Feature Transfer and Rapid Adaptation for Few-Shot Solar Power Forecasting

Xin Ren 1, Yimei Wang 1, Zhi Cao 2, Fuhao Chen 3, Yujia Li 3 and Jie Yan 3,*

1 China Huaneng Clean Energy Research Institute, Beijing 102209, China; x_ren@qny.chng.com.cn (X.R.);
ym_wang@qny.chng.com.cn (Y.W.)
2 China Huaneng Group Co., Ltd., Beijing 100031, China; zhi_cao@chng.com.cn
3 School of New Energy, North China Electric Power University, Beijing 102206, China;
fuhaochen0410@gmail.com (F.C.); liyujia011025@163.com (Y.L.)
* Correspondence: yanjie@ncepu.edu.cn

Abstract: A common dilemma with deep-learning-based solar power forecasting models is their heavy dependence on a large amount of training data. Few-Shot Solar Power Forecasting (FSSPF) has been investigated in this paper, which aims to obtain accurate forecasting models with limited training data. Integrating Transfer Learning and Meta-Learning, approaches of Feature Transfer and Rapid Adaptation (FTRA), have been proposed for FSSPF. Specifically, the adopted model will be divided into Transferable learner and Adaptive learner. Using massive training data from source solar plants, Transferable learner and Adaptive learner will be pre-trained through a Transfer Learning and Meta-Learning algorithm, respectively. Ultimately, the parameters of the Adaptive learner will undergo fine-tuning using the limited training data obtained directly from the target solar plant. Three open solar power forecasting datasets (G EFCom2014) were utilized to conduct 24-h-ahead FSSPF experiments. The results illustrate that the proposed FTRA is able to outperform other FSSPF approaches, under various amounts of training data as well as different deep-learning models. Notably, with only 10-day training data, the proposed FTRA can achieve an RMSR of 8.42%, which will be lower than the 0.5% achieved by the state-of-the-art approaches.

Keywords: Few-Shot Solar Power Forecasting; deep-learning; transfer learning; meta-learning

1. Introduction

On the one hand, due to their environmental pollution, greenhouse gas emissions, non-renewable nature, and traditional fossil energy sources are increasingly falling short of meeting the requirements of the world for sustainable development [1]. On the other hand, renewable energy sources, such as wind and solar power, are gaining significant global attention due to their clean, low-carbon, and renewable attributes [2]. Benefiting from technological advances as well as policy support, Solar Power (SP) has rapidly developed in recent years and become an important part of the new power system [3]. Nevertheless, SP output is susceptible to weather conditions, such as irradiance, resulting in significant volatility and randomness [4]. This poses a great challenge to the stability and security of the whole power system [5]. Solar Power Forecasting (SPF) is designed to forecast the SP for a desired future period, which can provide references for dispatch and control in power systems [6].

Deep-learning-based SPF has gained prominent attention in the current research, benefiting from its ability to learn nonlinear complex features and adaptability to various types of SP datasets [7–9]. Long Short-Term Memory Neural Networks (LSTM) [10], Gate Recurrent Unit (GRU) [11], and Transformer [12] have been widely adopted in SPF, to accurately capture the changing patterns of SP and the complex relationship between SP
and meteorological factors, thus improving the forecasting accuracy. However, the performance of the aforementioned deep-learning models heavily relies on the availability of a substantial amount of training data [13, 14].

For newly built or expanded solar plants, it poses a challenge to gather an adequate amount of training data due to their limited operating time [15]. To reduce the losses caused by data scarcity, **Few-Shot Solar Power Forecasting (FSSPF)** is investigated in this paper. The objective of FSSPF is to **obtain accurate SPF models using a limited amount of training data**.

Currently, there are three approaches to implementing FSSPF: Data Augmentation, Metric Learning, and Transfer Learning:

**Data Augmentation** refers to the utilization of auxiliary data or information to expand the limited number of original samples or enhance their features [16]. A dual-dimensional time series adversarial neural network has been proposed in [17] to enhance the low-value-density SP data by two dimensions (time dimension and feature dimension) and obtain high-value-density feature data.

