Pre-trained Models for Non-intrusive Appliance Load Monitoring

Lingxiao Wang, Shiwen Mao, *Fellow, IEEE*, Bogdan Wilamowski, *Fellow, IEEE* and R. M. Nelms, *Fellow, IEEE*

***Abstract*—Non-intrusive load monitoring (NILM) is to estimate individual appliance's power consumption from aggregated smart meter data, which is useful for optimized energy management and provisioning of customized services. While deep learning (DL) has achieved state-of-the-art NILM performance, it is still constrained by the dependency on large amounts of data and intensive computations on training. In this paper, we propose a pre-training approach to address the generalization of DL models for NILM. We develop a meta-learning based approach and an ensemble learning based approach, which pre-train a base model and then fine-tune it with few-short learning when applied to an unknown dataset. The models are validated with two real-world datasets and shown to achieve a superior transferability performance compared with traditional DL and transfer learning methods.**

***Index Terms*—Non-intrusive load monitoring, ensemble learning, meta learning, transfer learning, pre-trained model.**

I. Introduction

With extremely low latency, high data rate, and significant improvement of quality of service (QoS), the 5G and beyond wireless networks offer considerable benefits in many fields. However, the tremendous energy usage, estimated to be 10 times more than the existing 4G networks, has raised great concerns [1]. Inspired by the advances in green communications and networking (GCN), which aims to Send More Information bits with Less Energy (SMILE), energy-efficient techniques, such as BS switching [2]–[4], offline power allocation and online data scheduling, as well as sustainable energy powered base stations (BSs) have been developed to reduce the energy usage and boost network capacity [5]. GCN has a close interaction with the power grid. On one hand, for retailers, communication networks collect data and information from the power grid components, which can be analyzed and used to control the power system for real-time pricing, demand response, and protection [6]. On the other hand, for consumers, networks construct communication paths that integrate smart meters, home appliances, and renewable energy sources for Home Energy Management Systems (HEMS) [7], [8].

Among the HEMS applications, Non-Intrusive Load Monitoring (NILM) has been recognized as an essential component. The goal is to estimate each individual appliance's power consumption from the aggregated smart meter data; it is non-intrusive since only the aggregated power consumption is needed [9], [10]. The most crucial advantage of NILM is its nonintrusive character. Comparing to intrusive approaches, NILM can provide appliances energy usage information without sub-meter installation, which is expensive, hard to upgrade, and causes data privacy concerns [11]. With the NILM results, homeowners can enjoy the benefits of optimizing energy assets to achieve energy savings. It was reported in [12] that feedback on power usage stimulates energy savings ranging from 1.1% to over 20%. As residential consumers have increasingly adopted more electric vehicles (EVs) and home solar systems, instant information about their energy consumption and generation will help to optimize their energy utilization. Another benefit for consumers is equipment malefaction detection. NILM provides feedback when an appliance, e.g., air-conditioner or refrigerator, consumes more energy than expected with anomaly detection algorithms without the need for sub-meter level data. For retailers, NILM can also help to improve their energy management (power system scheduling and planning). Provided with customer's consumption behavior from NILM, retailers can provide customized services, such as offering energy-saving tips (i.e., informing consumers to lower power consumption when the wholesale market prices are high) and enabling different billing methods (static or dynamic), to improve customer satisfaction [13].

Although NILM brings about great benefits, it faces many challenges as well. The most successful approach to NILM, so far, is deep learning (DL), which achieves the state-of-the-art performance. However, it requires a large amount of labeled data to train the DL model. For NILM, this requires electrical submetering in the houses, which is to use additional electricity monitors to record the usage of individual appliances in the house, and thus incurs additional costs. Furthermore, as people are more concerned about their privacy, the active power data used for training the NILM model is hard to obtain. Moreover, as most data-driven models, the DL approach requires extensive computation. It would be desirable to eliminate the need to train a model every time it is used for a new house. Therefore, for practical deployment of NIML solutions, it is critical to develop DL models that are *generalizable*, such that we can train the model with data collected from a small number of submetered houses, and then easily apply the trained model to other houses without submetering. Such generalizable DL models are also useful to deal with houses with different appliances, different residents' usage behavior, and various aging degrees of circuits [14].

In this paper, we investigate the problem of pre-trained DL

This work is supported in part by the US National Science Foundation under Grant DMS-1736470 and by the Wireless Engineering Research and Education Center at Auburn University, Auburn, AL, USA.

The authors are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA. B. Wilamowski is also with The University of Information Technology and Management, Rzeszow 35-225, Poland. Email: lzw0039@auburn.edu, smao@ieee.org, wilambm@auburn.edu, nelmsrm@auburn.edu.

DOI: 10.1109/TGCN.2021.3087702

models for NILM, which is a promising means to address the above problems. With this approach, a base DL model is first trained with a larger dataset. When applied to a new environment, the base model is first fine-tuned with a small amount of new data, and then the fine-tuned model is used for inference in the new environment. On one hand, we do not need to train a new model from scratch for an unknown house. On the other hand, pre-trained models are able to quickly adapt to new tasks with few-shot learning, as pre-trained parameters outperform random initialization for deep neural networks [15]. Therefore, such models can not only save considerable computation in training and reduce the dependency on large amounts of data, but also achieve excellent performance in real-time when it is allowed to use new data to update the parameters.

In light of these, we propose a model-agnostic meta-learning (MAML) based approach and an ensemble learning based approach for the NILM problem in this paper. Our approaches are inspired by two of the most successful Natural Language Processing (NLP) pre-trained transformer models BERT [16] and GPT-3 [17]. Ensemble learning (BERT) and meta-learning (GPT-3) are the two effective solutions toward improving the pre-training language model's adaptability. We propose these two approaches to deal with the transferability of the NILM problem. Both methods obtain the pre-trained models using one dataset, and then fine-tunes their parameters using a small amount of new data when applied for inference with another dataset. To the best of our knowledge, this is the first work that applies meta-learning and ensemble learning for generalizable models to the NILM problem. We develop both models and evaluate their performance with two real-world datasets, using one dataset to pre-train the models and the other dataset to fine-tune the models and test their generalization performance. Our experiments validate the superior transferability of the proposed models for the NILM problem, which both outperform the state-of-the-art DL based approach and the transfer learning based approach [18]. We also find that the proposed models are effective in overcoming the negative transfer problem. The proposed models require greatly reduced amount of data and computation for real-world deployment, which lead to energy savings and is inline with the goals of GCN.

We organize the remainder of this paper as fellows. Related work is introduced in Section II. In Section III, we formulate the NILM problem and introduces several solution approaches. In Section IV, we present the two proposed methods. We present the datasets and experiment setup in Section V, and our experimental validation of the proposed models in Section VI. Section VII concludes this paper.

II. Related Works

A. The NILM Problem and Existing Solutions

The wide deployment of smart meters has triggered great interest in NILM, which is to estimate the power consumption of a target appliance from the aggregate meter readings of the entire house. Many algorithms have been developed to address the NILM problem. For example, the Additive Factorial Hidden Markov Model (AFHMM) and its variants have been used in many existing schemes [19]–[23]. The Graph Signal Processing (GSP) based method has also been shown to be quite effective [24], [25]. Other traditional machine learning approaches, such as Support Vector Machine (SVM) [26], Decision Trees [27], the hybrid classification method [28], k-nearest neighbors (k-NN) [29], and so forth, have been applied to solve the NILM problem as well. Interested readers are referred to the detailed reviews in [11], [30]. Note that such works only focus on training and inference with the same dataset, rather than the generalization problem.

Motivated by the success of deep learning in other fields, there has been great interest in applying deep learning to solve the NILM problem [31]. Convolutional Neural Networks (CNNs) models have been adopted in [32]–[34] to extract the temporal features from time series of aggregate electricity consumption data. In [35], Long Short-Term Memory (LSTM) or its equivalent Gated Recurrent Units (GRUs) models have been leveraged to capture the long and short-term patterns of state signatures of different appliances, which belong to the class of Recurrent Neural Networks (RNNs). De-noising auto-encoder has also been applied for noise reduction to better estimate the appliance profile [36].

B. Pre-trained Models

Recent work has shown that by pre-training a deep neural network on a large corpus of data, followed by fine-tuning when applied to a specific task, the model's performance on the target task can be effectively improved. This approach has been successfully applied in computer vision, speech recognition, and especially in NLP.

A pre-trained hidden Markov model for large-vocabulary speech recognition was proposed in [15]. The authors showed that the pre-trained model was robust and achieved good initialization of weights when training deep neural networks. For computer vision, the authors in [37], [38] explored image feature transferability of CNNs and found that the pre-trained model could boost the generalization performance to new image classification tasks. Recently, pre-trained models have drawn considerable attention in NLP. For example, ELMo (Embeddings from Language Models) [39] is a feature-based NLP pre-training approach, which combines individual feature extract LSTMs to improve the overall task performance. The pre-trained transformer language model BERT [16] can effectively handle multiple NLP tasks, after being fine-tuned directly without the need for task-specific architectures. In 2020, OpenAI launched GPT-3, a gigantic deep neural network with 175 billion parameters [17], to tackle task-agnostic NLP problems without needing any gradient updates or fine-tuning. Motivated by the success of pre-trained models in other fields, we investigate how to apply it to solve the NILM problem in this paper.

