Affinity-Driven Transfer Learning For Load Forecasting

Ahmed Rebei , Manar Amayri , Nizar Bouguila
The Concordia Institute for Information Systems Engineering (CIISE)
Concordia University, Montréal, QC, H3G1T7, Canada
{ahmed.rebei, manar.amayri, nizar.bouguila}@concordia.ca

Abstract

We propose a novel approach for electricity load forecasting that leverages the task affinity score to measure the distance between different tasks. The task affinity score provides a more effective method for measuring similarity between tasks in a transfer learning context. We demonstrate the efficacy of the task affinity score through empirical analysis using a synthetic dataset. Our results show that the task affinity score outperforms other intuitive metrics such as the loss function for task selection. To apply this approach, we present the Affinity-Driven Transfer Learning (ADTL) algorithm for load forecasting. The ADTL algorithm optimizes the transfer learning process by leveraging knowledge from pre-trained models and datasets to improve the accuracy of load forecasting for new and previously unseen datasets. We validate the effectiveness of the ADTL algorithm by testing it on two real-world challenging datasets. Overall, our study highlights the importance of the task affinity score in transfer learning for load forecasting applications.

1 Introduction

Load forecasting is the process of predicting future electricity demand. It is an essential task for electricity grids and power system operators in order to allow proper functioning and maintain a reliable supply of electricity. Therefore, accurate forecasting is very important to plan and manage electricity generation, transmission, storage, and distribution. In addition, it is crucial to ensure the stability and reliability of power grids, as well as for making informed decisions about electricity generation and transmission capacity [Yu *et al.*, 2019]. In recent years, the demand for electricity is higher and higher due to the emergence of new technologies. The complexity and variability of electricity data are increasing and making it more challenging to address the load forecasting problem. Many factors can impact the accuracy of load forecasting. For example, weather patterns, economic conditions, changes in electricity consumption patterns, extreme weather events such as heatwaves or cold spells, changes in the level of economic activity, and the adoption of energy-efficient technologies can significantly alter electricity demand. Additionally, the consumption patterns of individual customers can vary significantly over time, making it difficult to accurately predict the overall load on the grid.

To address these challenges, a variety of statistical algorithms have been developed for load forecasting. Before deep learning models became popular, several statistical methods were used to predict future values, such as autoregressive integrated moving average ARIMA [Pappas *et al.*, 2008] and exponential smoothing [Christiaanse, 1971]. However, these methods rely on the assumption that the data is stationary, which is not the case with electricity demand. Electricity demand is often volatile and exhibits complex trends and seasonal patterns that traditional statistical models cannot capture. Therefore, many machine learning methods were proposed to mitigate this issue [Hong *et al.*, 2020] such as support vector regressors [Hong, 2009], fuzzy logic [Ranaweera *et al.*, 1996], artificial neural network (ANN) [Park *et al.*, 1991; Chen *et al.*, 2001; Lu *et al.*, 1993], radial basis functional network (RBFN) [Xia *et al.*, 2010] and hybrid methods [Lv *et al.*, 2021; Alhussein *et al.*, 2020; Liao *et al.*, 2021]. In recent years, neural networks, especially recurrent neural networks (RNNs), have become popular for forecasting because they can model nonlinear features and take into account the temporal structure of the data, making them more effective at capturing the evolution of load data [Mansouri and Akbari, 2014]. More complex approaches were proposed in the literature combining different types of neural networks. In [Song *et al.*, 2022], for example, the authors propose a new forecasting approach that considers both temporal and spatial features using a graph convolution network (GCN) and a multiresolution convolution neural network (CNN) for short-term wind power forecasting. Similarly, Jiang [Jiang *et al.*, 2022] focused on the development of a new learning mechanism for enhancing the mapping capability of multi-step demand in building energy forecasting and proposed a deep-chain echo state network (DCESN) to effectively prevent error accumulation compared to sliding-window echo state networks and LSTM models.

In addition to the trivial forecasting problem, the context of load forecasting presents some other problems. Mainly, data scarcity and source volatility. One promising approach for addressing the load forecasting problem is the use of transfer learning, a machine learning technique that allows a model to quickly adapt to new tasks by learning from past experiences.

Transfer learning has been successfully applied to a variety of tasks in various fields, including computer vision, natural language processing, and robotics [Niu *et al.*, 2020; Weiss *et al.*, 2016; Gopalakrishnan *et al.*, 2017; Zoph *et al.*, 2016; Hua *et al.*, 2021]. In the context of load forecasting, transfer learning has the potential to improve the efficiency of load prediction by allowing the model to learn from a diverse set of past forecasting tasks and adapt to new ones more quickly. In this paper, we propose the use of transfer learning for solving the load forecasting problem. We begin by reviewing the existing literature on load forecasting and transfer learning, highlighting the challenges and limitations of traditional load forecasting approaches and the potential benefits of using transfer learning. We then describe our proposed transfer learning-based approach for load forecasting, including a review of the Task Affinity Score and how it is used in the context of transfer learning. Finally, we present the results of our experimental evaluation, demonstrating the effectiveness of our approach. Overall, this work's contributions can be summarized as follows: 1) In an empirical study, we demonstrate the usefulness of using the task affinity score as a measure of task distance for selecting the nearest source task from which to transfer knowledge. (see section 3.1), 2) Propose a transfer learning approach integrating the Task Affinity Score as a distance metric for source task selection. (see section 3.2), and 3) Improve the efficacy of load forecasting deep learning models in training time and prediction score. (see section 4). This paper is structured as follows. Section II provides a comprehensive review of the relevant literature related to transfer learning in load forecasting. Section III outlines the methodology including the formulation of the task affinity score and the algorithm used in the experimental section. In Section IV, we present two case studies that illustrate the practical application of our methodology. Section V outlines future work that could build upon our findings, and concludes the paper by summarizing our main findings and outlining their implications for future research and practice.