**Metric learning** refers to the process of selecting an appropriate distance function for calculating the similarity between different datasets. This similarity measurement serves as the foundation for expanding training data or weighting models [18]. The Mahalanobis Distance Similarity metric has been adopted in [19]. Firstly, the gray correlation between each meteorological factor and the output power of the solar plant is calculated. Secondly, several similar days are selected to expand the training set using the Mahalanobis distance.

Although the above two methods can enhance the accuracy of FSSPF to some extent, the robustness and effectiveness of these two approaches on deep-learning models can hardly be guaranteed, due to the low diversity of the original few-shot dataset [20]. In contrast, Transfer Learning methods dominate the current deep-learning-based FSSPF.

**Transfer learning** (TL), which aims to apply knowledge learned from the source domain to the target domain, is now accepted as the dominant approach in the field of Few-Shot Learning (FSL) [21]. Corresponding to the FSSPF, we will refer to solar plants that contain only a small amount of operational data as **Target Solar Plants (TSP)**, while solar plants that have a large amount of operational data are referred to as **Source Solar Plants (SSP)**. A TL-based FSSPF consists of two stages: pre-raining and fine-tuning. The former will optimize all the parameters in the models with massive training data from SSP, while the latter will use the limited training data from the TSP to selectively fine-tune the model parameters. A digital twin model for the FSSPF based on TL and LSTM was proposed in [22], which chooses to freeze the first layer of the models during the fine-tuning stage, and then fine-tunes the other layers using a small amount of training data from the TSP. Additionally, [23] has selected to freeze the earlier layers and only fine-tune the weight values of the last layer of the model in the fine-tuning stage.

Existing TL-based FSSPF methods directly use the parameters pre-trained from the SPF as initialization parameters for the fine-tuning stage. However, this implementation may make it difficult for the pre-trained model to rapidly adapt to the TSP, due to the large differences between the SSP and the TSP. To address this problem, Meta-Learning has been employed in this paper.

**Meta-Learning** [24], which is widely known as “learn to learn”, is the process of distilling the knowledge from multiple learning tasks and using this knowledge to improve future learning performance, and has excelled in many FSL tasks [25–28]. With the excellent adaptations and scalability to deep neural networks, Reptile [29] has greatly facilitated the development of related fields [30]. Compared with TL, the pre-trained models obtained through Reptile can more accurately identify the underlying data characteristics of the target task, and thus achieve faster adaptation.

Through the integration of TL and Reptile, an approach to **Feature Transfer and Rapid Adaptation (FTRA)** is proposed for FSSPF in this paper. Compared with previous studies, the contributions of this paper are summarized as follows:
In the proposed FTRA, the adopted deep-learning-based SPF model will be reasonably divided into Transferable Learner and Adaptive Learner, which are responsible for Feature Transfer and Rapid Adaptation, respectively.

TL and Reptile were integrated to develop different pre-training and fine-tuning strategies for parameters in different parts of the model, so as to extract valuable knowledge from the SSP to the TSP and adapt the pre-trained model to TSP rapidly.

In addition, to validate the proposed FTRA on FSSPF, three open-solar power forecasting datasets from GEFCom2014 [31] were utilized. A 24 h SPF will be conducted ahead of time, which will use Numerical Weather Prediction (NWP) to forecast the SP at the corresponding time. SPF models, LSTM-based, GRU-based, and Transformer-based, will be adopted to examine the generalizability of the proposed FTRA to various types of deep-learning models. Three different sizes of training data (10-day, 20-day, and 30-day) and cross-validation methods will be used to comprehensively compare the performance of different approaches on FSSPF.

This paper is organized as follows: Section 2 is the preliminary for the proposed approach, which will introduce the detailed structure of the adopted SPF models. The implementation of the proposed FTRA will be explained in Section 3. A case study for FSSPF will be presented in Section 4. Section 5 provides conclusions and outlooks for future works.

2. Solar Power Forecasting Models

The diagram of the adopted SPF models adopted is shown in Figure 1. The input of the SPF model is 24 h ahead NWP data, while the output is 24 h ahead SPF results.