C. Pre-trained Models for NILM

There has been very few existing works on pre-trained models for NILM. Most of the prior works trained and tested their models using the dataset from the same house,

by partitioning the same dataset into a training set and a testing set. The generalization performance of the models has not been verified. In [36], [40], the authors considered transferability across houses included the same dataset, i.e., testing a model trained by one house and on an untrained house, which both belonged to the same dataset. In [14], [18], houses from different datasets were used, where a model was pre-trained on a large dataset and then its transferability and generalization performance was verified through another domain. The difference between them was that the work [14] tested its model without any parameter updating, which is known as *zero-shot learning*. Usually transferring a model between two different datasets could lead to poor performance. In [18], the pre-trained model was fine-tuned by using data from the other dataset (i.e., few-shot fine-tuning). The limitation of [18] was that the fine-tuned model's performance was sometimes worse than the zero-shot models. This was because the data used in fine-tuning was quite different from that of the tested house, which led to *negative transfer*. The generative adversarial networks (GANs) are used as the pre-trained model in [41], [42]. By minimizing the statistical distance between source and target domains in the feature space, the authors in [41] overcame the drawback that the shared parameters of the pre-trained model are sensitive to the similarity between different domains. In [42], the joint adaptation loss was further introduced by adapting both the feature and the label distribution discrepancy, which improved the performance of GANs.

Another approach of using pre-trained models for NILM is to train a model on visual recognition tasks and the transfer the image feature extractor to the appliance recognition task. To bridge these two unrelated domains, i.e., computer vision and NILM, the authors in [43] introduced the concept of a load signature, i.e., the voltage-current (V-I) trajectory, to enable transfer learning. Since the features extracted from the NILM data is usually quite different from real-world images, it is challenging to verify the model's robustness to domain shits (i.e., from real images to power consumption data).

III. Problem Statement and Approaches

In this section, we first present the mathematical formulation for the Non-Intrusive Load Monitoring (NILM) problem. We will then introduce the conventional supervised machine learning (ML) and pre-training approaches (i.e., transfer learning and meta-learning) to solve the problem.

A. The NILM Problem

Consider a house that contains J appliances that consume electricity. The aggregated power consumption of the house, as measured by a smart meter, is given by

$$x(t) = \sum_{j=1}^{J} y_j(t) + e(t), \tag{1}$$

where $x(t)$ is the aggregated power consumption, $y_j(t)$ is the jth appliance's power consumption, and $e(t)$ is the measurement noise at time t. Given measurement of the total power consumption over a time period T, i.e., $\tilde{\mathbf{x}} = (x(1), x(2), ..., x(T))$, the goal of NILM is to estimate the individual appliance's power consumption trace for the same period T, i.e., $\tilde{\mathbf{y}}_j = (y_j(1), y_j(2), ..., y_j(T))$, for $j = 1, 2, ..., J$.

Supervised ML has been applied to solve the NILM problem, as reviewed in Section II, which is to train a model with observed pairs of $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}_j)$ (i.e., the labeled training set) to estimate (i.e., learn) an approximate function $f_{\boldsymbol{\theta}}(\cdot)$ over a parameter set $\boldsymbol{\theta}$ with a learning algorithm, which represents the relationship between $\mathbf{y}_j$ and $\mathbf{x}$ by

$$\mathbf{y}_j = f_{\boldsymbol{\theta}}(\mathbf{x}). \tag{2}$$

Various learning models can be applied to solve the NILM problem. For example, the conventional ML approach utilizes a single learning algorithm to learn the function $f_{\boldsymbol{\theta}}(\cdot)$. On the other hand, transfer learning leverages a base learner to learn the function, and then utilizes new data to adapt to a new domain. Meta-learning, known as *learning to learn*, incorporates several learning episodes to induce the learning algorithm itself. In the remainder of this section, we describe how to solve the NILM problem with these ML approaches from an optimization perspective. We will use two separate load monitoring datasets, i.e., a source dataset $\mathcal{S}$ and a target dataset $\mathcal{T}$, in the following discussions. Both datasets contain the aggregated power consumption data as well as the consumption data of individual appliances (as labels).

B. Conventional Machine Learning Approach

To solve the NILM problem with a conventional ML approach, only one dataset $\mathcal{S}$ is used. This dataset is split into two parts, i.e., a training set $\mathcal{S}^{tr}$ and a testing set $\mathcal{S}^{ts}$. The ML model is trained with the training set $\mathcal{S}^{tr}$ to determine its parameters $\boldsymbol{\theta}$. The trained model is then tested on the separate testing set $\mathcal{S}^{ts}$.

The goal of the training process is to minimize a loss function $\mathcal{L}$, give by

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \ \mathcal{L}\left(\boldsymbol{\theta}, \mathcal{S}^{tr}\right). \tag{3}$$

The model parameters $\boldsymbol{\theta}$ are usually updated with the gradient descent (GD) method as follows.

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{\theta}, \mathcal{S}^{tr}\right), \tag{4}$$

where η is the learning step size, and $\nabla_{\boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{\theta}, \mathcal{S}^{tr}\right)$ is the gradient of the loss function with respect to $\boldsymbol{\theta}$. When the model is well trained, its performance will be evaluated using the testing set $\mathcal{S}^{ts}$. Such a process is illustrated by the graphical model given in Fig. 1.

The conventional supervised ML approach to the NILM problem usually focuses only on a single dataset $\mathcal{S}$. The model parameters are optimized with respect to this dataset. Usually the trained model does not generalize well to an untrained dataset $\mathcal{T}$ (i.e., a new domain).

C. Transfer Learning Approach

In transfer learning, both a source dataset $\mathcal{S}$ and a target dataset $\mathcal{T}$ are used. The goal is to adapt the pre-trained model learned from the source dataset $\mathcal{S}$ to the target dataset $\mathcal{T}$.

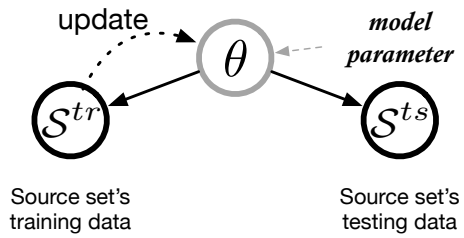


Fig. 1: Conventional supervised ML approach: the model parameters are updated based on a single task training dataset $\mathcal{S}^{tr}$ and tested on a testing set $\mathcal{S}^{ts}$.

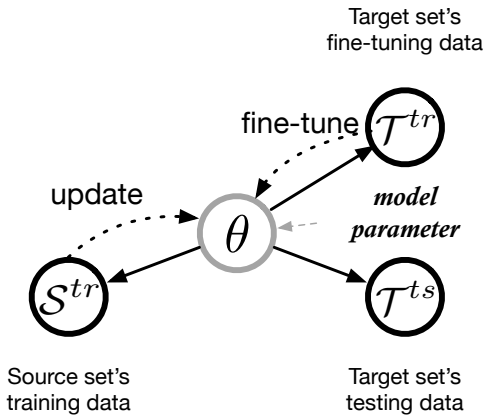


Fig. 2: Transfer learning approach: the model parameters are updated based on the training set $\mathcal{S}^{tr}$, fine-tuned on set $\mathcal{T}^{tr}$, and tested on set $\mathcal{T}^{ts}$.

The procedure is illustrated in Fig. 2. First, we create the training dataset $\mathcal{S}^{tr}$ from $\mathcal{S}$ to pre-train the model. The pre-training problem can be defined as

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \ \mathcal{L}\left(\boldsymbol{\theta}, \mathcal{S}^{tr}\right). \tag{5}$$

With the gradient descent (GD) method, the model parameters $\boldsymbol{\theta}$ are updated using the training set $\mathcal{S}^{tr}$ as:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{\theta}, \mathcal{S}^{tr}\right). \tag{6}$$

In the testing phase, the target dataset $\mathcal{T}$ is split into a fine-tuning set $\mathcal{T}^{tr}$ and a testing set $\mathcal{T}^{ts}$. We fine-tune the pre-trained model using the fine-tuning set $\mathcal{T}^{tr}$ for purpose of *domain adaptation*, where the parameters are updated as

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{\theta}, \mathcal{T}^{tr}\right). \tag{7}$$

Then the fine-tuned model is tested on the testing set $\mathcal{T}^{ts}$. During the fine-tuning stage in transfer learning, most existing works freeze the parameters in most of the layers, except the last fully-connected layer. The fine-tuned model is obtained by training the parameters of the last fully-connected layer using the new task's data $\mathcal{T}^{tr}$.

In our prior work [8], we developed an ensemble learning model for load forecasting in urban power systems, which includes multiple long short-term memory (LSTM) based first-level learners and a Fully Connected Cascade (FCC) neural network as the second-level learner. In this paper, we propose an ensemble learning based transfer learning approach to solve the NILM problem. The proposed model will be presented in Section IV.