2 Litterature Review

Accurate electric load forecasting is crucial for the safety and efficient operation of modern electric power systems and various methods have been proposed to improve it. To address the challenge of limited training data, several studies have proposed the use of transfer learning techniques. For example, in [Gao *et al.*, 2020], the authors proposed two deep learning models and a transfer learning framework to improve energy consumption prediction accuracy for buildings with limited data and demonstrated the effectiveness of the models through a case study of three office buildings. The proposed models, a sequence-to-sequence (seq2seq) model and a two-dimensional convolutional neural network with an attention layer, showed an improvement in forecast accuracy over a long memory network under a poor information state. Similarly in [Li *et al.*, 2021], the authors propose a transfer learning-based Artificial Neural Network model for one-hour ahead building energy prediction, to address the challenge of insufficient data for training data-driven predictive models for new buildings and existing buildings without advanced Building Automation Systems. The study uses data from 400 non-residential buildings from the open-source Building Genome Project to test the proposed method and finds that transfer learning can effectively improve the accuracy of Back Propagation Neural Network (BPNN) based building energy models for information-poor buildings with limited training data. The research also identifies the most influential building features that influence the effectiveness of transfer learning, particularly in selecting appropriate source buildings and datasets. In [Fang *et al.*, 2021], a novel hybrid deep transfer learning strategy was proposed to improve the accuracy of energy predictions in buildings with limited historical measurements. The strategy uses a combination of long short-term memory and a domain adversarial neural network to extract temporal features and domain-invariant features between source and target buildings, respectively. Experiments show that this strategy significantly enhances the building energy prediction performance compared to models trained on target-only data or source-only data, and without transfer learning. The results provide guidance for effectively using existing building data resources. Another approach that is intended to be applied effectively for intelligent energy management in smart buildings was introduced in [Le *et al.*, 2020]. The authors propose a new framework called MEC-TLL for forecasting electric energy consumption in smart buildings using transfer learning and Long Short-Term Memory models. The framework uses a k-means clustering algorithm to group the daily load demand of many profiles in the training set and then applies transfer learning to LSTM models to reduce computational time. The proposed approach is tested on two smart buildings in South Korea and results show that it is able to reduce computational time while achieving superior performance compared to other models. Zhou et al. proposed a load forecasting model for an Integrated Energy System (IES) to improve energy scheduling [Zhou *et al.*, 2020]. The model addresses the problem of insufficient data for new users in the IES by using a combination of Bidirectional Generative Adversarial Networks (BiGAN), data augmentation, and transfer learning techniques. The model is compared to ten other data-driven models on two different types of users, residential and commercial, and found to be more accurate on average for each user type respectively. The study also analyzes the impact of sample size and shows that the proposed model can improve the efficiency of other predictive models and can be used for load forecasting even when there is a lack of data. Peng et al. used a multi-source transfer learning guided ensemble LSTM method (MTE-LSTM) to address the problem of insufficient energy data [Peng *et al.*, 2022]. The method uses a two-stage source-domain building matching method to find similar buildings and an LSTM modeling strategy that combines transfer learning and fine-tuning to generate basic load forecasting models for the target building. An ensemble strategy is then used to weigh the output results of the basic forecasting models. The method was applied to multiple real buildings and was able to achieve high-precision load forecasting results when the target building data was relatively limited. In another work [Cai *et al.*, 2019], the authors found that a two-layer transfer learning-based architecture for

short-term load forecasting (STLF) can improve the forecasting accuracy of load in a target zone. The architecture utilizes load data from source zones and includes an inner layer where latent parameters are introduced to represent the differences in electricity consumption behavior between zones. In the outer layer, an iterative algorithm is developed to assign variant weights to datasets according to their fitness to the latent parameter-assisted model. Results from case studies show that this proposed STLF architecture is able to improve the forecasting accuracy of classic STLF algorithms, particularly when the load data of the target zone is limited. Another work [Hooshmand and Sharma, 2019] presents a solution to the problem of developing predictive models for energy assets, such as electricity loads and PV power generations, using limited data. The authors proposed an energy predictive model based on convolutional neural networks (CNNs) to capture patterns, trends, and seasonalities in energy assets time series. They then propose a transfer learning strategy to improve the model's performance when there is limited training data. The approach is demonstrated in a use case of daily electricity demand forecasting and results show that the transfer learning strategy improves existing forecasting methods. The authors in [Wu and Lin, 2022] addressed the problem of insufficient data by training graph neural network (GNN) based models in newly built residential neighborhoods. They proposed a transfer learning framework that uses knowledge learned from other areas with abundant data to assist the model learning for the area with limited data. Specifically, the authors propose an "attentive transfer framework" that ensembles GNN models trained from source domains and the GNN model trained on the target domain. The framework assigns dynamic weights to different GNN models based on the input data. The proposed framework was tested on real-world datasets and the results show that it is effective in various scenarios.

3 Methodology

We present the task affinity score as a task-distance metric. We develop empirical proof of the effectiveness of TAS as the distance between two tasks. In this context, we define a task as a model-dataset pair for simplicity. For a source dataset $\mathcal{X}_a$, we train a model f_a using $\mathcal{L}_a(\theta)$ as a loss function. The instigated distance is applied using a target dataset $\mathcal{X}_b$ and the same model f_a using the same loss function $\mathcal{L}_a(\theta)$.

3.1 Task affinity score

Theoretical formulation

The task affinity score is a measure based on the Fisher information to approximate the similarity between two tasks. It is used to determine how easily one task can gain knowledge from another task. It can help identify which tasks are most closely related and therefore most likely to benefit from shared knowledge [Le *et al.*, 2022]. We first need to define the Fisher Information Matrix to calculate the task affinity score. This matrix is a measure of the amount of information that is gained about a particular task after training. It is calculated by taking the expectation of the second derivative of the log-likelihood of the loss function with respect to the task parameters. Once we define the Fisher Information Matrix, we use it to calculate the task affinity score. This is done by comparing the Fisher Information Matrices of the source and target tasks to determine a pseudo distance between them. The greater the distance, the higher the task affinity score will be, indicating that knowledge gained from the source task is more likely to be useful for learning the target task. For a neural network f_θ with weights θ and the negative log-likelihood loss function $\mathcal{L}(\theta)$, we define the Fisher Information as:

$$\mathcal{F}(\theta) = \mathbb{E}\left[\nabla_\theta \mathcal{L}(\theta) \nabla_\theta \mathcal{L}(\theta)^t\right] = -\mathbb{E}\left[H(\mathcal{L}(\theta))\right] \quad (1)$$

where $\mathcal{H}$ is the Hessian matrix. To calculate the Fisher Information Matrix in practice, we use an empirical approach as shown in Equation 2.