![Figure 1. The diagram of the SPF models.](image)

Each SPF model contains three components: NWP Embedding Layer (NWPEL), Meteorological Encoder (ME), and Power Output Layer (POL). NWPEL is designed to embed NWP vectors into meteorological feature vectors, ME is utilized to map meteorological feature vectors into output vectors, and POL is used to map output vectors into the final SPF results. For the three deep-learning SPF models adopted in this paper, their NWPEL and POL are identical, composed of single-layer Fully Connected Neural Networks (FCNN). The main difference between the three models reflects on ME, which can be described as follows:

1. **Transformer-based**: The encoder layer in [32] is used as the ME in the Transformer-based SPF model, which contains a multi-head self-attention mechanism, residual connection, layer normalization, and position-wise FCNN;
2. **LSTM-based**: Single-layer LSTM in [33] is used as the ME in the LSTM-based SPF model, which comprises of forget gate, input gate, update gate, and output gate;
3. **GRU-based**: Single-layer GRU in [34] is used as the ME in the LSTM-based SPF model, which is made up of a reset gate and an update gate.
3. Methodology: FTRA

In this section, an approach to Feature Transfer and Rapid Adaptation (FTRA) for FSSPF will be proposed. It can be learned from Figure 2 that the proposed FTRA consists of 4 steps: Division of the SPF Model, Transfer-Pre-Training, Meta-Pre-Training, and fine-tuning. The detailed implementation is described as follows.

### 3.1. Division of SPF Model

In FTRA, each adopted SPF model will first be divided into Transferable Learner and Adaptive Learner. In particular, the Transferable Learner consists of ME and POL, while the Adaptive Learner is made up of NWPEL only. Formally, we consider an SPF model represented by a parametrized function $f_\theta$ with parameters $\theta$. Further, Transferable Learner will be represented by a function $f_{\theta_{TL}}$ with parameters $\theta_{TL}$, while Adaptive Learner will be represented by a function $f_{\theta_{AL}}$ with parameters $\theta_{AL}$.

### 3.2. Transfer-Pre-Training

In Transfer-Pre-Training, massive training data from SSP will be used to update all parameters in the adopted SPF model. The optimization objective of the Transfer-Pre-Training is to minimize the forecasting error of the model in all the SSP [20], and its detailed implementation can refer to Algorithm 1.

---

**Algorithm 1 Transfer-Pre-Training Algorithm**

Require: $N$, $l_{\text{TTP}}$, $L$, $U$

1. Randomly initialize $\theta$
2. while not done do
3. Randomly select $N$ SSPs as $\{spp_i\}$
4. for all $spp_i$ do
5. $\theta \rightarrow \theta_i$
6. Sample one batch datapoints $(nwp, p)$ from $spp_i$
7. Update $\theta_i \leftarrow U\left( L\left( f_{\theta_i}(nwp), p \right), l_{\text{TPP}} \right) \right)$
8. end for
9. Update $\theta \leftarrow \frac{1}{N} \cdot \sum_{i=1}^{N} \theta_i$
10. end while

---
Within Algorithm 1, $N$ denotes the number of the selected SSPs per updating epoch, where $\{nwp, p\}$ represents the value of the NWP data and their corresponding SP. Each batch contains $bs_{\text{tr}}$ training samples. $l_{\text{TPP}}$, $L$, and $U$ denote the learning rate, loss function and optimizer in the updating of the $\theta$, respectively.

3.3. Meta-Pre-Training

After the pre-training with Algorithm 1, the parameters of the Adaptive Learner, $\theta_{\text{AL}}$, will first be re-initialized randomly, and then be pre-trained with the Reptile [29] algorithm in Meta-Pre-Training. During this process, the parameters of the Transferable Learner, $\theta_{\text{TL}}$, will remain fixed.

The optimization objective of the Meta-Pre-Training is not only to minimize the forecasting error of the model in all SSPs, but also to maximize the gradient similarity of the samples from the same SSP [29]. Based on this, the Adaptive Learner is able to distinguish the respective gradient using limited available samples, and then make a rapid adaptation to TSP. The detailed implementation of Meta-Pre-Training can refer to Algorithm 2.

