limited training data, on average reducing the MAE by 44% with just 10 days of hour level training data. However, the input data to transfer knowledge from must be chosen wisely. Clustering the source terminals with the target terminal and training the source model just on the source terminals in the same cluster as the target is a computationally inexpensive way to realize the benefits of transfer learning without negative transfer.

I. Introduction

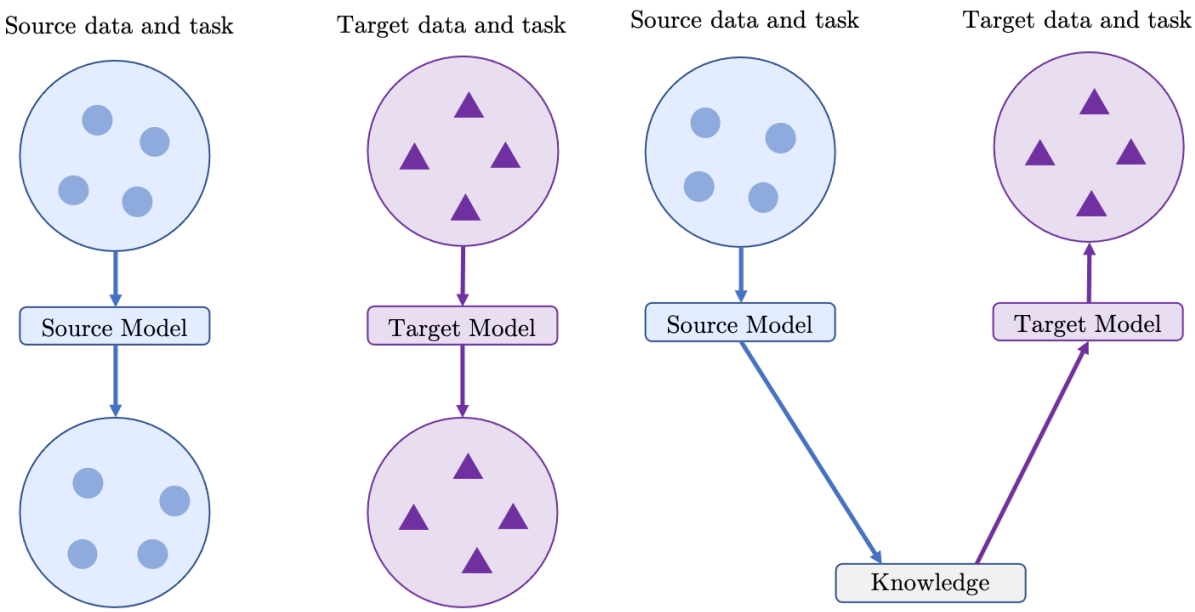
An important aspect of translating Machine Learning (ML) models to industry settings is consideration for the environment in which models will be deployed. As deeper neural networks are becoming more popular due to their powerful ability to capture complex patterns, they require an increasing amount of training data to avoid overfitting [1]. However, the necessary amount of labeled data to train deep neural networks can be expensive to produce, or even impossible. A couple of factors may contribute to a lack of the required training data:

1) Time-series data may change over a long period of time, leading to a big discrepancy from old data and new data
2) A new entity may need to be forecasted for, and there is little to no training data for that entity

A. Background

Transfer learning has emerged as a framework to help build accurate models that suffer from limited training data by transferring knowledge from one ML model to another, displayed in Fig. 1 [2]. Whereas traditionally, separate tasks train separate ML models, with transfer learning, a target task is able to take advantage of source knowledge, shifting the paradigm of needing to train separate models on vast amounts of target data. Instead, a model can learn from data in a different distribution, and also learn from a model that is performing a different but related task.

*Graduate student, Department of Aeronautical and Astronautical Engineering, eiskowit@mit.edu
†Ford Professor of Engineering, Department of Aeronautical and Astronautical Engineering, crawley@mit.edu
‡Director of System Architecture Group, Department of Aeronautical and Astronautical Engineering, bcameron@mit.edu



(a) Traditional machine learning **(b) Transfer learning**

Fig. 1 a) Traditional machine learning assumes training and evaluation on the same data and task, whereas b) transfer learning stores knowledge gained from solving a problem with a different but related task and data.

Transfer learning has been applied mainly to classification problems with great success [3], and recently to regression and clustering problems. Research aims to answer three questions [2]:

1) How to transfer?
2) What to transfer?
3) When to transfer?

Answering the question of 'how to transfer' requires an implementation or algorithm that will be able to extract knowledge from a source and share it with a target. In a time series forecasting setting, multiple approaches have been proposed, typically involving a pre-training process where a base model is trained on source data [4, 5]. Applying transfer learning to time series forecasting problems is a relatively new field, and authors are able to show that with a simple notion of weight sharing across source and target models, the target's forecasts are improved both in accuracy and computational resources. Specifically, it is shown that using a deep LSTM network and transferring weights, a model using transfer learning outperformed a model without transfer learning, no matter how much target training data was available [4]. More involved approaches have included the use of a deep CNN called DTr-CNNm, which is able to use the source dataset not only in pre-training, but also in training the target model [6]. Further, a hybrid algorithm that incorporates transfer learning, online sequential extreme learning machine with kernals (OS-ELMK) and ensemble learning was proposed that produced similar or better forecasts with an order of magnitude improvement in computational time [7]. Lastly, boosting approaches have been adapted for transfer learning problems, notably in TrAdaBoost, where source datasets are combined with the limited target data, and at each boosting step, the relative weights of target instances are adapted such that when source instances are misclassified, their weights are decreased. This way, the algorithm identifies source instances that are most similar to target data and ignores those that are dissimilar [8].

With insight into how to transfer knowledge, answering 'what to transfer' probes what known knowledge from other sources will help the target model, i.e., what is the shared knowledge between a *source* (knowledge you have) and the *target* (what you want to forecast for). Determining what to transfer for time series data is challenging and scarcely addressed in current works, yet it is important so that *negative transfer* is avoided, where transferring knowledge from a source actually degrades the target model's performance. Researchers turn to Multi Source Transfer Learning to handle this issue [9, 10], where given multiple options of sources to transfer knowledge from, measures of relativity between datasets is crucial. For time series forecasting, these include similarity metrics like Dynamic Time Warping (DTW) and Jensen-Shasson (JS) [6], Maximum Mean Discrepancy statistical indicator, [11] and clustering techniques like K-means

clustering [12].