$$\hat{\mathcal{F}}(\theta) = \frac{1}{|\mathcal{X}|} \sum_{i \in \mathcal{X}} \nabla_\theta \mathcal{L}^i(\theta) \nabla_\theta \mathcal{L}^i(\theta)^t \quad (2)$$

where for dataset $\mathcal{X}$, the $\mathcal{L}^i(\theta)$ is the loss on the i^{th} data point in the dataset. The Task Affinity Score between the source dataset $\mathcal{X}_a$ and the target dataset $\mathcal{X}_b$ is calculated using the Fréchet distance based on the Fisher Information Matrices of the network f_θ. f_θ is trained on the dataset $\mathcal{X}_a$. Specifically, the TAS is defined as follows:

$$s[a,b] = \frac{1}{\sqrt{2}} Trace\left(\mathcal{F}_{a,a} + \mathcal{F}_{a,b} - 2\left(\mathcal{F}_{a,a}\mathcal{F}_{a,b}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}} \quad (3)$$

where $F_{a,a}$ is the Fisher Information matrix of f_θ with the source dataset $\mathcal{X}_a$, $F_{a,b}$ is the Fisher Information matrix of f_θ with the target dataset $\mathcal{X}_b$. The full Fisher Information matrix is not used because it is computationally expensive to calculate in the large space of neural network parameters. Instead, we calculate the diagonal approximation of the Fisher Information matrix. These matrices are also normalized to have a unit trace. As a result, the TAS formula in equation 3 can be simplified to the following form:

$$\begin{aligned} s[a,b] &= \frac{1}{\sqrt{2}} \|\mathcal{F}_{a,a}^{\frac{1}{2}} - \mathcal{F}_{a,b}^{\frac{1}{2}}\| \\ &= \frac{1}{\sqrt{2}} \left[\sum_i \left((\mathcal{F}_{a,a}^{ii})^{\frac{1}{2}} - (\mathcal{F}_{a,b}^{ii})^{\frac{1}{2}}\right)\right]^{\frac{1}{2}} \end{aligned} \quad (4)$$

where $\mathcal{F}_{a,a}^{ii}$ and $\mathcal{F}_{a,b}^{ii}$ denote the diagonal element of $\mathcal{F}_{a,a}$ and $\mathcal{F}_{a,b}$ respectively. The value of the TAS ranges from 0 to 1 where a score of 0 indicates a perfect similarity and a score of 1 indicates complete dissimilarity.

Empirical justification

To investigate the usefulness of task affinity score as a measure of similarity between tasks, we conducted a simulation on synthetic data. In this simulation, we added a linear trend and Gaussian noise information to a pseudo-sine function in a recursive manner to simulate the increasing difference between the datasets. The pseudo-sine function we used tries to mimic the behavior of load data from week to week as described in equation 5. Figure 1 shows the difference between some of the datasets we used in this simulation.

$$y_i = G(t).\left[1 + \sin(2\pi t - \frac{\pi}{2})\right] + \tau_i(t) + \epsilon_i(t) \quad (5)$$

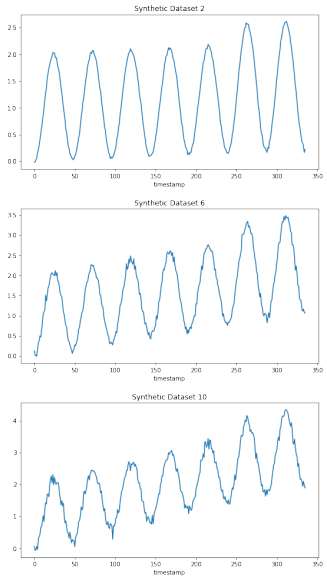

Figure 1: Synthetic data 2, 6, and 10

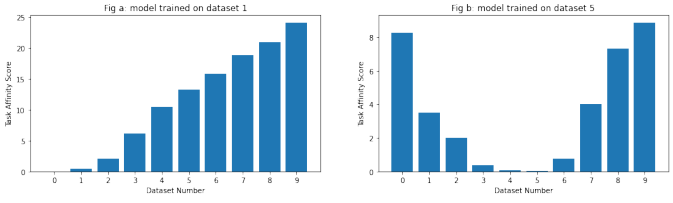

Figure 2: Task Affinity Score between the different data and task 0 and 5, respectively.

$$G(t) = \mathbb{1}_{weekdays}(t) + \alpha.\mathbb{1}_{weekdays}(t) \qquad \alpha = 1.7$$
$$\tau_i(t) = k_i t \qquad k_i \in [0, 1]$$
$$\epsilon(t) \sim \mathcal{N}(0,\, \sigma_i^2) \qquad \sigma_i \in [0.1, 1]$$

where t, k_i, and σ_i are linearly spread in the respective intervals. We used 48 data points to simulate one day (Although it is not necessary, we tried to mimic the 30-minute sampling rate used in the datasets in the experimental section). We made use of 10 datasets. Our results showed that as the difference between the datasets increased, the task affinity score consistently increased, indicating that the task affinity score effectively captured the increasing dissimilarity between the tasks. This is clear from Figure 2 where we train two models for a few epochs on dataset 1 and dataset 5. As we go further away from the main dataset, the task affinity score increases. This simulation supports the intuition behind using task affinity score as a measure of similarity between tasks, as it demonstrates that the metric is sensitive to changes in the differences between the datasets. Overall, our simulation results support the use of task affinity score as a reliable and valid measure of the dissimilarity between tasks.