**Algorithm 2 Meta-Pre-Training Algorithm**

**Require:** $N$, $S$, $l_{\text{inner}}$, $l_{\text{outer}}$, $L$, $U$

1. Randomly initialize $\theta_{\text{AL}}$
2. while not done do
3. Randomly select $N$ SSPs as $\{spp_i\}$
4. for all $spp_i$ do
5. $\theta_{\text{AL}} \rightarrow \theta_{i_{\text{AL}}}$
6. Sample $S - \text{batch} \{nwp^i, p^i\}$ from $spp_i$ //Inner Loop
7. for all $[nwp^i, p^i]$ do
8. Update $\theta_{i_{\text{AL}}} \leftarrow U(L(f_{[g_{n}, g_{\text{ AL}}]}(nwp^i), p^i), l_{\text{inner}})$
9. end for
10. end for //Outer Loop
11. Update $\theta_{\text{AL}} \leftarrow \theta_{\text{AL}} + \frac{l_{\text{inner}}}{N} \Sigma_{i=1}^{N}(\theta_i^{\text{AL}} - \theta_{\text{AL}})$
12. end while

As shown in Algorithm 2, the “Inner–Outer Loop” mechanism has been introduced to achieve the above bilevel optimization [29]. $S$ denotes the number of Inner-Loop inside each Outer Loop, while $l_{\text{inner}}$ and $l_{\text{outer}}$ represent the learning rates in the Inner-Loop and Outer Loop, respectively. $\{nwp^i, p^i\}$ represents the value of the NWP data and their corresponding wind power in the $j$th Inner Loop. Each batch contains $bs_{\text{tr}}$ training samples. The other notations have the same meaning as in Algorithm 1.

3.4. Fine-Tuning

After pre-training with Algorithms 1 and 2, the Adaptive Learner will further be fine-tuned using limited training data from TSP itself. During this process, the parameters of the Transferable Learner will remain fixed. After fine-tuning, the final SPF model for TSP can be obtained.
The optimization objective of fine-tuning is to minimize the forecasting error of the model in the TSP, and its detailed implementation can refer to Algorithm 3. $l_{\text{FT}}$ in Algorithm 3 denotes the learning rate in updating $\theta^\text{AL}$. Each batch contains $bs_R$ training samples here. Other notations have the same meaning as in Algorithm 1.

**Algorithm 3 Fine-Tuning Algorithm**

Require: $l_{\text{FT}}$, $L$, $U$

1. while not done do
2. Sample one batch datapoints $(\text{nwp}, p)$ from TSP
3. Update $\theta^\text{AL} \leftarrow U(L \left(f(\theta^\text{R}, \theta^\text{AL}((\text{nwp})), p), l_{\text{FT}}\right)$
4. end while

4. Case Study
4.1. Data Description

This paper utilizes the publicly available dataset of three SPPs from the 2014 Global Energy Competition (GEFCom2014) [31] for the case study. The three SPPs are located in a region of Australia, and the exact location is unknown. Each SPP contains NWP data and SP data, and the time resolution is 1-h. Two-year data (from 1 April 2012 to 1 April 2014) will be adopted in this case.

As for NWP data, there are 12 meteorological variables included: total column liquid water, total column ice water, surface pressure, relative humidity at 1000 mbar, total cloud cover, 10-metre U wind component, 10-metre V wind component, 2-metre temperature, surface solar rad down, surface thermal rad down, top net solar rad, and total precipitation. The forecasting horizon of NWP is 24 h (1:00 today to 0:00 tomorrow). To be able to be used for model training, original NWP data need to be normalized. Specifically, we will use the Maximum–Minimum normalization method to pre-process the different NWP items. It is worth noting that, unlike the NWP items from the SSPs, which are normalized according to their own maximum and minimum values, the NWP items in the TSP will be normalized according to the maximum and minimum values of the SSPs.

In addition to SP data, it has been normalized to values between 0 and 1.

4.2. Settings of SPP and TPP

For three SPPs in GEFCom2014, we will mark them as SPP1, SPP2, and SPP3. Each SPP will be picked out once to be the TSP and the remaining two SPPs will be treated as SSPs, corresponding to one FSSPF setting. Finally, there are a total of three FSSPF settings in this case, which detailed information can refer to Table 1.