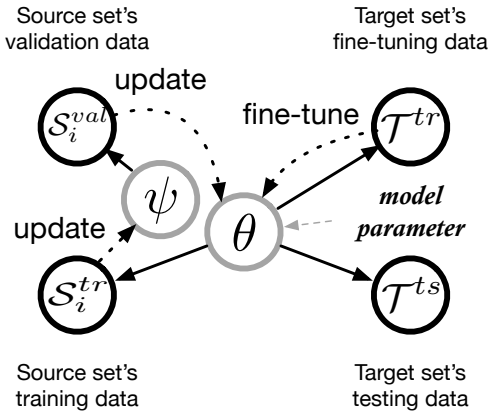


Fig. 3: Meta learning approach: The model parameters are updated based on two meta training datasets $\mathcal{S}^{tr}$ and $\mathcal{S}^{val}$, fine-tuned on set $\mathcal{T}^{tr}$, and tested on set $\mathcal{T}^{ts}$.

D. Meta-learning Approach

Meta-learning, a.k.a. learning to learn, is inspired by human's quickly learning new things with only a few examples. By applying automatic learning algorithms to metadata, it induces the learning algorithm itself. The goal is to enable an intelligent agent (i.e., model) learn and adapt quickly from few-shot of examples, and is able to keep adjusting as more data are coming in [44].

In general, meta-learning can be seen as training a general model that can generalize across different tasks or datasets. Here, we define a single task or dataset as $\mathcal{S}_i$, which is sampled from the source set $\mathcal{S}$. We sample the source set $\mathcal{S}$ for N times to obtain N tasks. Each task $\mathcal{S}_i$ is split into a training set $\mathcal{S}_i^{tr}$ and a validation set $\mathcal{S}_i^{val}$. In meta-training, the model (meta-learner) shares the parameters $\boldsymbol{\theta}$, which will be updated with each task's loss. The average of parameters optimized by each task (i.e., the base learners represented by parameters $\boldsymbol{\psi}$) will update the meta-learner at last. This way, the meta-learner will fit all tasks at the same time, akin to cross-validation. The target set $\mathcal{T}$ will be partitioned into a fine-tuning set $\mathcal{T}^{tr}$ and a testing set $\mathcal{T}^{ts}$. The pre-trained meta-learner will be fine-tuned on $\mathcal{T}^{tr}$ and tested on $\mathcal{T}^{ts}$.

In this paper, we will adopt Model-Agnostic Meta-Learning (MAML) [44], which is an optimization scheme, to solve the NILM problem. Detailed implementation of the proposed model will be described in the next section.

IV. Proposed Approaches

In this section, we present two approaches to the NILM problem, focusing on the generalization of the models. The first model is a meta-learning based approach, i.e., MAML, that relies on fine-tuning. The second model, termed Ensemble, is based on ensemble learning and is a feature-based approach. Both models adopt the sequence-to-point (s2p) methodology [32], which will be explained in the following.

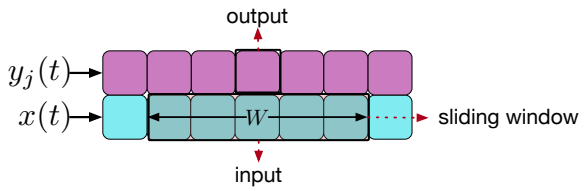


Fig. 4: One training sample instance consisting of the aggregated power consumption and appliance j's power consumption data. The sliding window size is $W = 5$ in this example.

A. Sequence-to-point Method

Traditional NILM solutions are sequence-to-sequence (seq2seq) learning, where a machine learning model maps an input sequence (i.e., the aggregate power consumption time series) to an output sequence (i.e., the power consumption time series of the target appliance). This method does not work well for NILM, where the extremely long sequences requires more memory and may cause the vanishing gradient problem in the training process. Although using a sliding window of size W could help to address the limitations, each $y_i(t)$ will be predicted W times, leading to larger errors at the edge [32].

Both our proposed methods utilize the s2p methodology instead [32]. S2p is motivated by the observation that an appliance's state at the center of the window is related to the aggregated power consumption samples before and after that point [18]. Therefore, a better prediction can be obtained for the center of the window using a full window of input data. In the example shown in Fig. 4, one training sample instance's input consists of the aggregate power consumption samples in a sliding window of size W. The learning model uses this window of input to predict the appliance's consumption at the midpoint of the window. In [45], the authors found that the s2p model outperformed 11 other power consumption disaggregation algorithms.

The s2p architecture used in this paper is shown in Fig. 5(a), which consists of five convolutional layers followed by several dense layers. We also incorporate the dropout technique to deal with the overfitting problem [46]. Mean-square Error (MSE) is used as the default loss function for training the model.

B. MAML-based Approach

The first proposed solution is a fine-tuning method that is based on Model-Agnostic Meta-Learning (MAML), which is illustrated in the upper part of Fig. 6. First, the pre-training set is sampled to obtain meta-learning's training and validation sets, which are used to pre-train the base learner (i.e., the s2p model given in Fig. 5(a)). When applied to a new dataset, a small new fine-tuning set will be used to fine-tune the pre-trained model to achieve good transferability.

Gradient-based meta-learning is regarded as an effective approach for few-shot learning. MAML is most widely used to adapt pre-trained models to new tasks by only using a few samples. It aims to find a good initialization of model parameters suitable for varying tasks (i.e., different datasets). For few-shot learning problems, only a small amount of data is fed into a pre-trained model for several gradient updates in the fine-tuning phase. In meta-training, MAML introduces two loops of training (i.e., the inner and outer loops). In the inner-loop, a base learner is trained with $\mathcal{S}_i^{tr}$ by a base learning algorithm. In the outer-loop, a meta-algorithm updates the base learning algorithm to improve the model learned by the inner loop when dealing with new data $\mathcal{S}_i^{val}$, indicating the generalization performance of a model [47]. This is shown in the graphical model in Fig. 3 for task i. In the pre-training stage (i.e., meta-training), the base learner's parameters $\boldsymbol{\psi}$ are first initialized by $\boldsymbol{\theta}$, which is trained with the training set $\mathcal{S}_i^{tr}$ (in the inner-loop). The validation set $\mathcal{S}_i^{val}$ will be used to update the meta learner's parameters $\boldsymbol{\theta}$ (in the outer-loop). In the meta-testing stage, the pre-trained model is updated with additional gradient update steps using new data $\mathcal{T}^{tr}$ (i.e., fine-tuning). Instead of freezing the parameters of some layers, all the parameters $\boldsymbol{\theta}$ of the pre-trained model will be updated in the fine-tuning procedure. Finally, the well-trained model will be used for inference with new data $\mathcal{T}^{ts}$.

Formally, the problem that MAML solves in the meat-training stage is defined as follows

$$\min_{\boldsymbol{\theta}} \sum_{\text{Task } i} \mathcal{L}\left(\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{\theta}, \mathcal{S}_i^{\text{tr}}\right), \mathcal{S}_i^{\text{ts}}\right), \tag{8}$$

where $\boldsymbol{\theta}$ are the initialized parameters of the meta-leaner and base-leaner. The loss function of the inner-loop is defined as

$$\mathcal{L}\left(\boldsymbol{\theta}, \mathcal{S}_i^{tr}\right) = \mathbb{E}\left[\sum_{\mathbf{x},\mathbf{y}\sim\mathcal{S}_i^{tr}} \|f_{\boldsymbol{\theta}}(\mathbf{x}) - \mathbf{y}\|_2^2\right], \tag{9}$$

where $f_{\boldsymbol{\theta}}(\cdot)$ represents the inference model. In the inner-loop of the meta-training procedure, the base-learner's parameters are updated by

$$\boldsymbol{\psi}_i = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{\theta}, \mathcal{S}_i^{\text{tr}}\right), \tag{10}$$

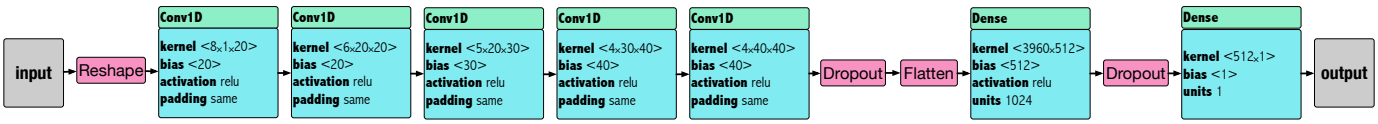
where α is the inner-loop's learning rate. In the outer-loop, the loss function is defined as

$$\mathcal{L}\left(\boldsymbol{\psi}_i, \mathcal{S}_i^{val}\right) = \mathbb{E}\left[\sum_{\mathbf{x},\mathbf{y}\sim\mathcal{S}_i^{tval}} \|f_{\boldsymbol{\psi}_i}(\mathbf{x}) - \mathbf{y}\|_2^2\right]. \tag{11}$$