TAS vs MSE

In this section, we compare the task affinity score (TAS) and the mean squared error (MSE) as metrics for measuring the distance between tasks. We trained a fully connected neural network and a long short-term memory (LSTM) model on the same datasets from the previous experiment (first and sixth datasets) and compared the distances between all tasks to these target datasets. To evaluate the performance of the TAS and MSE metrics, we computed the distance between every task (trained models and respective datasets) and the target task using both metrics and compared the results. The results in figure 3 show that the TAS metric outperforms the MSE loss in terms of selecting the appropriate initial task to transfer the knowledge. Comparing figure 2 and figure 3, we can see how the loss function does not show the same pattern as the TAS distance. In particular, the TAS was more reliable at identifying the nearest tasks and distinguishing dissimilar tasks, allowing the model to converge faster while requiring fewer training epochs. In contrast, the MSE loss was not consistent, leading to less efficient model selection. On the other hand, we trained different fully connected neural networks and LSTMs with random weights and calculated the number of epochs to get the same performance as the nearest (and the second nearest neural network). We show the results in table I. In conclusion, the empirical study demonstrates that the TAS metric is a superior choice for measuring the distance between tasks, as compared to the MSE loss. The TAS metric is more accurate and consistent in identifying similar tasks and distinguishing dissimilar tasks.

Source	nearest dataset performance		second nearest dataset performance	
	LSTM	FCN	LSTM	FCN
dataset 0	7	3	7	5
dataset 1	7	3	7	5
dataset 2	7	3	8	4
dataset 3	8	6	5	5
dataset 4	5	4	5	3
dataset 5	6	3	6	5
dataset 6	8	3	7	4
dataset 7	8	3	7	4
dataset 8	5	5	7	6
dataset 9	7	4	7	6

Table 1: Number of epochs an LSTM (a fully connected ANN respectively) needs to match the MAPE performance of a one-epoch-trained pre-trained model of the nearest dataset and the second nearest dataset.

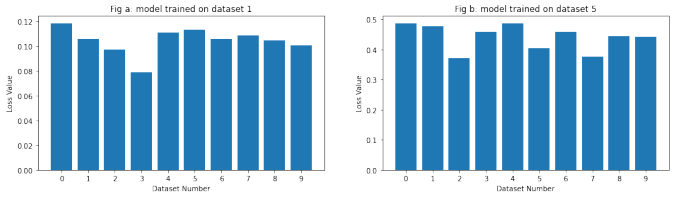

Figure 3: The TAS metric outperforming the MSE loss in terms of selecting the appropriate initial task to transfer the knowledge.

3.2 Affinity-Driven Transfer Learning

As shown in figure 4 and detailed in algorithm 1, the algorithm has two steps: A learning step and a transfer learning step. In the first step, we train different algorithms on different elements of a particular grid. The elements are usually household historical electricity data. The models are then stored for future use. The second step is the selection of the

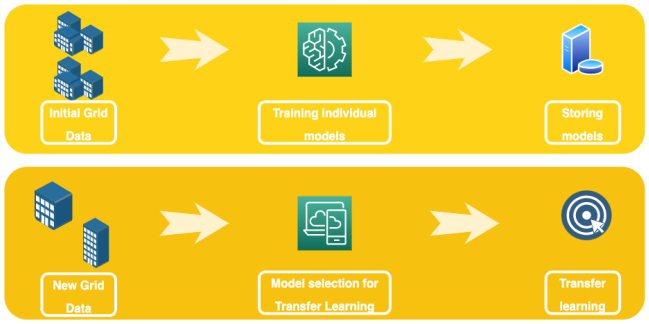

Figure 4: Affinity-Driven Transfer Learning Diagram.

right model to transfer knowledge from which we have new elements added to the grid by calculating the nearest task in terms of the task affinity score.

Algorithm 1 Affinity-Driven Transfer Learning

Input: Old grid elements Electricity Demand (OGTS)
Input: New grid element Electricity demand (NGTS)
Output: h^* ***I - pretraining:***
1: train h_i models on $\forall i \in \{1..m\}$ dataset from OGTS
II - Transfer Learning:
2: $h^* = \min_{h_i, i \in \{1..m\}} \ TAS(\tau_i, tau_{new})$
3: Train h^* for few epochs
4: **return** h^*

3.3 Models and metrics

Fully Connected Neural Nets

A fully connected network, also known as a fully connected layer or a dense layer, is a type of artificial neural network in which each neuron in a layer is connected to every neuron in the previous layer. Each neuron in a fully connected layer receives input from all the neurons in the previous layer [Goodfellow *et al.*, 2016]. Let's consider a fully connected layer with n inputs and m outputs [LeCun *et al.*, 2015], where the inputs are represented by a vector $\mathbf{x}$ of size n, and the outputs are represented by a vector $\mathbf{y}$ of size m. Each output y_i is computed as a weighted sum of the inputs x_j, plus a bias term b_i, and then passed through an activation function g:

$$y_i = g(\sum_{j=1}^{n} w_{ij} x_j + b_i)$$

where w_{ij} represents the weight of the connection between the jth input and the ith output, and b_i represents the bias term for the ith output. This equation can also be represented in matrix form:

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

where $\mathbf{W}$ is a weight matrix of size $(m \times n)$, $\mathbf{b}$ is a bias vector of size m, $\mathbf{x}$ is the input vector of size n, and σ is the sigmoid function that operates on the elements of the output vector $\mathbf{y}$. In the case of a multilayer neural network, the output of one fully connected layer is typically fed as input to the next fully connected layer, and so on, until the final output layer is reached.

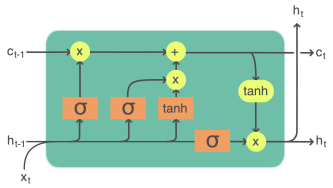

Figure 5: A Long-Short Term Memory LSTM cell.