Lastly, 'when to transfer' asks in which situations transferring knowledge should be done, which most current works take for granted by assuming the source and targets are related to each other enough such that negative transfer will be avoided [2].

B. Objectives

We focus on a satellite communication problem where the demand for Internet traffic must be forecasted, and user terminals are the entities that need to be forecasted for. As a new user terminal joins, there is little training data from which to build a model. A satellite operator must plan for the allocation of resources, so they must over-provision due to this uncertainty in demand, which leads to wasted unused capacity. To better forecast the new user terminal's internet demand, we show that knowledge can be transferred from similar user terminals.

C. Overview

This paper will touch on all three questions that transfer learning research aims to address, how to transfer, what to transfer, and when to transfer. In Section II we formulate problem and introduce the experiments. In Section II.C we answer the question of 'how to transfer,' in Section II.D we answer the question of 'what to transfer,' and in Section II.E we answer the question of 'when to transfer.' The results of these approaches are discussed in Section III.

II. Methods

A. Problem Formulation

A domain, $\mathcal{D}$ consists of a marginal probability distribution $P(X)$ over the feature space $\mathcal{X}$, where $X = \{x_1, ..., x_n\}$. Two domains being different may mean they have different feature spaces or different marginal probability distributions [2]. A task, $\mathcal{T}$ is composed of a label space $\mathcal{Y}$ and the conditional probability $P(Y|X)$ that a model learns from training examples in a supervised learning setting.

In transfer learning, given a source domain, $\mathcal{D}_S$, a source task $\mathcal{T}_S$, a target domain, $\mathcal{D}_T$, and a target task $\mathcal{T}_T$, the goal is to learn the target conditional probability distribution $P(\mathcal{Y}_T|\mathcal{X}_T)$ in $\mathcal{D}_T$, but from information learned from $\mathcal{D}_S$ and $\mathcal{T}_S$. We apply *inductive* transfer learning, meaning that limited labeled data in $\mathcal{D}_T$ is required to *induce* $P(\mathcal{Y}_T|\mathcal{X}_T)$, and $\mathcal{D}_S \neq \mathcal{D}_T$, $\mathcal{T}_S \neq \mathcal{T}_T$ [2]. In the case of user terminals in a satellite communication network, $\mathcal{D}_T$ represents a new user terminal that has limited training examples, and $\mathcal{D}_S$ is a similar user terminals that has a vast amount of training data.

B. Algorithm Implementations

1. Data

We simulate a multi source transfer learning problem with an internet demand dataset provided by a satellite operator, displayed in Fig. 2. It contains the forward data rate in Mbps for seven user terminals at a 1-minute resolution from 2/1/20 to 3/31/20. To simulate a new user entering the network, we partition the user terminals into six source datasets and one target dataset, and the terminal that is taken to be the target terminal is truncated to emulate limited training data. Throughout the upcoming sections, we analyze both the raw minute level data as well as down sampled hourly data.

2. Metrics

Mean Absolute Error (MAE) is chosen to compute the accuracy of the forecasted time series to the actual values:

$$MAE = \sum_{t=1}^{N} \frac{|Actual_t - Predicted_t|}{N} \tag{1}$$

As a method to compare improvements in MAE across terminals, we use percentage change in MAE:

$$\%\ Change\ in\ MAE = \frac{MAE_{TL} - MAE_{baseline}}{MAE_{baseline}} \tag{2}$$

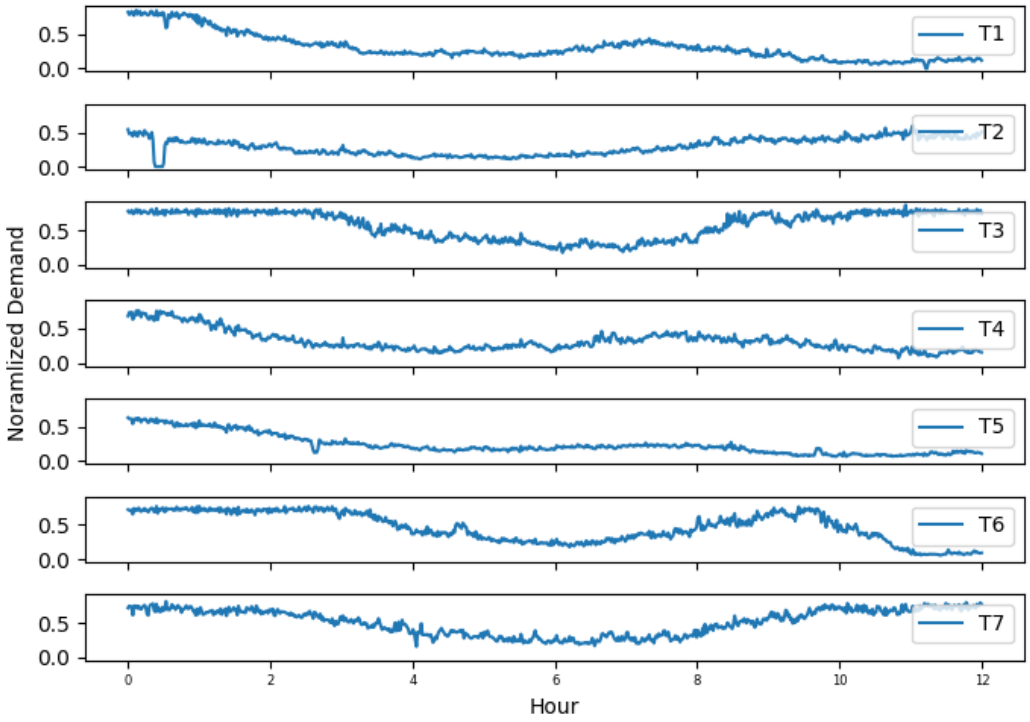


Fig. 2 Half-day profiles for the seven user terminals showing the different input data distributions.