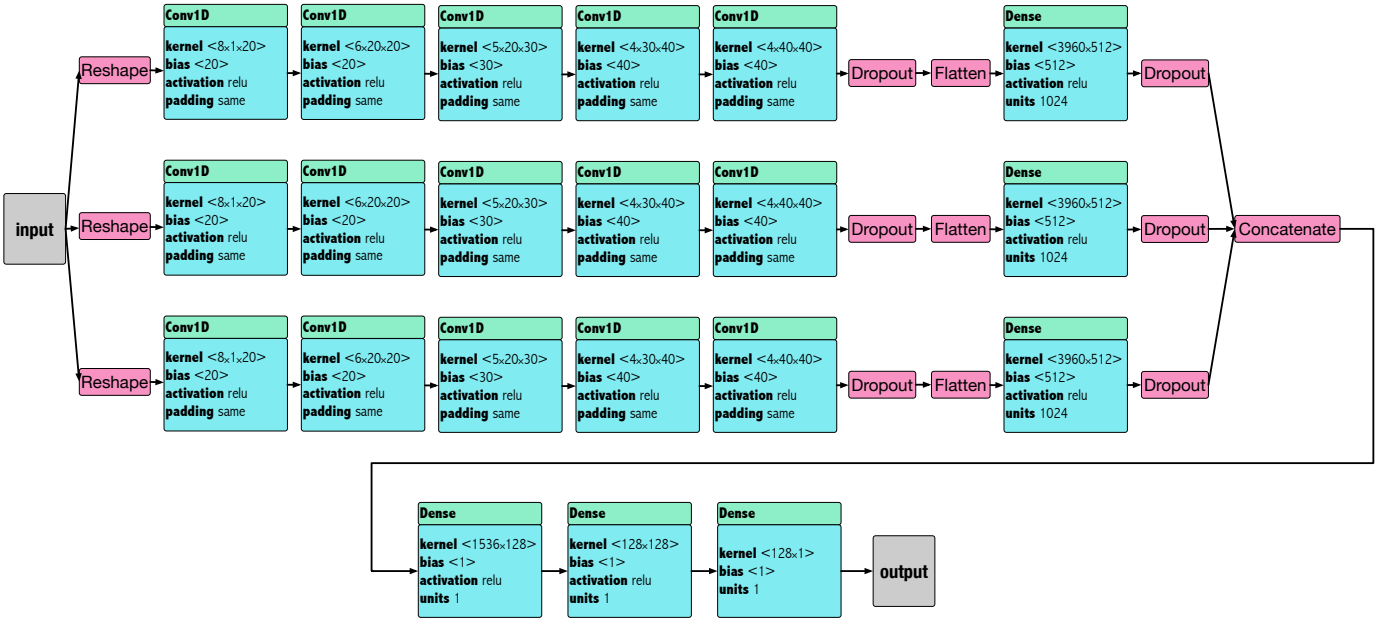
MAML solves problem (8) by using stochastic gradient descent (SGD), which involves a gradient through a gradient (i.e., need to compute the Hessian matrix). To speed-up the training process, we do not calculate the Hessian matrix, but use its first-order approximation (i.e., the Jacobian matrix). The simplified MAML algorithm is presented in Algorithm 1.

C. Ensemble Learning based Approach

Our feature-based approach is motivated by ensemble learning [48], which aims to tackle the challenge of generalization i.e., to boost the performance of the pre-trained model on any unknown dataset. Ensemble methods, i.e., stacking, have been shown to be effective for time series forecasting problems [8]. Usually, data is partitioned by a clustering algorithm, and each cluster is used to train a first-level learner. Then another neural network is used as a second-level learner to fuse the outcomes from the first-level learners for improved forecasting results.



(a) The s2p architecture used in the MAML based model.



(b) The architecture of the ensemble learning feature based pre-training model.

Fig. 5: The architecture of the proposed pre-training neural network models.

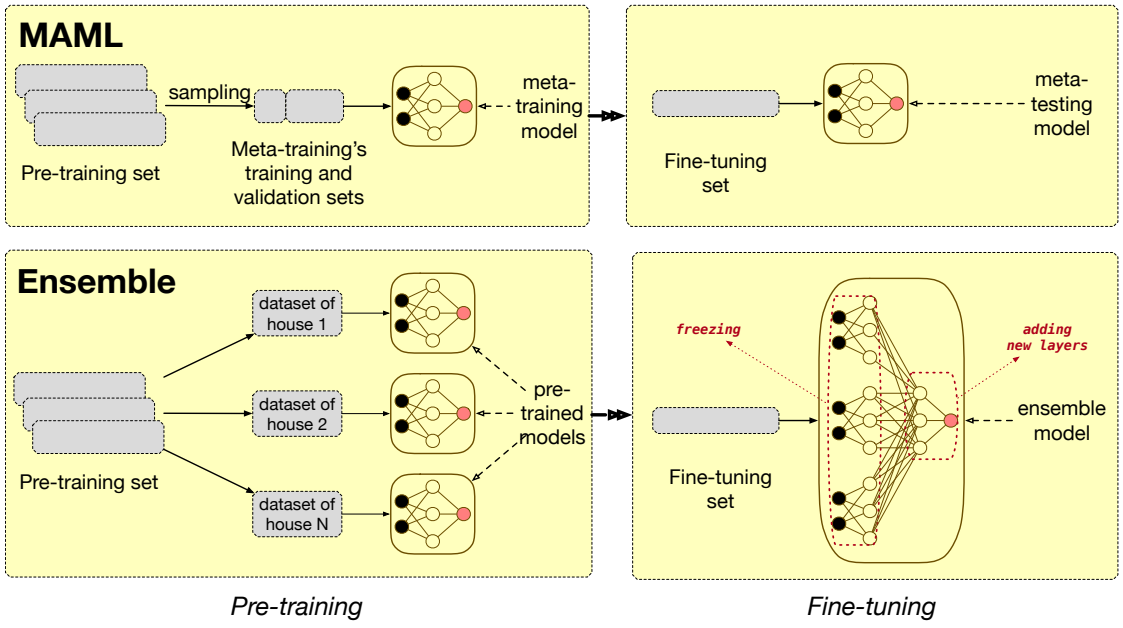


Fig. 6: The proposed methods: (i) Upper: the MAML-based approach; (ii) Lower: the ensemble learning-based approach.

The architecture of the proposed ensemble learning based model, termed Ensemble, is illustrated in the lower part of Fig. 6. Since the data from each house naturally form a cluster, the clustering algorithm is not needed here. The data from each house is used to train a first-level learner (i.e., a pre-trained model). As in the MAML based approach, a similar architecture of five convolutional layers followed by dense layers is adopted for the pre-trained models, as shown in Fig. 5(b). Similarly, dropout is incorporated to mitigate over-fitting [46]. The ensemble model then integrates the outcomes (except for the last layer) from the pre-trained learners with a concatenate module followed by several dense layers to provide the final prediction. The fusion process in fine-tuning is to select a proper combination of the feature extractors

Algorithm 1: First-order MAML [44]

Require α: inner-loop step size;
Require β: outer-loop step size;
Require A: inner-loop epochs;
Require B: outer-loop epochs;
Require Dataset $\mathcal{S}$;
for $a = 1 : B$ **do**
  Sample the source set $\mathcal{S}$ for N times to obtain $\left\{\mathcal{S}_i^{tr}, \mathcal{S}_i^{val}\right\}_{i=1}^{N}$;
  for $i = 1 : N$ **do**
    for $a = 1 : A$ **do**
      Evaluate $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathcal{S}_i^{tr})$;
      Implement gradient descent: $\boldsymbol{\psi}_i = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathcal{S}_i^{tr})$;
    end
    Calculate gradient: $\nabla_{\boldsymbol{\psi}} \mathcal{L}(\boldsymbol{\psi}_i, \mathcal{S}_i^{val})$;
  end
  Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \sum_{i=1}^{N} \nabla_{\boldsymbol{\psi}} \mathcal{L}(\boldsymbol{\psi}_i, \mathcal{S}_i^{val})$;
end

(pre-trained learners) to deal with unknown data. Due to the diversity of the feature extractors, each for a suitable case, as well as a well-designed fusion model, the ensemble model is suitable for adapting the pre-trained models to unknown datasets.

As shown in the lower part of Fig. 6, our ensemble model has two phases of training, i.e., pre-training and fine-tuning. In the pre-training phase, we split the original pre-training set $\mathcal{S}$ into several subsets, each consisting of the data from a different house. Each subset will be used to train an s2p model. In the fine-tuning phase, we first freeze each base learner's parameters. Then we concatenate all the parameters from every base model except the last dense layer as feature extractors. Three layers of a fully-connected deep neural network is then used to combine these feature extractors (i.e., pre-trained learners), as shown in Fig. 5(b). The parameters of the dense layers are trained with the fine-tuning set $\mathcal{T}^{tr}$.

V. Dataset and Experiment Setup

We evaluate the performance of the two proposed methods with extensive experiments using open-source NILM datasets. They are compared with several baseline schemes, e.g., traditional transfer learning, to validate their advantages. The datasets used in the evaluation and the experiment configurations are presented in this section.

A. Datasets

We use two real-world datasets, REFIT [49] and UK-DALE [50], to evaluate the performance of the proposed energy disaggregation methods. Both datasets are from England and provide house-level aggregate energy consumption and individual appliances' power consumption data measured by sensors deployed in the houses, while the households were conducting their usual domestic activities when the data was collected. The features of the two datasets are summarized in Tables I and II, respectively.