LSTMs

LSTM is a gated recurrent neural network [Hochreiter and Schmidhuber, 1997]. It is used in various areas such as image generation [Gregor *et al.*, 2015], speech recognition [Graves *et al.*, 2013], natural language processing [Mikolov *et al.*, 2010], and time series forecasting [Taieb and Atiya, 2015]. LSTM uses the same weights on every time stamp. It takes the input sequence element by element and carries hidden information from one timestamp to the next one as shown in Figure 5. The classic Recurrent Neural Network (RNN) fails to carry information from long in the past because of the gradient vanishing problem. On the other hand, LSTM uses gates to carry different information when scanning the input sequence. So, instead of one simple activation function, LSTM uses the equations 6-11:

$$i_t = \sigma(W_i.h_{t-1} + U_i.x_t + P_i.C_{t-1} + b_i) \quad (6)$$
$$f_t = \sigma(W_f.h_{t-1} + U_f.x_t + P_f.C_{t-1} + b_f) \quad (7)$$
$$\tilde{C}_t = \psi(W_c.h_{t-1} + U_c.x_t + b_c) \quad (8)$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (9)$$
$$o_t = \sigma(W_o.h_{t-1} + U_o.x_t + P_o.C_{t-1} + b_o) \quad (10)$$
$$h_t = o_t \odot \psi(C_t) \quad (11)$$

Here, the subscripts i, f, and o denote respectively the different gates: input, forget, and output. h denotes the hidden state vector, and C is the long-term state vector. W_i, W_f, W_o, and W_c represent the weight matrices of the hidden information from the last timestamp. U_i, U_f, U_o, and U_c represent the weight matrices of the input information and P_i, P_f, P_o, and P_c represent the weight matrices of the long-term state C. The terms b_i, b_f, and b_o are biases of the gates. σ is the sigmoid operator, ψ is the hyperbolic tangent function (tanh) and $\odot$ is the element-wise multiplication operator.

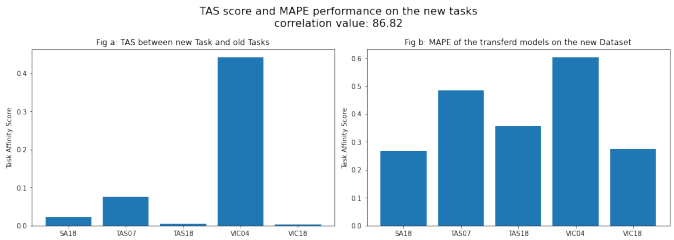

Figure 6: Average Performance on the different Query sets.

Metrics

To evaluate the accuracy of our proposed model, we relied on commonly used evaluation metrics, which are: the Root

Mean Square Error (RMSE) and the Mean Absolute Percentage Error (MAPE) presented in equations (12) and (13).

$$RMSE(y, \hat{y}) = \sqrt{\frac{\sum_{t=1}^{n} |y_t - \hat{y}_t|^2}{n}} \quad (12)$$

$$MAPE(y, \hat{y}) = \frac{1}{n} \sum_{t=1}^{n} |\frac{y_t - \hat{y}_t}{y_t}| \quad (13)$$

4 Case Study

In this experimental section, we present two case studies that focus on comparing the forecasting performance on the AEMO dataset between using the ADTL algorithm and an initialized network. In the second case study, we extend our analysis to investigate the performance of the nearest model and random network over different forecast horizons. By doing so, we aim to identify whether the relative performance of each model remains consistent across different time horizons.

4.1 Case Study 1: Australian Dataset

The Australian Energy Market Operator (AEMO) dataset [Operator, 2014] is a collection of energy market data from the National Electricity Market (NEM) in Australia. The NEM is a wholesale electricity market that covers the eastern and southeastern parts of Australia. The AEMO dataset includes data on electricity demand and supply in the NEM. This data is collected in real-time from market participants, including electricity generators, retailers, and transmission network service providers, and is used to manage the operation of the electricity grid and ensure a reliable supply of electricity to consumers. In this work, we selected a few datasets from different parts of the market. We covered different years and locations to get the spatial and temporal differences between the datasets. The time series are selected from Queensland, New South Wales, Victoria, South Australia, and Tasmania and are detailed in table II. In this study, we split

Dataset Name	Location	Year
NSW03	New South Wales	2003
NSW18	New South Wales	2018
QLD03	Queensland	2003
QLD18	Queensland	2018
SA03	South Australia	2003
SA18	South Australia	2018
TAS07	Tasmania	2007
TAS18	Tasmania	2018
VIC04	Victoria	2004
VIC18	Victoria	2018

Table 2: Datasets Locations and years used in experiment 1

Source	Nearest task			Second nearest task			Random FCN		
Number of epochs	0	1	5	0	1	5	0	1	5
NSW03	**0.33**	**0.27**	**0.14**	0.55	0.49	0.41	0.98	0.92	0.72
NSW18	**0.23**	**0.22**	**0.15**	0.54	0.47	0.32	1.18	1.08	0.85
QLD03	**0.37**	**0.29**	**0.24**	0.63	0.58	0.37	1.78	1.40	1.22
QLD18	**0.26**	**0.21**	**0.20**	0.62	0.49	0.46	1.33	1.29	0.94
SA03	**0.29**	**0.20**	**0.18**	0.50	0.48	0.43	1.54	1.49	1.02

Table 3: MAPE performance of the selected pre-trained neural network with comparison to a random neural network.

our datasets into two subsets: a meta subset and a query subset. The meta subset was used to train multiple models, and the query subset was used to evaluate the performance of the transferability of these models. Specifically, we trained 5 different models on the 5 datasets from the meta subset and evaluated their performance when used on the datasets from the query subset.

Source	Nearest Task Neural Network Performance (MAPE)			Average of random Neural Networks Performances (MAPE)		
	10%	20%	50%	10%	20%	50%
NSW03	**0.28**	**0.28**	**0.27**	0.65	0.61	0.58
NSW18	**0.34**	**0.33**	**0.24**	0.52	0.52	0.48
QLD03	**0.51**	**0.46**	**0.40**	0.44	0.43	0.41
QLD18	**0.37**	**0.34**	**0.27**	0.69	0.63	0.59
SA03	**0.48**	**0.47**	**0.29**	0.68	0.57	0.53

Table 4: Performance of the selected pre-trained neural network and a random neural network on down-sampled datasets