3. Test Cases

The performance is computed on the target time series, which is first split into training and testing subsets, with a holdout period of seven days. The data is first imputed by first removing 'drops' by calculated large percentage differences in the data and replacing dropped values with previous data points, then scaled before training and testing. The training lengths for the source terminals are 50,000 samples for the minute level dataset and 1,440 samples for the hourly dataset, but the total training length into the source model varies based on the experiment due to different amounts of source terminals being fed in. The amount of training data for the target model also varies based on the experiment, as this is a parameter we explore to answer the question 'when to transfer' in Section II.E.

C. How to Transfer

1. Long Short-Term Memory (LSTM) Weight Sharing

An LSTM network is a type of recurrent neural network (RNN), which models time series autoregressively, exploiting dependencies between sequential inputs by means of a hidden state and three gates: input, forget, and output gates [13]. An LSTM network is chosen as the base architecture shown in Fig. 3 for both the source and target models. First, the source model is trained on source data (which data is selected is described in Section II.D), then the learned weights are transferred to a target model. Finally, the target model will undergo a training phase with what limited data it has. Since the weights of the target model are left open to changing with the same learning rate as the source model, this transferring of knowledge is embodied as the target model's 'warm start.'

The parameters were tuned using a grid search procedure on the source model. Each terminal showed similar performance on the selected model except for terminal five, which is excluded from the analysis in select figures for visualization and scaling purposes. The selected parameters are shown in Table 1 and those that were tuned differently for the down sampled hourly data are shown in Table 2.

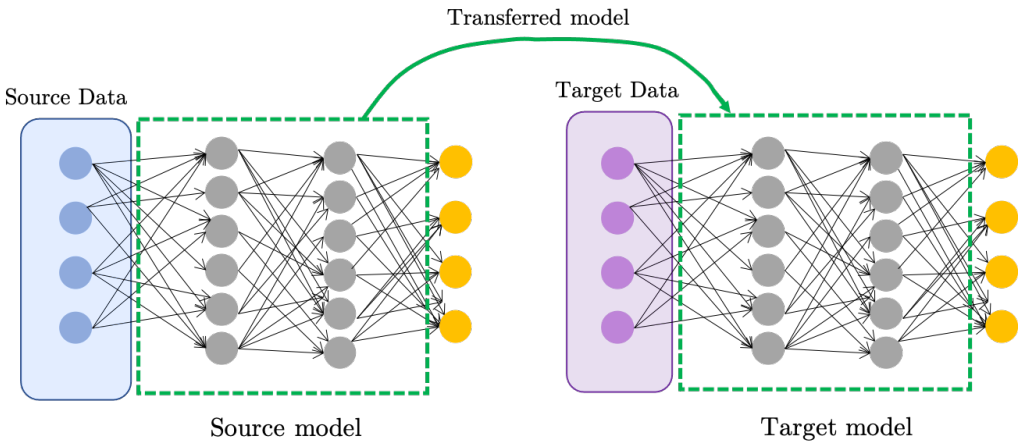


Fig. 3 The weight sharing algorithm first trains a model with source data, then transfers the weights to a new target model. Then the limited target data will be used to update the weights in a training period of the target model.

Table 1 Common LSTM parameters across the raw and down sampled datasets

Parameter	Value
RNN layers	2
RNN nodes	75
Dropout rate	0.2
Loss function	mse
Optimzier	Adam

D. What to Transfer

1. Grid Search

In the first implementation of selecting which data to input in the source model of the weight sharing algorithm, we perform a grid search, exhaustively examining the different source and target terminal pair-wise combinations.

2. K-Means Clustering

To avoid having to exhaustively search which terminal provides the best knowledge to transfer, clustering the datasets together allows for shorter computational times as well as the possibility that knowledge from not just one, but multiple terminals can be used to train the source model. K-means clustering is an unsupervised learning technique, drawing conclusions of groupings from unlabeled datasets. It has been successfully applied in a variety of industries like economics [14], energy [15], and computer vision [16]. Clustering has typically been applied to transfer learning classification problems, where clustering itself is the target task [17–19], but in this paper clustering is used as a step to

Table 2 Different LSTM Parameters between the raw and down sampled datasets

Parameter	minute level Data	Hourly Data
Embedding dimension	70	15
Batch size	360	45
Epochs	9	20

help in the input selection process for the larger regression problem as in [20, 21].

Fig. 4 shows the approach where both source and target data is input to a clustering algorithm. We implement K-means clustering, where the cluster amount is set apriori, and similar time series are clustered together based on a similarity metric. We implement K-means clustering with tslearn's clustering library* with Dynamic Time Warping (DTW). DTW, shown in Fig. 5 allows for comparisons of time series with different lengths, which is necessary because the target terminal has significantly less data than the source terminals it needs to be clustered with. Finally, the source data that is clustered with the target data is used to train the source model.