<table>
<thead>
<tr>
<th>Names</th>
<th>TSP</th>
<th>SSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-1</td>
<td>SPP1</td>
<td>SPP2, SPP3</td>
</tr>
<tr>
<td>S-2</td>
<td>SPP2</td>
<td>SPP1, SPP3</td>
</tr>
<tr>
<td>S-3</td>
<td>SPP3</td>
<td>SPP1, SPP2</td>
</tr>
</tbody>
</table>

4.3. Evaluation Metric

In order to evaluate the FSSPF performance of the proposed approach, Root Mean Square Error (RMSE) and Mean Average Error (MAE) have been adopted as the evaluation metrics, which calculation can refer to (1) and (2), respectively.
$$\text{RMSE} = \left( \frac{1}{T} \sum_{t=1}^{T} \left( \frac{P_M^t - P_F^t}{P_R} \right) \right) \times 100\%$$

$$\text{MAE} = \frac{1}{T} \sum_{t=1}^{T} \left| \frac{P_M^t - P_F^t}{P_R} \right| \times 100\%$$

$P_R$ is the rated capacity of the SPP, and its value is 1 in this case. $P_M^t$ and $P_F^t$ are the measured power and the forecasted power at $t$-th time step, respectively. $T$ is the total number of steps in the forecasting time horizon, and its value is 24 in this case.

### 4.4. Evaluation Method

To comprehensively evaluate the forecasting performance of the proposed approach in different scenarios, the K-Fold Cross-Validation method (KFCV) will be introduced. The schematic diagram of the KFCV for FSSPF is shown in Figure 3.

![Figure 3. The diagram of the KFCV for FSSPF.](image)

Given a limited amount of the training dataset, the total dataset will be divided into $K$ sub-datasets in chronological order, corresponding to different operational scenarios. Each sub-dataset is given one chance to be viewed as the training dataset, while the others are treated as testing datasets. There are a total of $K$ FSSPF experiments, where the final evaluation results rely on their average results.

### 4.5. Comparison Methods

In order to demonstrate the advanced nature of the proposed FTRA, we are going to compare it with other six typical methods:

- **Baseline** [23]: supervised learning method. Models will be trained from random initialization parameters, using limited training data from the TSP.

- **Upper-Bound**: supervised learning method. Models will be trained from random initialization parameters, using massive training data from the TSP.

- **TL-FT-Out** [23]: TL method. Models will be first pre-trained with Algorithm 1, and then only the parameters in POL will be fine-tuned using limited training data from the TSP.

- **TL-FT-All** [22]: TL method. Models will be first pre-trained with Algorithm 1, and then all parameters will be fine-tuned using limited training data from the TSP.

- **TL-FT-Inp**: TL method. Models will be first pre-trained with Algorithm 1, and then only the parameters in NWPEL will be fine-tuned using limited training data from the TSP.
Reptile [29]: Meta-Learning method. Models will be first pre-trained with the Reptile algorithm and massive training data from SSP, and all parameters will be fine-tuned using limited training data from the TSP.

4.6. Hyperparameters

The detailed hyperparameters in the proposed FTRA have been described as below:

Firstly, for three adopted SPF models, the dimension of their hidden layer is 64. Secondly, \( L \) and \( U \) will be set as \( L2 \) and Adam, respectively. Secondly, Early Stopping [35] will be adopted in this paper to timely terminate the training process. Particularly, in Algorithms 1–3, 20% of samples will be randomly selected from the training dataset to act as the validation dataset, and the maximum tolerance to overfitting is \( E \) epochs. The other hyperparameters will vary with respective conditions:

**Transformer-based:**

\[ S-1: N, l_{TPT}, bs_{ptu}, \text{ and } E, \text{ will be set as } 2, 0.0001, 64, \text{ and } 50, \text{ in Algorithm 1}; \]
\[ S-2: N, l_{TPT}, bs_{ptu}, \text{ and } E, \text{ will be set as } 2, 0.0001, 64, \text{ and } 50, \text{ in Algorithm 2}; \]
\[ S-3: N, l_{TPT}, bs_{ptu}, \text{ and } E, \text{ will be set as } 2, 0.0001, 64, \text{ and } 50, \text{ in Algorithm 3}. \]