To evaluate the transferability of knowledge from the meta-tasks to new tasks, we calculated the task affinity score distance between the meta-tasks and each one of the query tasks. We then measured the average performance after transferring knowledge from the meta-models to each task from the query set. We found a high correlation between the MAPE performance and the task affinity score distance. In figure 6, we can see two subfigures labeled Fig.6.a and Fig.6.b, respectively. Fig.6.a illustrates the Task Affinity Score (TAS) applied to each dataset from the metaset with regard to the QLD18 dataset. On the other hand, Fig.6.b displays the Mean Absolute Percentage Error performance of each model from the metaset trained on the QLD18 dataset from the query dataset. This figure shows a very high correlation between the performance of the models and the distance between tasks calculated by TAS. We listed the different correlations of the same experiment with the different datasets from the query set in table V. The correlation between the proposed TAS distance and the MAPE performance is consistently high with a mean value of 88.78%. In this experiment and to ensure the stability of the TAS metric, we experimented with each one of the Query sets 10 times and averaged the results. These results indicate that the transferability of knowledge from the meta-models to new models is influenced by the similarity between the meta-tasks and the query tasks. In particular, the closer the task affinity score distance between the two sets of tasks, the better the transferability of knowledge. In addition to the results discussed above, we trained the nearest and the second nearest models to further compare the performance using the TAS information and train the model from scratch. In table III, we list the MAPE of the models -that were pre-trained on the different datasets of the meta set- on the query set datasets. We performed the experiment for 0, 1, and 5 epochs. The results indicate that the gained knowledge from the nearest task made the models faster in terms of epoch convergence. The results from the second nearest task indicate that transferring the knowledge even from the second nearest task is still a better starting point for training neural networks than training them from random weights. The average MAPE of the nearest model is 0.29 without training compared to an average of 0.95 when we train a random FCN for 5 epochs. Furthermore, we downsampled the datasets from the query set to 10%, 20%, and 50% of their original size. The downsampling simulates the data scarcity in this con-

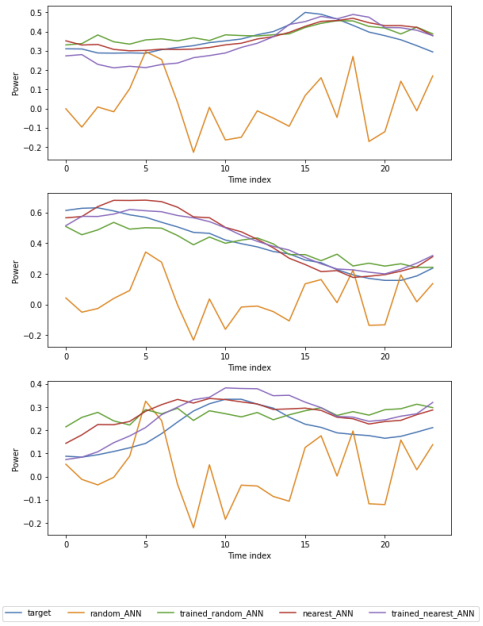

Figure 7: Forecasting sample in different learning levels.

text. Table IV shows the MAPE performance of the nearest and second nearest models when trained on different datasets. The results demonstrate that even with a reduction in data due to down-sampling, transfer learning from the nearest dataset can result in a good performance. In all cases, the nearest model performed well comparing the models trained on the whole dataset, indicating that the knowledge transferred from the nearest task overcame the data scarcity. For example, the average MAPE performance is better by a factor of 32% between the using the pertrained model and a random FCN. In figure 7, we show the performance of different networks from the experiment. The figure shows clearly how the nearest model gives a better forecast ability compared to the random network.

Dataset Name	Correlation Value
NSW03	89.10
NSW18	88.45
QLD03	86.89
QLD18	86.82
SA03	87.94

Table 5: Correlations between the TAS distance and the MAPE on the query datasets

4.2 Case Study 2: Smart Grid Data Set

The second case study is presented in the Appendix.

5 Future Work and Conclusion

Our work results suggest that the TAS metric should be given more consideration as a metric for evaluating the transferability of machine learning models in real-world applications, particularly in scenarios where efficiency and convergence speed are important considerations. This finding has important implications for the design of meta-learning algorithms and suggests that incorporating task similarity information can improve the performance of these algorithms. In particular, MAML [Finn *et al.*, 2017] was used as a part of the transferrable model-agnostic meta-learning (T-MAML) approach [He *et al.*, 2022]. This research work proposes an approach for load forecasting for single households. It enables multiple households to collaboratively train a generic artificial neural network (ANN) model and then further train the model at each target household node for the purpose of STLF. We propose that we use the TAS distance to select the subset of the dataset used in the meta-learning phase to minimize the number of learning steps instead of doing a random selection.
In this paper, we have presented empirical evidence to support the use of Task Affinity Score (TAS) as a reliable and effective distance measure for task similarity in transfer learning. Our study also introduced a new transfer learning algorithm called Affinity Driven Transfer Learning (ADTL), which leverages TAS to select the most appropriate source task for knowledge transfer. To evaluate the effectiveness of ADTL, we conducted experiments on two datasets: the AEMO dataset and the smart Apartment dataset. Our results demonstrate that ADTL outperforms traditional transfer learning approaches in terms of Mean Absolute Percentage Error (MAPE), across both datasets. These findings suggest that ADTL can successfully identify the most relevant source task for knowledge transfer, based on the task affinity score. Furthermore, we investigated the effectiveness of ADTL under data scarcity conditions. To simulate this scenario, we downsampled the datasets to 10%, 20%, and 50% of their original size. Our experiments show that even under these data-scarce conditions, ADTL continues to perform significantly better than randomly initialized models. This demonstrates the potential of ADTL to address the data scarcity problem in transfer learning and suggests that it is a promising approach for real-world applications. Overall, our study provides important insights into the use of TAS as a distance measure for task similarity in transfer learning and highlights the potential of ADTL as an effective transfer learning approach. Future research in this area could further refine the use of TAS in transfer learning, and explore additional approaches for leveraging task similarity to improve knowledge transfer.

Acknowledgment

The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC) and a start-up grant from Concordia University. The complete source code of this work is available upon request.

References

[Alhussein *et al.*, 2020] Musaed Alhussein, Khursheed Aurangzeb, and Syed Irtaza Haider. Hybrid cnn-lstm model for short-term individual household load forecasting. *Ieee Access*, 8:180544–180557, 2020.

[Cai *et al.*, 2019] Long Cai, Jie Gu, and Zhijian Jin. Two-layer transfer-learning-based architecture for short-term load forecasting. *IEEE Transactions on Industrial Informatics*, 16(3):1722–1732, 2019.