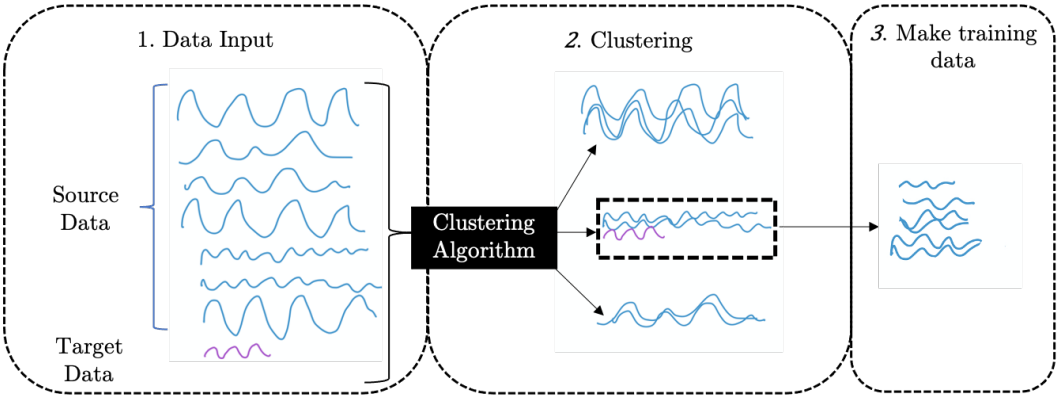
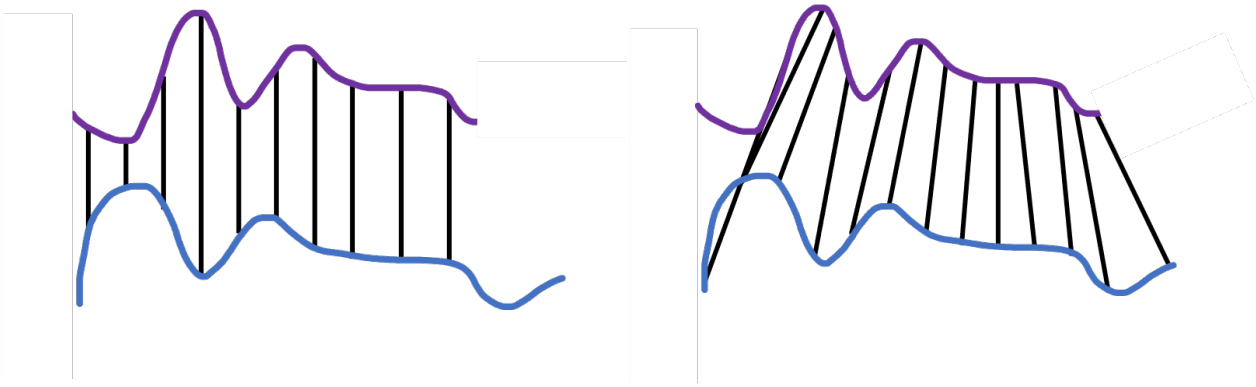


Fig. 4 The source data selection methodology consists of 1. Accumulating both source and target data, 2. Clustering the data using the Dynamic Time Warping (DTW) similarity measure 3. Making training examples from the source data that is clustered with the target data.



(a) Euclidean distance **(b) Dynamic Time Warping**

Fig. 5 To measure similarity between two time series, the a) Euclidean distance simply matches values at the same time, where b) Dynamic Time Warping allows for series with different lengths or speeds to be matched, allowing for time series with different lengths to be compared.

E. When to Transfer

Finally, an analysis is performed varying the amount of target training data available to help answer the question 'when to transfer.' When the target has accumulated enough training data, it is intuitive to question when training a model strictly on target data will produce comparable or better results, i.e. when transfer learning is no longer necessary.

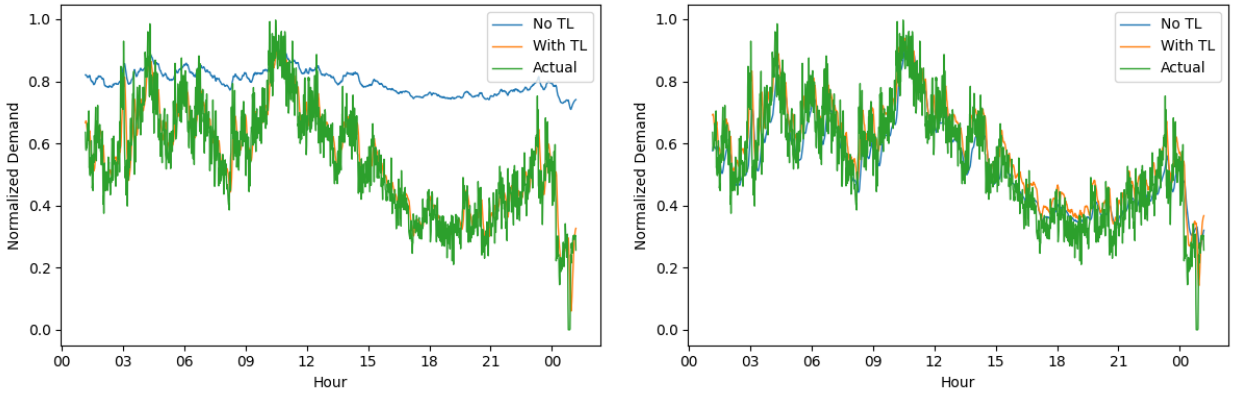
*https://tslearn.readthedocs.io/en/stable/gen_modules/clustering/tslearn.clustering.TimeSeriesKMeans.html

III. Results

Throughout this section we simulate a target terminal by limiting the amount of data it has. Only one target is simulated at a time, but which terminal out of the seven that is designated as the target is varied in the experiments. We compare five models:

1) **Lag**: The Lagged (i.e., naïve or persistence) forecast is the baseline forecast where the forecasted value is the time series shifted by the offset amount. The forecast is thus the last known value.
2) **No TL**: This represents the RNN in Section II.C.1 trained on the limited amount of target data only
3) **All**: This source model is the same RNN trained on all six source terminals
4) **Best**: This source model is the same RNN trained on only one terminal that produced the best MAE results based on the exhaustive grid search described in Section II.D.1
5) **Cluster**: This model uses the methodology described in Section II.D.2, first clustering six source terminals, then transferring weights from the RNN trained on the source data in the same cluster as the target

First, the normalized actual demand of the minute level data is plotted in green in Fig. 6 along with the No TL model in blue, and the best performing transfer learning model in orange. In Fig. 6a, the target only has one day of its own data to train on, and in Fig. 6b, the target has 10 days of training data. In Table 3 we compute the MAE for the week forecast and see that that the performance of the No TL model greatly benefits with more data, and by the tenth day, it is able to outperform the best transfer learning model. The performance of the model with TL is not sensitive to the amount of target training data.