TABLE I: Appliances and Houses in the REFIT Dataset

Meta-training dataset (pre-training): the REFIT dataset [49]			
	Training and validation dataset		
Appliances	Houses	Time period	Samples (M)
Kettle	9, 12, 20	2013-12-07 to 2015-07-08	17.20
Microwave	10, 12, 17, 19	2013-11-20 to 2015-06-30	29.80
Washing Machine	2, 7, 9, 16, 17	2013-09-17 to 2015-07-08	19.92
Dish Washer	7, 9, 13, 16	2013-09-26 to 2015-07-08	23.38
Fridge	2, 5, 9, 12	2013-09-17 to 2015-07-08	31.33

TABLE II: Appliances and Houses in the UK-DALE Dataset

Meta-testing dataset: the UKDALE dataset [50]			
	Training (fine-tuning) dataset		
Appliances	House	Time period	Samples (M)
Kettle, Microwave, Fridge, Dish Washer, Washing Machine	2	2013-5-20 to 2013-5-29	0.108
	Testing dataset		
Appliances	House	Time period	Samples (M)
Kettle, Microwave, Fridge, Dish Washer, Washing Machine	2	2013-5-30 to 2013-10-10	1.592
	Validation dataset		
Appliances	House	Time period	Samples (M)
Kettle, Microwave, Fridge, Dish Washer, Washing Machine	1	2012-11-9 to 2012-11-18	0.102

In particular, the REFIT dataset consists of data from 21 houses, while the UK-DALE dataset has data from five houses. The data in the REFIT dataset was recorded every 8 seconds, to mimic the data collected by the SMETS2 smart meter standard2 [49]. Each house was equipped with nine appliance monitors and one current transformer sensor. The time duration of the REFIT dataset was from September 2013 to July 2015. We use a cleansed version of the REFIT dataset, where the missing values in each house (i.e., the NaN values) have been either zeroed or forward filled. In the UK-DALE dataset, each house's aggregated power consumption was recorded every 1 or 6 seconds, and each individual appliance was measured every 6 seconds. The 6-second dataset is used in our experiment. It should be noticed that the UK-DALE dataset has been preprocessed; but we use the original dataset as it is. In order to be consistent with the data in REFIT, the UK-DALE data are down-sampled to 8 seconds.

We apply standard score normalization in data before all the models are trained and tested. The value of each appliance's mean and standard deviation can be found in [18]. In our experiments, the REFIT dataset is used for pre-training, while the UK-DALE dataset is used for fine-tuning and testing, to test the models' generalization performance.

B. Hyper-parameters and Neural Network Training

Detailed information of the hyper-parameters of the proposed models are summarized in Table III. All the models are

TABLE III: Hyper-parameters of the Proposed Models

MAML	
Window size	99
Batch size	2000
SGD (inner-loop step size)	0.001
Adam (outer-loop step size)	0.001
Meta-training inner-loop epochs	1
Max meta-training outer-loop epochs	50
Max meta-testing's training epochs	10
Ensemble	
Window size	99
Batch size	2000
Adam	0.001
Maximum pre-training epochs	50
Maximum fine-tuning epochs	10

implemented with TensorFlow 2.2.0 and trained on NVIDIA RTX 2070 Mobile with the Ubuntu 18.04 operating system. The window size W is set to be 99, 299, and 499. We find that the difference in performance between the different window sizes is small. So we select the smallest value W for computational efficiency. For the meta-learning model (i.e., MAML), the inner-loop of meta-training has a step size $\alpha = 0.001$ using the SGD optimizer [51]. We implement one gradient update in the inner-loop. The outer-loop is solved using the Adam optimizer [52], which is implemented with 50 gradient updates. During meta-testing, all layers of the trained model are fine-tuned with the Adam optimizer with ten gradient updates. For the Ensemble model, the Adam optimizer is used for all the pre-trained base models.

VI. Experiment Results and Discussions

A. Experiment Methodology

In this section, we evaluate the two proposed approaches and compare them with several baseline power disaggregation algorithms. We choose five appliances in the experiments, including kettle, fridge, washing machine, dishwasher, and microwave. Each model for the appliances is trained individually, which means, for every appliance, a distinct dataset (con-training both meta-training and meta-testing sets) is constructed and used.

As mentioned in Section III, to test the transferability of the models, two different datasets are used for pre-training and fine-tuning, respectively. In our experiments, we use REFIT as pre-training dataset. This is because REFIT is a relatively large dataset, which is expected to be able to equip the trained model with better generalization ability. UK-DALE is used as the testing dataset, where the house 2 data is used to fine-tune and test the model, and the house 1 data is used as the validation set. The detailed dataset split information is provided in Table I and Table II.

Two stages of learning are conducted. Take the fridge's model as an example. For MAML, during the meta-training process, data of houses 2, 5, 9, and 12 in REFIT are first used to pre-train the model (as shown in Fig. 5(a)). The *zero-shot* results are obtained by directly applying the pre-trained model for inference for house 2 in UK-DALE. The *few-shot* results are obtained by using a few house 2 data in UK-DALE to fine-tune the model and then using the fine-tuned model for inference for house 2 in UK-DALE. For Ensemble (i.e., the feature-based pre-train method), data of houses 2, 5, 9, and 12 in REFIT are used to pre-train multiple base models (as shown in Fig. 5(b)). During the fine-tuning process, we will first examine each model's performance without any parameter updates using the meta-testing's test data to obtain the zero-shot results. The best pre-trained model, which scores the highest performance on house 2 in UK-DALE, will be fine-tuned with new data in the same way as MAML to obtain the few-short results.

The following three baseline schemes are used in our comparison study:

- Sequence-to-point (s2p): this is the model shown in Fig. 5(a) that is trained from scratch using only meta-testing's fine-tuning dataset (see Table II). This is regarded as the bottom-line benchmark.
- Transfer learning for NILM (TL) [18]: this is the traditional transfer learning approach that uses s2p as the base model. It is trained with REFIT and tested on UK-DALE, with and without fine-tuning.
- Pre-trained sequence-to-point (pre-s2p) model uses the REFIT dataset to train the base model, which is similar to TL [18]. The difference between pre-s2p and TL [18] is that data from different houses is used to build several models in pre-s2p, while TL [18] utilized the entire dataset to build only one model. We only choose the base model with the best zero-shot MAE performance for fine-tuning.

Note that the authors in [18], [45] compared the s2p scheme with other traditional machine leaning methods, and found s2p achieved the best performance. Therefore, we choose s2p as a baseline scheme in this section.

Two performance metrics are used in the evaluations, which is the mean absolute error (MAE) and the signal aggregate error (SAE). These two metrics are defined as fellows.

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T}|\hat{y}_j(t) - y_j(t)| \tag{12}$$

$$\text{SAE} = \frac{1}{r_j}\left|\hat{r}_j - r_j\right|, \tag{13}$$

where $\hat{y}_j(t)$ and $y_j(t)$ are the estimated power consumption of appliance j and the ground truth, respectively; T is the duration of the time period; and $\hat{r}_j$ and r_j are the predicted total energy consumption and the ground truth of appliance j, respectively. MAE is used to measure the difference between the predict appliance power usage at every time instance and the ground truth of the appliance. SAE shows the relative error of the total energy consumption of the appliance [18].

B. Results and Discussions

The evaluation results (i.e., MAE and SAE) are presented in Table IV, where zero-shot means the pre-trained models are tested on the testing set directly, and few-shot means the pre-trained models' parameters are updated with the fine-tuning set and then the fine-tuned models are tested on the testing set.

There are no zero-shot results for s2p and Ensemble, since s2p is trained from scratch using the fine-tuning data and Ensemble requires fine-tuning data to combine the individual pre-s2p models. We use 10K fine-tuning sample instances' for updating the model parameters, which is collected on the first day of the fine-tuning set. The results of the transfer learning method (TL) proposed in [18] are presented as well. We found that with TL, the pre-trained model with fine-tuning performs even worse than the one without fine-tuning. We include both TL results with or without fine-tuning in the table. The parentheses following each pre-s2p model indicate the specific house in dataset REFIT used to pre-train the model. Only the pre-s2p model that achieves the best MAE performance for zero-shot of learning will be further updated with fine-tuning.

As can be seen from Table IV, both proposed pre-trained methods, i.e., MAML and Ensemble, outperform the traditional machine learning and transfer learning methods for all the tested appliances with respect to MAE and SAE. Next, we analyze the results in more detail in the following.

1) From Zero-shot to Few-shot: By updating the parameters (i.e., from zero-shot to few-shot fine-tuning), the transfer learning method used in [18] got an even worse result, with an -598.71% average improvement in MAE. This is because the TL method uses weak-relevant data in fine-tuning, which is the data from house 1 in UK-DALE. We try to diversify the data used for pre-training as in [18]. However, there is no guarantee that the data for fine-tuning comes from a similar distribution. Thus, we further improve fine-tuning by using only a small amount of data of house 2 (with no overlap with the unknown testing data). However, in some cases (e.g., pre-s2p model 1 of appliance kettle), negative transfer still happens, where the few-shot MAE (7.518) is slightly larger than the zero-shot MAE (6.124). Moreover, the improvements achieved by the pre-s2p models for other appliances are all insignificant. If we regard the pre-trained model's parameters as the neural network's starting point in the search space, the weight initialization of traditional transfer learning used for NILM is not optimal. Consequently, the DNNs get stuck in local minima with sub-optimal solutions.

The two proposed methods overcome this problem. On one hand, MAML achieves 53.41%, 25.67%, 21.61%, 40.24%, and 39.05% improvements in MAE for the kettle, dishwasher, washing machine, fridge, and microwave, respectively. On the other hand, Ensemble achieves 32.59%, 37.20%, 44.68%, 41.20% and 41.22% improvements in MAE compared to all the pre-trained models it uses.