[Chen *et al.*, 2001] Hong Chen, Claudio A Canizares, and Ajit Singh. Ann-based short-term load forecasting in electricity markets. In *2001 IEEE power engineering society winter meeting. Conference proceedings (Cat. No. 01CH37194)*, volume 2, pages 411–415. IEEE, 2001.

[Christiaanse, 1971] WR Christiaanse. Short-term load forecasting using general exponential smoothing. *IEEE Transactions on Power Apparatus and Systems*, (2):900–911, 1971.

[Fang *et al.*, 2021] Xi Fang, Guangcai Gong, Guannan Li, Liang Chun, Wenqiang Li, and Pei Peng. A hybrid deep transfer learning strategy for short term cross-building energy prediction. *Energy*, 215:119208, 2021.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[Gao *et al.*, 2020] Yuan Gao, Yingjun Ruan, Chengkuan Fang, and Shuai Yin. Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data. *Energy and Buildings*, 223:110156, 2020.

[Goodfellow *et al.*, 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[Gopalakrishnan *et al.*, 2017] Kasthurirangan Gopalakrishnan, Siddhartha K Khaitan, Alok Choudhary, and Ankit Agrawal. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and building materials*, 157:322–330, 2017.

[Graves *et al.*, 2013] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013.

[Gregor *et al.*, 2015] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In *International conference on machine learning*, pages 1462–1471. PMLR, 2015.

[He *et al.*, 2022] Yu He, Fengji Luo, and Gianluca Ranzi. Transferrable model-agnostic meta-learning for short-term household load forecasting with limited training data. *IEEE Transactions on Power Systems*, 37(4):3177–3180, 2022.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Hong *et al.*, 2020] Tao Hong, Pierre Pinson, Yi Wang, Rafał Weron, Dazhi Yang, and Hamidreza Zareipour. Energy forecasting: A review and outlook. *IEEE Open Access Journal of Power and Energy*, 7:376–388, 2020.

[Hong, 2009] Wei-Chiang Hong. Electric load forecasting by support vector model. *Applied Mathematical Modelling*, 33(5):2444–2454, 2009.

[Hooshmand and Sharma, 2019] Ali Hooshmand and Ratnesh Sharma. Energy predictive models with limited data using transfer learning. In *Proceedings of the tenth ACM international conference on future energy systems*, pages 12–16, 2019.

[Hua *et al.*, 2021] Jiang Hua, Liangcai Zeng, Gongfa Li, and Zhaojie Ju. Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. *Sensors*, 21(4):1278, 2021.

[Jiang *et al.*, 2022] Ruiqi Jiang, Shaoxiong Zeng, Qing Song, and Zhou Wu. Deep-chain echo state network with explainable temporal dependence for complex building energy prediction. *IEEE Transactions on Industrial Informatics*, 19(1):426–435, 2022.

[Le *et al.*, 2020] Tuong Le, Minh Thanh Vo, Tung Kieu, Eenjun Hwang, Seungmin Rho, and Sung Wook Baik. Multiple electric energy consumption forecasting using a cluster-based strategy for transfer learning in smart building. *Sensors*, 20(9):2668, 2020.

[Le *et al.*, 2022] Cat Phuoc Le, Juncheng Dong, Mohammadreza Soltani, and Vahid Tarokh. Task affinity with maximum bipartite matching in few-shot learning. In *International Conference on Learning Representations*, 2022.

[LeCun *et al.*, 2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[Li *et al.*, 2021] Ao Li, Fu Xiao, Cheng Fan, and Maomao Hu. Development of an ann-based building energy model for information-poor buildings using transfer learning. In *Building simulation*, volume 14, pages 89–101. Springer, 2021.

[Liao *et al.*, 2021] Wenlong Liao, Zhe Yang, Xinxin Chen, and Yaqi Li. Windgmmn: Scenario forecasting for wind power using generative moment matching networks. *IEEE Transactions on Artificial Intelligence*, 3(5):843–850, 2021.

[Lu *et al.*, 1993] C-N Lu, H-T Wu, and S Vemuri. Neural network based short term load forecasting. *IEEE Transactions on Power Systems*, 8(1):336–342, 1993.

[Lv *et al.*, 2021] Lingling Lv, Zongyu Wu, Jinhua Zhang, Lei Zhang, Zhiyuan Tan, and Zhihong Tian. A vmd and lstm based hybrid model of load forecasting for power grid security. *IEEE Transactions on Industrial Informatics*, 18(9):6474–6482, 2021.

[Mansouri and Akbari, 2014] Vahid Mansouri and Mohammad E Akbari. Efficient short-term electricity load forecasting using recurrent neural networks. *Journal of Artificial Intelligence in Electrical Engineering*, 3(9):46–53, 2014.

[Mikolov *et al.*, 2010] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010.

[Niu *et al.*, 2020] Shuteng Niu, Yongxin Liu, Jian Wang, and Houbing Song. A decade survey of transfer learning (2010–2020). *IEEE Transactions on Artificial Intelligence*, 1(2):151–166, 2020.

[of Climate Change and Energy, 2014] Australian Government Department of Climate Change and Energy. Smart grid, smart city, 2014. Available online at: https://data.gov.au/dataset/ds-dga-4e21dea3-9b87-4610-94c7-15a8a77907ef/details?q=smart%20grid%20smart%20city,[Accessed 2023-02-06].

[Operator, 2014] Australian Energy Market Operator. Aggregated price and demand data, 2014. Available online at: https://aemo.com.au/energy-systems/electricity/national-electricity-market-nem/data-nem/aggregated-data.

[Pappas *et al.*, 2008] S.Sp. Pappas, L. Ekonomou, D.Ch. Karamousantas, G.E. Chatzarakis, S.K. Katsikas, and P. Liatsis. Electricity demand loads modeling using autoregressive moving average (arma) models. *Energy*, 33(9):1353–1360, 2008.

[Park *et al.*, 1991] Dong C Park, MA El-Sharkawi, RJ Marks, LE Atlas, and MJ Damborg. Electric load forecasting using an artificial neural network. *IEEE transactions on Power Systems*, 6(2):442–449, 1991.