(a) 1 day of minute level target training data

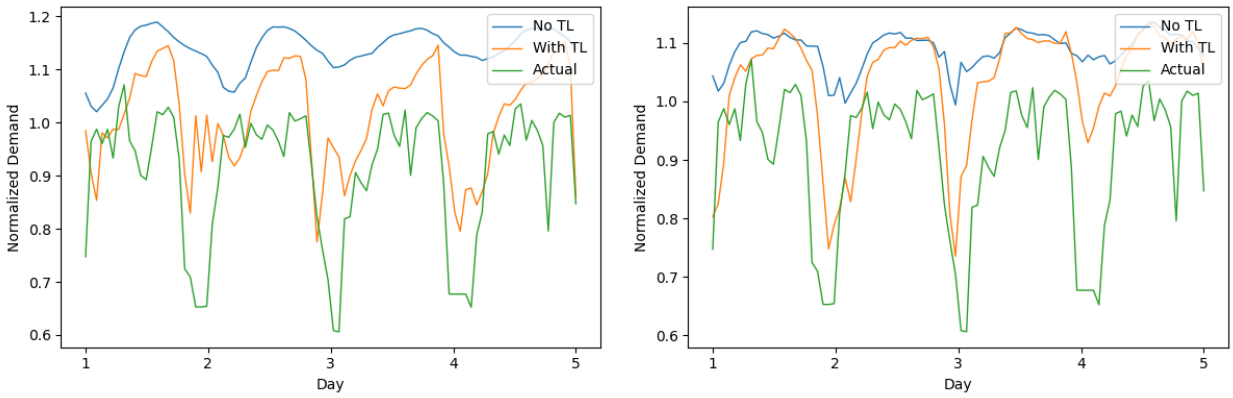
(b) 10 days of minute level target training data

Fig. 6 As the target user terminal only has one day of training data (a), using transfer learning greatly benefits the forecast, but as more training data is accumulated (b), this effect of using transfer learning is less evident.

Table 3 MAE of minute level data forecasting over a week

Target Training Days	MAE No TL (Mbps)	MAE With TL (Mbps)
1	48.2	5.3
5	7.8	5.1
10	4.5	5.1

The same analysis is done on the down sampled hourly data, where the forecasting problem is now forecasting for the next hour rather than the next minute, and the target data is varied to have 10 to 20 training days in Fig. 7. With hourly data, Table 4 shows that the best transfer learning model always outperformed the No TL, likely due to the limited amount of target data that is left after down sampling.



(a) 10 days of hourly target training data

(b) 20 days of hourly target training data

Fig. 7 With down sampled data, the target user terminal is left with a scarce amount of training data even at 20 days (b), so using transfer learning greatly benefits the forecast. Even though the performance of the No TL model improves with more training data, it still does not perform well with such a limited amount of data, and transferring knowledge is greatly beneficial.

Table 4 MAE of hourly data forecasting over a week

Target training days	MAE Without TL (Mbps)	MAE With TL (Mbps)
10	22.3	13.2
15	24.0	11.2
20	17.6	11.4

A. What to Transfer

In order to determine which terminals are the best to transfer knowledge from, first the exhaustive grid search described in Section II.D.1 is performed on the minute level data. Each terminal on the x-axis in Fig. 8 represents a target terminal, and each terminal on the y-axis is the source terminal that trains the source model whose weights will be transferred to the target model. Each subsequent heat map possesses more target training data (from 1 day to 5 days, and finally 10 days), and the value in the heat map is the percentage change in MAE with transfer learning. The closer the value is to zero, the lower the benefit of transfer learning. Note that we choose not to include terminal five for minute level analysis as it was much harder to forecast for in the short-term than in the long-term. The results affirm that as more target training data is available, the benefit of transfer learning decreases (the heat maps get darker).

Another takeaway from Fig. 8 is the asymmetry in knowledge transfer. When one source terminal best transfers knowledge to a certain target terminal, it does not necessarily mean that the target terminal will best transfer knowledge to the source terminal when their roles are reversed.

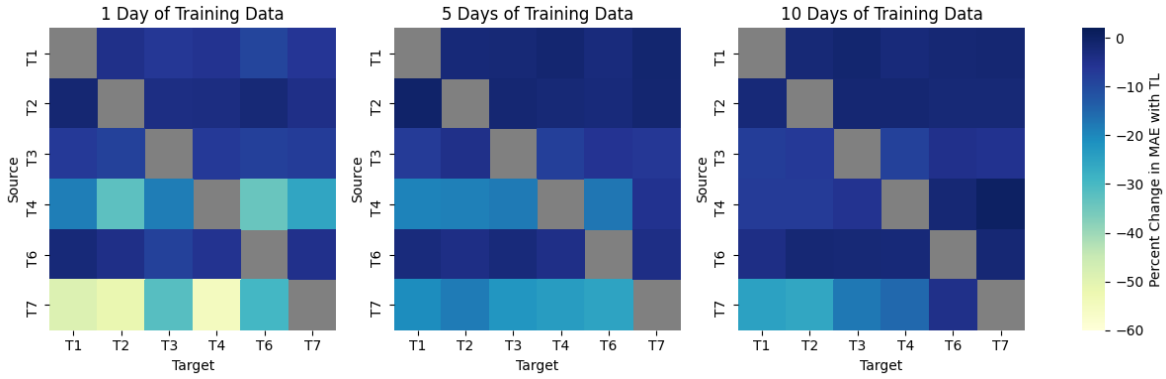


Fig. 8 As more days of target training minute level data is available (each subsequent figure has more training data), the percent change in MAE with TL goes closer to zero, showing the lower added benefit from transfer learning.

The same analysis was performed on hourly data in Fig. 9. Noting the different scales between the figures, positive numbers are now included to account for the negative transfer that often occurred. The results show that transfer learning has the ability to benefit a target model's forecasting performance, but it is essential to understand where to take the knowledge from. Otherwise, the model is prone to negative knowledge transfer.

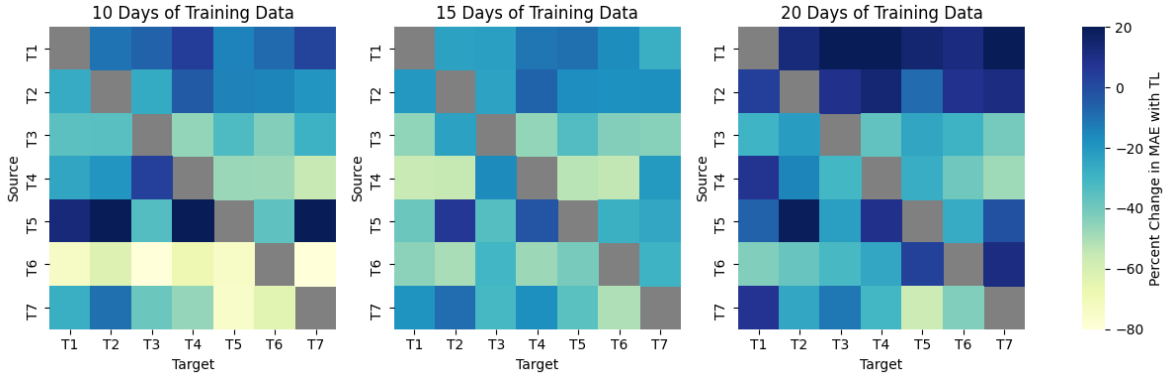


Fig. 9 As the target (y-axis) accumulates more hourly training data (each subsequent figure has more training data), the percentage MAE improvement using transfer learning decreases. In some cases negative transfer occurs where the model performance degrades with transfer learning.