2) With or Without Pre-training: We also compare the pre-trained models with the one trained from scratch (i.e., s2p). From the table, we can see that the best pre-trained model always outperforms s2p when using the same 10k new data samples. The improvement in MAE are 83.92%, 35.23%, 45.91%, 42.58%, and 75.60%, respectively, for different appliances. The improvement in SAE are 83.92%, 35.23%, 45.91%, 42.58%, and 75.60%, respectively, for the appliances.

We next investigate how much fine-tuning data is needed to achieve a good performance on the NILM task. We further expand the results for appliance kettle in house 2 in UK-DALE with sample size increased from 0 to 100k. The new MAE results are shown in Fig. 7. The pre-s2p model is pre-trained with house 9 data in REFIT. As can be seen, except for transfer

TABLE IV: Performance When Transferred to UK-DALE

Kettle	**Zero-shot**		**Few-shot**	
Model	MAE	SAE	MAE	SAE
s2p	-	-	21.287	0.367
TL [18]	6.260	0.060	16.879	0.043
pre-s2p model 1 (house 9)	6.124	0.155	7.518	0.140
pre-s2p model 2 (house 12)	9.539	0.248	-	-
pre-s2p model 3 (house 20)	32.889	0.816	-	-
MAML (proposed)	12.485	0.198	5.817	0.043
Ensemble (proposed)	-	-	**3.424**	**0.008**

Dish washer	**Zero-shot**		**Few-shot**	
Model	MAE	SAE	MAE	SAE
s2p	-	-	20.552	0.096
TL [18]	16.490	0.130	41.106	0.516
pre-s2p model 1 (house 5)	18.776	0.028	-	-
pre-s2p model 2 (house 7)	18.633	0.243	-	-
pre-s2p model 3 (house 9)	28.516	0.523	-	-
pre-s2p model 4 (house 13)	16.191	0.346	15.130	0.243
pre-s2p model 5 (house 16)	40.922	0.958	-	-
MAML (proposed)	17.882	0.361	**13.292**	0.254
Ensemble (proposed)	-	-	13.746	**0.033**

Washing machine	**Zero-shot**		**Few-shot**	
Model	MAE	SAE	MAE	SAE
s2p	-	-	9.574	0.733
TL [18]	14.840	0.500	22.941	0.899
pre-s2p model 1 (house 2)	9.629	0.679	-	-
pre-s2p model 2 (house 7)	8.356	0.626	7.751	0.431
pre-s2p model 3 (house 9)	9.631	0.785	-	-
pre-s2p model 4 (house 16)	11.070	0.767	-	-
pre-s2p model 5 (house 17)	8.613	0.600	-	-
MAML (proposed)	9.332	0.648	7.315	0.487
Ensemble (proposed)	-	-	**5.179**	**0.288**

Fridge	**Zero-shot**		**Few-shot**	
Model	MAE	SAE	MAE	SAE
s2p	-	-	28.842	0.109
TL [18]	17.000	0.090	33.078	0.266
pre-s2p model 1 (house 2)	27.793	0.191	-	-
pre-s2p model 2 (house 5)	30.245	0.165	-	-
pre-s2p model 3 (house 9)	26.497	0.085	26.210	0.116
pre-s2p model 4 (house 12)	30.787	0.417	-	-
MAML (proposed)	27.714	0.280	**16.562**	**0.068**
Ensemble (proposed)	-	-	16.887	0.088

Microwave	**Zero-shot**		**Few-shot**	
Model	MAE	SAE	MAE	SAE
s2p	-	-	13.174	1.194
TL [18]	4.770	0.080	10.973	0.019
pre-s2p model 1 (house 10)	4.767	0.345	4.498	0.259
pre-s2p model 2 (house 12)	7.739	0.755	-	-
pre-s2p model 3 (house 17)	6.849	0.112	-	-
pre-s2p model 4 (house 19)	5.275	0.093	-	-
MAML (proposed)	5.275	0.093	**3.215**	0.120
Ensemble (proposed)	-	-	3.490	**0.018**

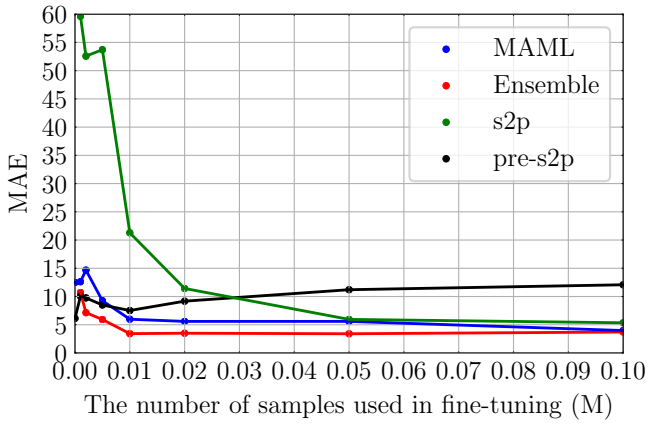


Fig. 7: The relationship between models' performance (MAE) and amount of new samples used in additional 10 gradient updates for appliance kettle.

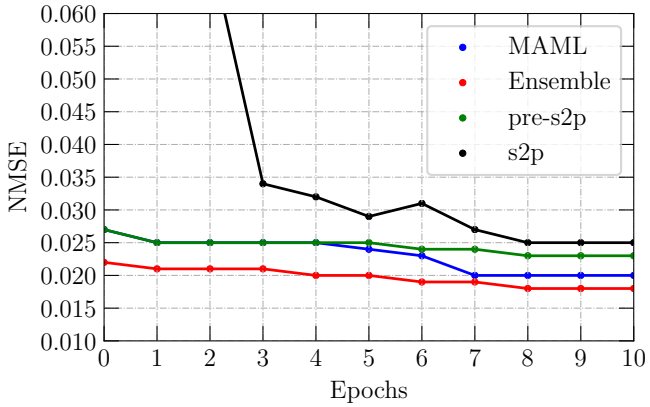


Fig. 8: Validation error (NMSE) of appliance kettle using 10k samples for domain adaptation in fine-tuning.

learning (pre-s2p), all other methods, including the model trained from scratch (s2p), achieve improved performance when more samples are used in fine-tuning. The pre-s2p model again suffers from the negative transfer problem, no matter how many samples are used to fine-tune its parameters. The MAEs of the two proposed methods (MAML and Ensemble) are initially (i.e., zero-shot) lower than that of s2p, and quickly reduces to stable values when 10K samples are used in fine-tuning. We also find the ensemble model outperforms MAML with a slightly smaller MAE. The s2p model needs at least 50k new samples to achieve the same MAE as MAML and at least 100K new samples to achieve the same MAE as Ensemble.

Fig. 8 presents an ablation study of validation error for appliance kettle using house 1's data in UK-DALE with different gradient steps for few-shot learning. We observe that all methods continue to improve (with a decreasing Normalized Mean Square Error (NMSE)) as there are more gradient steps, and the NMSEs converge to stable values after 8 gradient updates. The NMSE of the model trained from scratch (i.e., s2p) drops dramatically and is the highest among the four schemes. The two proposed methods both achieve smaller errors than the transfer learning model (i.e., pre-s2p).