[Peng *et al.*, 2022] Chao Peng, Yifan Tao, Zhipeng Chen, Yong Zhang, and Xiaoyan Sun. Multi-source transfer learning guided ensemble lstm for building multi-load forecasting. *Expert Systems with Applications*, 202:117194, 2022.

[Ranaweera *et al.*, 1996] DK Ranaweera, NF Hubele, and GG Karady. Fuzzy logic for short term load forecasting. *International journal of electrical power & energy systems*, 18(4):215–222, 1996.

[Song *et al.*, 2022] Yue Song, Diyin Tang, Jinsong Yu, Zetian Yu, and Xin Li. Short-term forecasting based on graph convolution networks and multiresolution convolution neural networks for wind power. *IEEE Transactions on Industrial Informatics*, 19(2):1691–1702, 2022.

[Taieb and Atiya, 2015] Souhaib Ben Taieb and Amir F Atiya. A bias and variance analysis for multistep-ahead time series forecasting. *IEEE transactions on neural networks and learning systems*, 27(1):62–76, 2015.

[Weiss *et al.*, 2016] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[Wu and Lin, 2022] Di Wu and Weixuan Lin. Efficient residential electric load forecasting via transfer learning and graph neural networks. *IEEE Transactions on Smart Grid*, 14(3):2423–2431, 2022.

[Xia *et al.*, 2010] Changhao Xia, Jian Wang, and Karen McMenemy. Short, medium and long term load forecasting model and virtual load forecaster based on radial basis function neural networks. *International Journal of Electrical Power & Energy Systems*, 32(7):743–750, 2010.

[Yu *et al.*, 2019] Zeyuan Yu, Zhewen Niu, Wenhu Tang, and Qinghua Wu. Deep learning for daily peak load forecasting–a novel gated recurrent neural network combining dynamic time warping. *IEEE access*, 7:17184–17194, 2019.

[Zhou *et al.*, 2020] Dengji Zhou, Shixi Ma, Jiarui Hao, Dong Han, Dawen Huang, Siyun Yan, and Taotao Li. An electricity load forecasting model for integrated energy system based on bigan and transfer learning. *Energy Reports*, 6:3446–3461, 2020.

[Zoph *et al.*, 2016] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*, 2016.

Appendix

5.1 Case Study 2: Smart Grid Data Set

This dataset "The Smart-Grid Smart-City Customer Trial Data" is collected as part of a trial conducted by the Australian Government Department of Climate Change, Energy, the Environment, and Water [of Climate Change and Energy, 2014]. The dataset contains electricity consumption data from around 1,300 households in New South Wales, Australia, collected over a period of 12 months. The data includes half-hourly electricity consumption readings. In the context of transfer learning for load forecasting in an electricity grid, the metaset can be seen as a set of pre-trained models that have already learned to forecast the load for some subset of elements in the grid. These models have been trained on historical data and can be thought of as "experts" in predicting the load for those specific elements. The query set, in this case, would refer to the new models that need to be trained for forecasting the load for previously unseen or new elements in the same electricity grid. These new models would need to be trained on a smaller amount of data, as compared to the pre-trained models in the metaset. The goal of transfer learning in this scenario would be to leverage the knowledge learned by the pre-trained models on the elements they have already forecasted, to improve the accuracy and efficiency of training the new models for the new elements.

The primary goal of our experiments on this dataset was to evaluate the effectiveness of our approach in transferring knowledge from a set of known source apartments to forecast the electricity consumption in a target apartment. Specifically, we sought to determine whether TAS could identify the most relevant source task for a given target task and whether this approach could speed up the transfer learning process. We select a sample of 20 random apartments as the meta set and a sample of 30 apartments as the query set. Then,

	1 day data		3 day data		7 day data	
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
Random LSTM	2.25	0.340	1.98	0.313	2.25	0.349
Random LSTM (5 epochs)	1.75	0.211	1.72	0.237	2.01	0.251
Nearest LSTM	0.81	0.201	0.77	0.199	0.85	0.208
Nearest LSTM (5 epochs)	**0.73**	**0.195**	**0.72**	**0.195**	**0.75**	**0.198**

Table 6: Performance of the selected pre-trained neural network with comparison to a random neural network

	10% downsampling		20% downsampling		50% downsampling	
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
Random LSTM	3.73	0.537	3.35	0.552	3.64	0.532
Random LSTM (5 epochs)	3.58	0.530	3.11	0.415	3.51	0.304
Nearest LSTM	0.92	0.238	0.83	0.225	0.79	0.182
Nearest LSTM (5 epochs)	**0.82**	**0.199**	**0.80**	**0.202**	**0.75**	**0.197**

Table 7: Performance of the selected pre-trained neural network and a random neural network on down-sampled datasets

we train 20 different LSTM models on the apartments' data from the metaset. This will serve as the different possible sources for the query set apartments data. To evaluate the performance of ADTL, we used the MAPE and the RMSE. Overall, our experiments on the dataset demonstrate the potential of ADTL as an effective transfer learning approach when the source and target tasks are closely related. In table VI, we present a comparison between a random LSTM and the picked LSTM using the ADTL approach. We evaluated the algorithm's performance by comparing the results of a random initialized LSTM with the results of the nearest LSTM in terms of task distance. After training for 5 epochs, the random network's performance could not be improved and remained very poor for all prediction windows and horizons (1 day, 3 days, and 7 days), with an average of 1.83 MAPE and 0.233 RMSE values. In contrast, the nearest LSTM showed significant improvement in performance after being trained for 5 epochs, achieving high accuracy for all prediction horizons, with an average of 0.73 MAPE and 0.196 RMSE values. These results highlight the limitations of traditional deep learning models in learning from new and unseen data and demonstrate the potential of our transfer learning approach to improve their performance. In Table VII, we conducted experiments with reduced data samples from each apartment to simulate data scarcity issues. We downsampled the data from the query set. We present the average performance of a randomly generated LSTM network and the nearest LSTM network in the TAS sense. Our results demonstrate that the nearest LSTM outperformed the randomly generated network. Notably, after only five epochs of training, the nearest LSTM achieved performance comparable to being trained on the complete dataset, without downsampling.