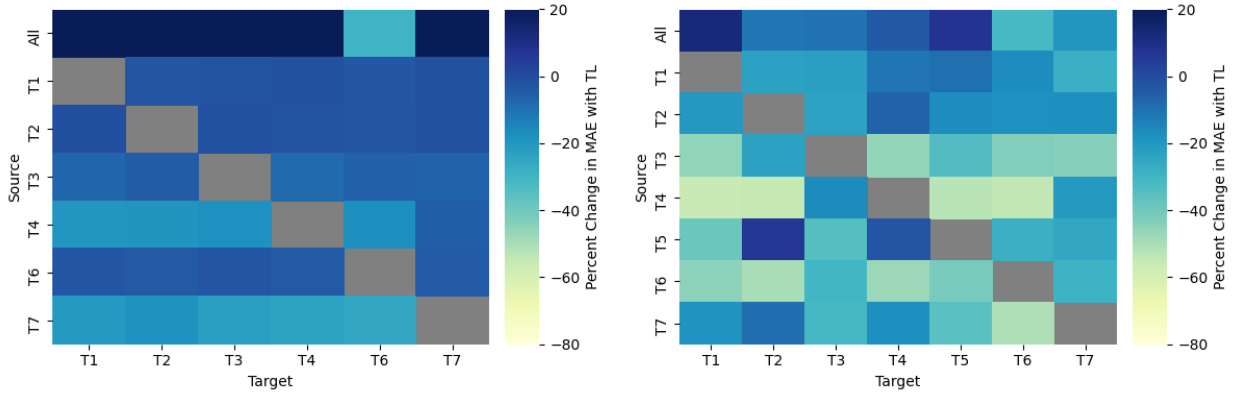
A sensitivity analysis is performed on how the initial choice of the best terminal to transfer knowledge from changes as the target accumulates more minute level data. In Table 5, each row represents a different scenario of which terminal is the target terminal, and the chosen source terminal is fixed based on which source terminal led to the best results when the target only had one day of training data. This chosen source is kept constant as the target accumulates more data, but the results suggest that by the fifth day, there is often a new 'best' source terminal. Operationally, this would require a new exhaustive grid search to be done on which terminal(s) to transfer knowledge from as the target accumulates data. However, in this scenario, transfer learning's payoff may be deemed negligible at the tenth day anyway.

From an operational point of view, Fig. 8 and Fig. 9 can be used to map which terminals should transfer knowledge to each other, but it is computationally expensive and would require a held-out test set from the target terminal, which is infeasible due to the already limited amount of data. To mitigate this, we experiment with training the source model on all the available data, a simple approach that requires no knowledge of source and target terminal compatibility. The results are shown in Fig. 10 as the top most row on the source axis. The minute level data target terminal is simulated with 5 days of target data and the hourly data with 15 days of target data. Here, the importance of selecting the right input source data is highlighted, as often negative transfer occurs i.e., the percent change in MAE is a positive number when the source model is trained on all terminals, especially in the minute level case where negative knowledge transfer

Table 5 Sensitivity analysis on how the best source terminal changes as the target accumulates more data shows that for minute level data, once 10 days of target data is accumulated, the source terminal picked on the first day is no longer the best source terminal.

Target Terminal	Chosen Source Terminal	% Change in MAE From the Chosen to the Next Best Source Terminal (%)		
		1 day	5 days	10 days
T1	T7	30.2	1.3	16.8
T2	T7	19.3	-0.8	18.8
T3	T7	13.6	4.7	11.9
T4	T7	48.0	15.8	6.6
T6	T3	4.7	-7.7	-2.9
T7	T3	18.3	-1.3	-5.5

is occurring for most terminals. This establishes that training a source model on all available data severely degrades the model performance.



(a) Minute level data **(b) Hour level data**

Fig. 10 The percent change in MAE with transfer learning when the source model is trained on all terminals usually degrades model performance, and it is beneficial to rather choose the best source terminal from an exhaustive search. This is more evident in the minute level dataset as negative knowledge transfer degrades the performance for most target terminals.

So far the best approach to selecting which source terminals to transfer knowledge from is to perform and exhaustive grid search over all separate terminals to see which best transfers knowledge, however this is problematic for two reasons:

1) It is computationally intensive
2) This selection process only considers one terminal as the source terminal at a time, and perhaps knowledge from multiple terminals should be taken advantage of
3) This would require a held-out test set on target data to evaluate each model's performance, which is infeasible due to the already limited amount of data.