TABLE V: Execution Time and Model Size of the Proposed Models for Kettle

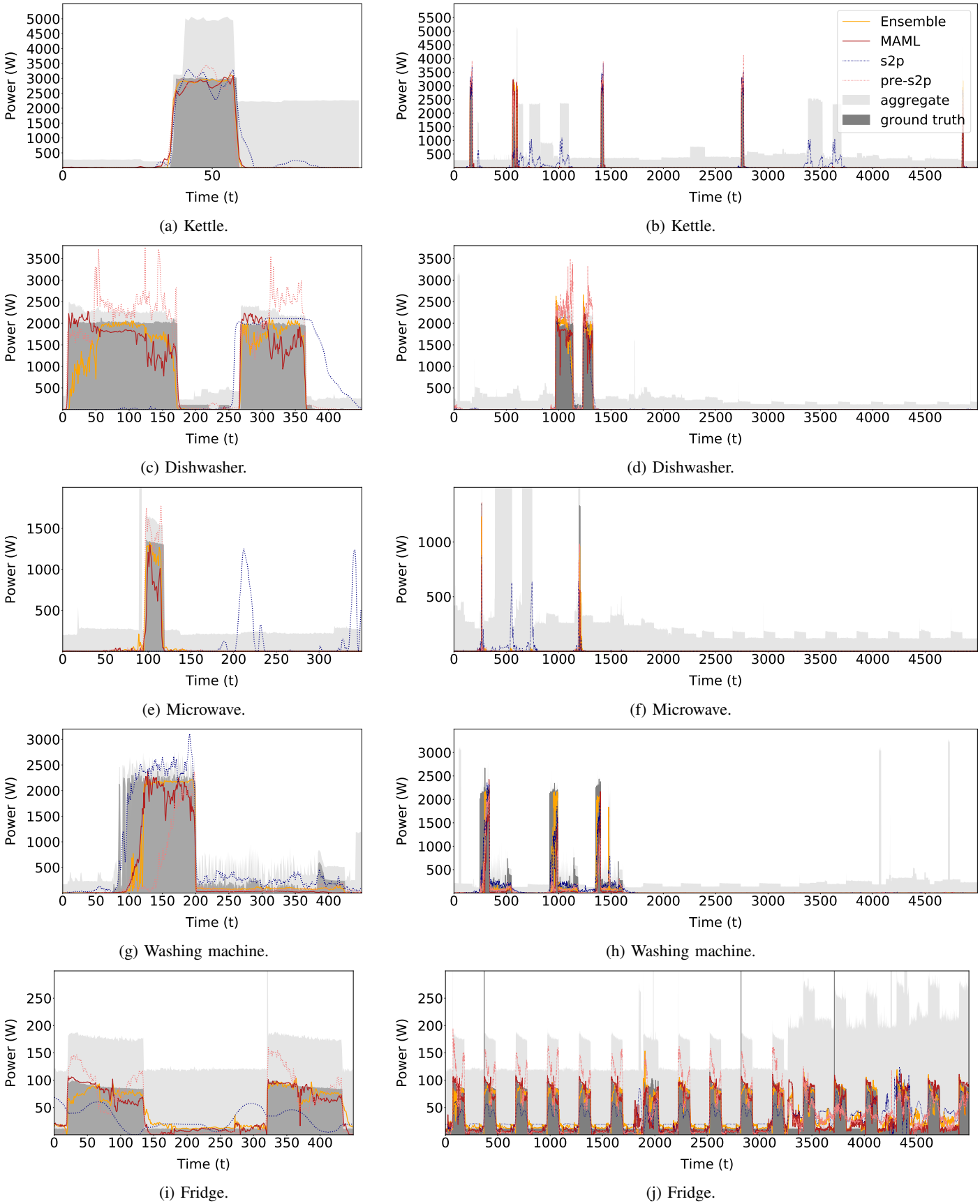
Model	Model size (MB)	Training time (Min)	Fine-tuning time (Min)
MAML	16.3	953.72	0.68
Ensemble	49.0	1173.06	2.01

3) Feature-based vs. Fine-tuning-based: We also plot the predicted power consumption values along with the ground truth for the five appliances, as well as the aggregated power consumption in house 2 in UK-DALE, including kettle, dishwasher, microwave oven, washing machine, and fridge, obtained with the four methods for a specific time period in Fig. 9. For each appliance, we include a zoomed-in plot of the curves as well as a plot for the entire time period. Since the same legend is used in all the plots, we only show the legend in Fig. 9(b) to make the plots more readable. The aggregated consumption is in the shade of light gray and the ground truth of the target appliance is in the shade of dark gray. It can be seen that the s2p model fails to predict the appliance's power consumption at some time instances, i.e., the appliance's state is off but it is predicted as on. This is quite obvious in Fig. 9(e) from 200 to 250, and from 320 to 350. The transfer learning model (pre-s2p) tends to overestimate the appliance's power consumption (e.g., see Fig. 9(c)) or underestimate it (e.g., see Fig. 9(g)). The two proposed methods' predictions match the ground truth much more closely than the other two schemes.

For the two proposed methods, it can be seen from Table IV that they achieve similar MAE performance for dishwasher, fridge, and microwave. For kettle and washing machine, Ensemble outperforms MAML in MAE with an improvement ratio of 41.13% and 29.20%, respectively. Ensemble also outperforms MAML by achieving a smaller SAE for the kettle, dish washer, washing machine, and microwave. While both methods achieve good prediction performance, the Ensemble results are slightly better than that of MAML. This may due to the fact that MAML is a fine-tuning based approach; training all its parameters using a small amount of fine-tuning data may result in the overfitting problem [53].

4) Computational Complexity and Execution Time: In Table V, we present the execution time and model size of the proposed models. The pre-training time of the Ensemble model given in the table is the accumulative time consumed by individual models. The results show that the Ensemble model requires more training time and larger model size than MAML. Training the base models in parallel on multiple GPUs will greatly improve the time efficiency of the pre-training process of the Ensemble model. Due to limited computing resources, we did not use this method in our experiments.

5) Limitation and Future Work: Due to limited datasets, the pre-training dataset and testing dataset are from the same country. The generalization of the pre-trained models across different countries needs to be further studied, as differences in appliances and usage behavior between different regions may be larger. Furthermore, we only investigate the transferability of the pre-trained models predicting the same type of appliance between different regions. Transferability among



(a) Kettle.

(b) Kettle.

(c) Dishwasher.

(d) Dishwasher.

(e) Microwave.

(f) Microwave.

(g) Washing machine.

(h) Washing machine.

(i) Fridge.

(j) Fridge.

Fig. 9: Comparison of predicted appliance power consumption obtained by Ensemble, MAML, sequence-to-point (s2p), and transfer learning (pre-s2p) models with ground truth for five appliances (i.e., kettle, microwave, fridge, washing machine, and dishwasher) with the house 2 meta testing set.

different types of appliances would be an interesting problem to study. Finally, deep learning models are vulnerable to designed adversarial training samples. Attackers may fool the training process by introducing tainted data in the fine-tuning data. It is essential to improve the security and robustness of the pre-trained deep learning model.

VII. Conclusions

In this paper, we developed two types of pre-trained models based on CNNs for solving the NILM problem with a focus on generalization. The Ensemble model uses a neural network to connect several trained base models, and few-shot learning fine-tuning to adapt to a new task. The MAML approach initializes the pre-trained model with good weights, and can quickly adapt to a new task with a few gradient updates. The proposed pre-trained models can effectively solve the NILM problem. Compared to transfer learning, our models require fewer data for adaptation, and can quickly adapt to new NILM tasks. In addition, our proposed methods outperform transfer learning with respect to prediction accuracy and can effectively avoid negative transfer. The proposed schemes are validated with two open-source datasets and comparison with the baseline schemes.

References

[1] Z. Niu, "Green communication and networking: A new horizon," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 3, pp. 629–630, Aug. 2020.

[2] M. Feng, S. Mao, and T. Jiang, "Boost: Base station on-off switching strategy for energy efficient massive mimo hetnets," in *Proc. IEEE INFOCOM 2016*, San Francisco, CA, Apr. 2016, pp. 1395–1403.

[3] M. Feng, S. Mao, and T. Jiang, "Dynamic base station sleep control and RF chain activation for energy efficient millimeter wave cellular systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9911–9921, Oct. 2018.

[4] M. Feng, S. Mao, and T. Jiang, "BOOST: Base station on-off switching strategy for green massive MIMO HetNets," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7319–7332, Nov. 2017.

[5] S. Hu, X. Chen, W. Ni, X. Wang, and E. Hossain, "Modeling and analysis of energy harvesting and smart grid-powered wireless communication networks: A contemporary survey," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 2, pp. 461–496, Apr. 2020.

[6] F. Uddin, "Energy-aware optimal data aggregation in smart grid wireless communication networks," *IEEE Trans. Green Commun. Netw.*, vol. 1, no. 3, pp. 358–371, Sept. 2017.

[7] M. Collotta and G. Pau, "An innovative approach for forecasting of energy requirements to improve a smart home management system based on BLE," *IEEE Trans. Green Commun. Netw.*, vol. 1, no. 1, pp. 112–120, Mar. 2017.

[8] L. Wang, S. Mao, B. M. Wilamowski, and R. M. Nelms, "Ensemble learning for load forecasting," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 2, pp. 616–628, Apr. 2020.

[9] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.

[10] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *MDPI Sensors*, vol. 12, no. 12, pp. 16 838–16 866, Dec. 2012.

[11] A. Ruano, A. Hernandez, J. Ureña, M. Ruano, and J. Garcia, "NILM techniques for intelligent home energy management and ambient assisted living: A review," *MDPI Energies*, vol. 12, no. 11, p. 2203, June 2019.

[12] C. Fischer, "Feedback on household electricity consumption: a tool for saving energy?" *Springer Energy Efficiency*, vol. 1, no. 1, pp. 79–104, May 2008.

[13] J. Leitão, P. Gil, B. Ribeiro, and A. Cardoso, "A survey on home energy management," *IEEE Access*, vol. 8, pp. 5699–5722, Jan. 2020.

[14] D. Murray, L. Stankovic, V. Stankovic, S. Lulic, and S. Sladojevic, "Transferability of neural network approaches for low-rate energy disaggregation," in *Proc. IEEE ICASSP 2019*, Brighton, UK, May 2019, pp. 8330–8334.

[15] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, May 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[17] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, July 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[18] M. D'Incecco, S. Squartini, and M. Zhong, "Transfer learning for non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1419–1429, Aug. 2019.

[19] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Proc. 2012 Int. Conf. Artif. Intell. Stat.*, La Palma, Canary Islands, Apr. 2012, pp. 1472–1482.

[20] M. Zhong, N. Goddard, and C. Sutton, "Signal aggregate constraints in additive factorial HMMs, with application to energy disaggregation," in *Proc. NIPS 2014*, Montreal, CA, Dec. 2014, pp. 3590–3598.

[21] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Proc. AAAI 2012*, Toronto, CA, July 2012, pp. 356–362.

[22] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proc. 2011 SIAM Int. Conf. Data Mining*, Mesa, USA, Apr. 2011, pp. 747–758.