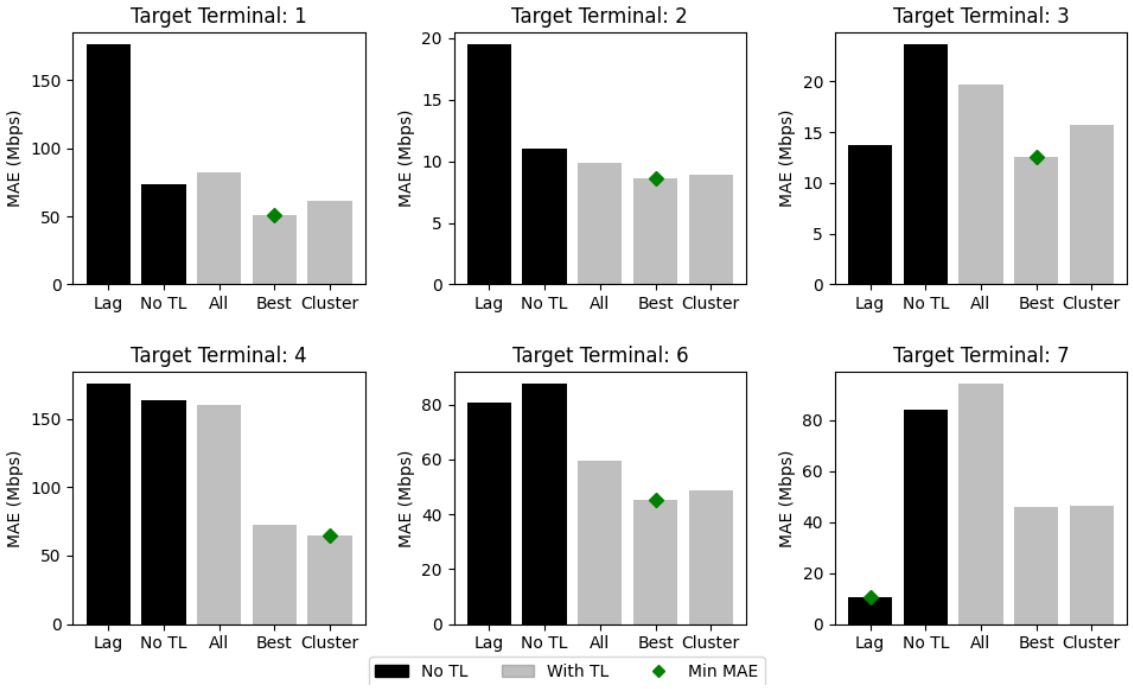


Fig. 11 Full comparison of all input training source methods for hourly data with 15 days of target training data available. Significant improvement is shown with transfer learning approaches compared to the baseline model and the lagged forecast. Although choosing the single best terminal as source data most often outperformed other methods, it is computationally expensive compared to the clustering approach which achieved similar performances.

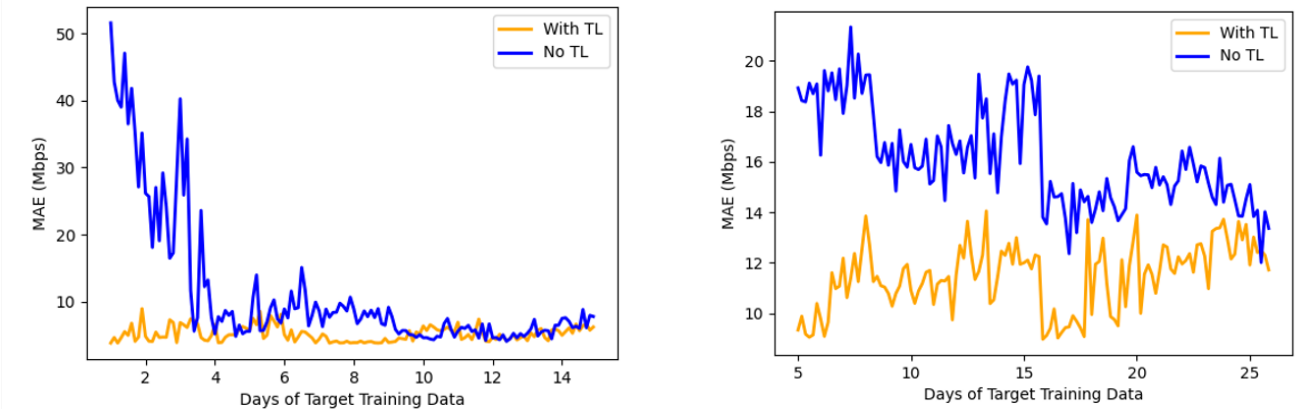
Table 6 Analysis of the different number of clusters in K-Means clustering

Number of clusters	Silhouette Score
2	0.330
3	0.287

Thus, we implement the clustering methodology discussed in Section II.D.2, clustering the source terminals and using the data in the cluster that the target falls into to train the source model. Since there are only six terminals used for the target and source analysis, this limits the amount of clusters that can be made such that there will be more than one terminal in each cluster. We found that allowing for more than three clusters does not ensure enough training data per cluster in order to then train a model. Silhouette scores for feasible number of clusters are shown in Table 6, and we decide on two clusters based on its higher Silhouette score. A full comparison of all source training input selection methods is shown in Fig. 11. Although choosing the single best terminal as source data most often outperformed other methods, the clustering approach is more computationally feasible and achieved a similar performance.

B. When to Transfer

Next, a more detailed analysis is performed to answer the question 'when to transfer knowledge.' In Fig. 12, the MAE for a one week held out test set is plotted as a function of how much target data is available for training. The plot shows a transfer from terminal three to terminal one. For this source and target pair, using transfer learning consistently improves the MAE, but this effect becomes negligible once the target has accumulated around one week's worth of data for the minute level case and around three weeks' worth of data for the hourly case.



(a) minute level data **(b) Hourly data**

Fig. 12 MAE as a function of how much target data is available for training shows a significant performance improvement as the target accumulates more data without using transfer learning (blue). Transfer learning is able to mitigate the issue of not enough training data and consistently achieves a lower MAE than without using transfer learning. This is evident in a) minute level data and b) hourly data

IV. Conclusion

Transfer learning can greatly benefit a forecast for a target terminal with limited training data simply with a weight sharing algorithm that transfers weights from a source LSTM to a target LSTM, serving as a warm-start. However, the input data to transfer knowledge from must be chosen wisely. Although taking the best source terminal and training a source model with just its data provided the best performance four of the six times, it is infeasible to perform this exhaustive search through all possible source datasets. Thus, clustering provides a practical and far less computationally expensive approach to choosing what the source dataset(s) should be, while maintaining comparable model performance. Lastly, a sensitivity analysis shows that if the best source terminal(s) to transfer knowledge from are chosen early on (when the target only has one day's worth of data), that source remains the best choice until around the fifth day, when there was found to be a better option for three out of the six scenarios. However, at this point, we also show that transfer learning's payoff begins to degrade, meaning it is sufficient to stick with the best source terminal(s) chosen early on to continue to transfer knowledge from.

Future work includes further exploring clustering to aid in 'what to transfer' for a multi-source transfer learning problem. Different approaches to clustering source data with target data can be explored, for instance, instead of clustering full series of sources of data, training examples of all the source data can first be produced, and then clustered, allowing for a collection of training instances to be clustered with the target, not just full series of sources. This way, pieces of sources can be found to best contribute to knowledge transfer. Lastly, more separate source entities could form better clusters to then train the source model.