[23] R. Bonfigli, E. Principi, M. Fagiani, M. Severini, S. Squartini, and F. Piazza, "Non-intrusive load monitoring by using active and reactive power in additive factorial hidden Markov models," *Elsevier Applied Energy*, vol. 208, pp. 1590–1607, Dec. 2017.

[24] B. Zhao, L. Stankovic, and V. Stankovic, "On a training-less solution for non-intrusive appliance load monitoring using graph signal processing," *IEEE Access*, vol. 4, pp. 1784–1799, Apr. 2016.

[25] K. He, L. Stankovic, J. Liao, and V. Stankovic, "Non-intrusive load disaggregation using graph signal processing," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1739–1747, Aug. 2016.

[26] G.-Y. Lin, S.-C. Lee, Y.-J. Hsu, and W.-R. Jih, "Applying power meters for appliance recognition on the electric panel," in *Proc. 2010 IEEE Conf. Ind. Electron. Appl.*, Taichung, Taiwan, June 2010, pp. 2254–2259.

[27] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, "Non-intrusive appliance load monitoring using low-resolution smart meter data," in *Proc. IEEE SmartGridComm'14*, Venice, Italy, Jan. 2014, pp. 535–540.

[28] Y.-H. Lin and M.-S. Tsai, "Non-intrusive load monitoring by novel neuro-fuzzy classification considering uncertainties," *IEEE Trans. Smart Grid*, vol. 5, no. 5, pp. 2376–2384, Aug. 2014.

[29] M. B. Figueiredo, A. De Almeida, and B. Ribeiro, "An experimental study on electrical signature identification of non-intrusive load monitoring (NILM) systems," in *Proc. 2011 Int. Conf. Adaptive Natural Comput. Algorithms*, Ljubljana, Slovenia, Apr. 2011, pp. 31–40.

[30] S. M. Tabatabaei, S. Dick, and W. Xu, "Toward non-intrusive load monitoring via multi-label classification," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 26–40, June 2016.

[31] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key technologies and open issues," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 4, pp. 3072–3108, Fourth Quarter 2019.

[32] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *Proc. AAAI 2018*, New Orleans, LA, Feb. 2018, pp. 1–8.

[33] C. Shin, S. Joo, J. Yim, H. Lee, T. Moon, and W. Rhee, "Subtask gated networks for non-intrusive load monitoring," in *Proc. AAAI 2019*, vol. 33, Honolulu, HI, Jan. 2019, pp. 1150–1157.

[34] K. Chen, Y. Zhang, Q. Wang, J. Hu, H. Fan, and J. He, "Scale-and context-aware convolutional non-intrusive load monitoring," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2362–2373, Nov. 2019.

[35] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, and A. Doulamis, "Context aware energy disaggregation using adaptive bidirectional LSTM models," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3054–3067, July 2020.

[36] J. Kelly and W. Knottenbelt, "Neural NILM: Deep neural networks applied to energy disaggregation," in *Proc. ACM BuildSys'15*, Seoul, South Korea, Nov. 2015, pp. 55–64.

[37] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. NIPS 2014*, Montreal, CA, Dec. 2014, pp. 3320–3328.

[38] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. ICML 2015*, vol. 37, Lille, UK, July 2015, pp. 97–105.
[39] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, Mar. 2018. [Online]. Available: https://arxiv.org/abs/1802.05365
[40] P. P. M. do Nascimento, "Applications of deep learning techniques on NILM," PhD Dissertation, Universidade Federal do Rio de Janeiro, 2016.
[41] A. M. Ahmed, Y. Zhang, and F. Eliassen, "Generative adversarial networks and transfer learning for non-intrusive load monitoring in Smart Grids," in *Proc. IEEE SmartGridComm 2020*, Tempe, AZ, Nov. 2020, pp. 1–7.
[42] Y. Liu, L. Zhong, J. Qiu, J. Lu, and W. Wang, "Unsupervised domain adaptation for non-intrusive load monitoring via adversarial and joint adaptation network," *IEEE Trans. Industrial Inform.*, Mar. 2021.
[43] Y. Liu, X. Wang, and W. You, "Non-intrusive load monitoring by voltage–current trajectory enabled transfer learning," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5609–5619, Sept. 2019.
[44] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv:1703.03400*, July 2017. [Online]. Available: https://arxiv.org/pdf/1703.03400
[45] N. Batra, R. Kukunuri, A. Pandey, R. Malakar, R. Kumar, O. Krystalakos, M. Zhong, P. Meira, and O. Parson, "Towards reproducible state-of-the-art energy disaggregation," in *Proc. ACM BuildSys 2019*, New York City, NY, Nov. 2019, pp. 193–202.
[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, June 2014.
[47] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *arXiv preprint arXiv:2004.05439*, Apr. 2020. [Online]. Available: https://arxiv.org/abs/2004.05439
[48] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL: Chapman and Hall/CRC, 2012.
[49] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study," *Scientific Data*, vol. 4, no. 1, pp. 1–12, Jan. 2017.
[50] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Scientific Data*, vol. 2, no. 1, pp. 1–14, Mar. 2015.
[51] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT 2010*, Sept. 2010, pp. 177–186.
[52] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," *arXiv preprint, arXiv:1412.6980*, Jan. 2017, [online] Available: https://arxiv.org/abs/1412.6980.
[53] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," *arXiv preprint arXiv:1707.03141*, Feb. 2017. [Online]. Available: https://arxiv.org/abs/1707.03141

Lingxiao Wang received the M.S. degree in Electrical and Computer Engineering from Auburn University, Auburn, AL in 2016 and the B.E. degree in Electrical Engineering and Automation from Nanjing University of Information Science and Technology, Nanjing, China in 2012. Since 2016, he has been pursuing a Ph.D. degree in the Department of Electrical and Computer Engineering at Auburn University. His research interests include deep learning, neural network optimization, and time-series prediction.

Shiwen Mao [S'99-M'04-SM'09-F'19] SHIWEN MAO received his Ph.D. in electrical engineering from Polytechnic University, Brooklyn, NY in 2004. After joining Auburn University, Auburn, AL in 2006, he held the McWane Endowed Professorship from 2012 to 2015 and the Samuel Ginn Endowed Professorship from 2015 to 2020 in the Department of Electrical and Computer Engineering. Currently, he is a professor and Earle C. Williams Eminent Scholar Chair, and Director of the Wireless Engineering Research and Education Center at Auburn University. His research interest includes wireless networks, multimedia communications, and smart grid. He is an Associate Editor-in-Chief of IEEE/CIC China Communications, an Area Editor of IEEE Transactions on Wireless Communications, IEEE Internet of Things Journal, IEEE Open Journal of the Communications Society, and ACM GetMobile, and an Associate Editor of IEEE Transactions on Cognitive Communications and Networking, IEEE Transactions on Network Science and Engineering, IEEE Transactions on Mobile Computing, IEEE Multimedia, IEEE Network, and IEEE Networking Letters, among others. He is a Distinguished Lecturer of IEEE Communications Society (ComSoc) and IEEE Council of RFID, and a Distinguished Speaker of IEEE Vehicular Technology Society. He received the IEEE ComSoc TC-CSR Distinguished Technical Achievement Award in 2019, Auburn University Creative Research & Scholarship Award in 2018, and NSF CAREER Award in 2010. He is a co-recipient of the 2021 IEEE Communications Society Outstanding Paper Award, the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the IEEE ComSoc MMTC 2018 Best Journal Award and 2017 Best Conference Paper Award, the Best Demo Award of IEEE SECON 2017, the Best Paper Awards from IEEE GLOBECOM 2019, 2016 & 2015, IEEE WCNC 2015, and IEEE ICC 2013, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is a Fellow of the IEEE.

Bogdan M. Wilamowski [SM'83-F'00-LF'14] received the M.S. degree in computer engineering, the Ph.D. degree in neural computing, and the Habilitation degree in integrated circuit design from the Gdansk University of Technology, Gdansk, Poland, in 1966, 1970, and 1977, respectively. He was the Director of the Alabama Micro/Nano Science and Technology Center, Auburn University, Auburn, AL, USA, from 2003 to 2016, where he is currently a Professor Emeritus with the Department of Electrical and Computer Engineering. He is also with The University of Information Technology and Management, Rzeszow, Poland. Dr. Wilamowski served as the Vice President of IEEE Computational Intelligence Society from 2000 to 2004, the President of IEEE Industrial Electronics Society from 2004 to 2005, and a member of the IEEE Board of Directors from 2012 to 2014. He was the Editor-in-Chief of the IEEE Transactions on Industrial Electronics from 2007 to 2010 and the IEEE Transactions on Industrial Informatics from 2011 to 2013. He also served as an associate editor for numerous other journals.

R. M. Nelms [S'78-M'82-SM'93-F'04] received the B.E.E. and M.S. degrees in electrical engineering from Auburn University, AL in 1980 and 1982, respectively. He received the Ph.D. degree in electrical engineering from Virginia Polytechnic Institute and State University, Blacksburg, VA in 1987. He is currently Professor and Chair of the Department of Electrical and Computer Engineering at Auburn University. His research interests are in power electronics, power systems, and electric machinery. In 2004, he was named an IEEE Fellow "for technical leadership and contributions to applied power electronics." He is a registered professional engineer in Alabama.