Acknowledgments

This work was supported by SES S.A. The authors would like to thank SES S.A. for their financial and non-financial support during the development of this work.

References

[1] Hippert, H. S., Pedreira, C. E., and Souza, R. C., "Neural networks for short-term load forecasting: A review and evaluation," *IEEE Transactions on Power Systems*, Vol. 16, No. 1, 2001, pp. 44–55. https://doi.org/10.1109/59.910780.

[2] Pan, S. J., and Yang, Q., "A survey on transfer learning," , 2010. https://doi.org/10.1109/TKDE.2009.191.

[3] Fung, G. P. C., Yu, J. X., Lu, H., and Yu, P. S., "Text classification without negative examples revisit," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 1, 2006, pp. 6–20. https://doi.org/10.1109/TKDE.2006.16, URL http://ieeexplore.ieee.org/document/1549824/.

[4] Laptev, N., Yu, J., Rajagopal, R., Cho, M., Song, M., Yoo, S., Reijers, H. A., Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T., "Reconstruction and Regression Loss for Time-Series Transfer Learning," *International Journal of Forecasting*, Vol. 36, No. 3, 2019, pp. 15239–15249.

[5] Gupta, P., Malhotra, P., Vig, L., and Shroff, G., "Transfer Learning for Clinical Time Series Analysis using Recurrent Neural Networks," *arXiv*, 2018. URL http://arxiv.org/abs/1807.01705.

[6] Ye, R., and Dai, Q., "Implementing transfer learning across different datasets for time series forecasting," *Pattern Recognition*, Vol. 109, 2021. https://doi.org/10.1016/j.patcog.2020.107617.

[7] Wang, X., and Han, M., "Online sequential extreme learning machine with kernels for nonstationary time series prediction," *Neurocomputing*, Vol. 145, 2014, pp. 90–97. https://doi.org/10.1016/j.neucom.2014.05.068.

[8] Dai, W., Yang, Q., Xue, G. R., and Yu, Y., "Boosting for transfer learning," *ACM International Conference Proceeding Series*, Vol. 227, 2007, pp. 193–200. https://doi.org/10.1145/1273496.1273521.

[9] Eaton, E., and Desjardins, M., "Selective Transfer Between Learning Tasks Using Task-Based Boosting," Tech. rep., ???? URL www.aaai.org.

[10] Fang, M., Guo, Y., Zhang, X., and Li, X., "Multi-source transfer learning based on label shared subspace," *Pattern Recognition Letters*, Vol. 51, 2015, pp. 101–106. https://doi.org/10.1016/j.patrec.2014.08.011, URL https://linkinghub.elsevier.com/retrieve/pii/S0167865514002712.

[11] Gu, Q., and Dai, Q., "A novel active multi-source transfer learning algorithm for time series forecasting," *Applied Intelligence*, 2020. https://doi.org/10.1007/s10489-020-01871-5.

[12] Le, T., Vo, M. T., Kieu, T., Hwang, E., and Rho, S., "Using a Cluster-Based Strategy for Transfer Learning in Smart Building," ????

[13] Gers, F., "Learning to forget: continual prediction with LSTM," *9th International Conference on Artificial Neural Networks: ICANN '99*, Vol. 1999, IEE, 1999, pp. 850–855. https://doi.org/10.1049/cp:19991218, URL https://digital-library.theiet.org/content/conferences/10.1049/cp_19991218.

[14] Falat, L., and Pancikova, L., "Quantitative Modelling in Economics with Advanced Artificial Neural Networks," *Procedia Economics and Finance*, Vol. 34, 2015, pp. 194–201. https://doi.org/10.1016/s2212-5671(15)01619-6.

[15] Liu, G., Yang, J., Hao, Y., and Zhang, Y., "Big data-informed energy efficiency assessment of China industry sectors based on K-means clustering," *Journal of Cleaner Production*, Vol. 183, 2018, pp. 304–314. https://doi.org/10.1016/j.jclepro.2018.02.129.

[16] Vedaldi, A., and Fulkerson, B., "Vlfeat - An open and portable library of computer vision algorithms," *MM'10 - Proceedings of the ACM Multimedia 2010 International Conference*, ACM Press, New York, New York, USA, 2010, pp. 1469–1472. https://doi.org/10.1145/1873951.1874249, URL http://dl.acm.org/citation.cfm?doid=1873951.1874249.

[17] Yang, Q., Chen, Y., Xue, G.-R., Dai, W., and Yu, Y., "Heterogeneous Transfer Learning for Image Clustering via the Social Web," Tech. rep., 2009. URL http://www.flickr.com.

[18] Chen, Y., Zhao, Z., Liu, J., Shen, Z., and Liu, M., "Cross-People Mobile-Phone Based Activity Recognition. Loving hut View project Optimization of multi-agent organizations View project Cross-People Mobile-Phone Based Activity Recognition," ???? https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-423, URL https://www.researchgate.net/publication/220815130.

[19] Guyon, I., Dror, G., Lemaire, V., Taylor, G., and Aha, D. W., "Unsupervised and transfer learning challenge," *Proceedings of the International Joint Conference on Neural Networks*, 2011, pp. 793–800. https://doi.org/10.1109/IJCNN.2011.6033302.

[20] Le, T., Vo, M. T., Kieu, T., Hwang, E., Rho, S., and Baik, S. W., "Multiple Electric Energy Consumption Forecasting Using a Cluster-Based Strategy for Transfer Learning in Smart Building," *Sensors*, Vol. 20, No. 9, 2020, p. 2668. https://doi.org/10.3390/s20092668, URL https://www.mdpi.com/1424-8220/20/9/2668.

[21] Zhang, C., Zhang, H., Qiao, J., Yuan, D., and Zhang, M., "Deep Transfer Learning for Intelligent Cellular Traffic Prediction Based on Cross-Domain Big Data," *IEEE Journal on Selected Areas in Communications*, Vol. 37, No. 6, 2019, pp. 1389–1401. https://doi.org/10.1109/JSAC.2019.2904363.