**LSTM-based:**

\[ S-1: N, l_{TPT}, bs_{ptu}, \text{ and } E, \text{ will be set as } 2, 0.0001, 64, \text{ and } 50, \text{ in Algorithm 1}; \]
\[ S-2: N, l_{TPT}, bs_{ptu}, \text{ and } E, \text{ will be set as } 2, 0.0001, 64, \text{ and } 50, \text{ in Algorithm 2}; \]
\[ S-3: N, l_{TPT}, bs_{ptu}, \text{ and } E, \text{ will be set as } 2, 0.0001, 64, \text{ and } 50, \text{ in Algorithm 3}. \]

**GRU-based:**

\[ S-1: N, l_{TPT}, bs_{ptu}, \text{ and } E, \text{ will be set as } 2, 0.0001, 64, \text{ and } 50, \text{ in Algorithm 1}; \]
\[ S-2: N, l_{TPT}, bs_{ptu}, \text{ and } E, \text{ will be set as } 2, 0.0001, 64, \text{ and } 50, \text{ in Algorithm 2}; \]
\[ S-3: N, l_{TPT}, bs_{ptu}, \text{ and } E, \text{ will be set as } 2, 0.0001, 64, \text{ and } 50, \text{ in Algorithm 3}. \]

4.7. FSSPF Results

Using 10-day, 20-day, and 30-day training data, the FSSPF results of the proposed FTRA and other comparison approaches have been presented in Tables 2 and 3.

From Tables 2 and 3, it can be deduced that: (1) Under the same amount of training data from TSP, the proposed FTRA can significantly improve the accuracy of FSSPF, compared with Baseline. These results illustrate the effectiveness of the proposal.

**Table 2.** RMSE of the proposed FTRA and other comparison approaches.
Transformer-based

TL-FT-All [22] 8.79 8.93 9.03 8.92 8.16 8.17 8.56 8.30 7.94 8.14 8.34 8.14
Reptile [29] 8.61 9.23 8.97 8.94 8.28 8.53 8.48 8.43 8.22 8.32 8.31 8.28
FTRA(Ours) 8.01 8.81 8.43 8.42 7.74 8.07 8.21 8.01 7.49 8.08 8.16 7.91

LSTM-based

TL-FT-All [22] 8.79 8.93 9.03 8.92 8.16 8.17 8.56 8.30 7.94 8.14 8.34 8.14
Reptile [29] 8.27 9.23 8.97 8.94 8.28 8.53 8.48 8.43 8.22 8.32 8.31 8.28
FTRA(Ours) 8.22 9.06 9.54 8.94 7.61 8.35 9.12 8.36 7.40 8.24 8.95 8.20

GRU-based

TL-FT-All [22] 8.79 8.93 9.03 8.92 8.16 8.17 8.56 8.30 7.94 8.14 8.34 8.14
TL-FT-Inp [29] 8.27 9.23 8.97 8.94 8.28 8.53 8.48 8.43 8.22 8.32 8.31 8.28
Reptile [29] 8.27 9.23 8.97 8.94 8.28 8.53 8.48 8.43 8.22 8.32 8.31 8.28
FTRA(Ours) 8.22 9.06 9.54 8.94 7.61 8.35 9.12 8.36 7.40 8.24 8.95 8.20

* In Upper-bound, the amount of training data is 23 months. AVE represents the average results of the three FSSPF settings.

Table 3. MAE of the proposed FTRA and other comparison approaches.
In **Upper-bound**, the amount of training data is **23 months**. "AVE" represents the average results of the three FSSPF settings.

FTRA. (2) The forecasting accuracy of the proposed FTRA is, to varying degrees, better than the other comparison approaches for different models and different amounts of training data, which demonstrates the effect of the Feature Transfer and Rapid Adaptation. (3) When the length of the training samples is 10-day, 20-day and 30-day, respectively, the forecasting accuracy of FTRA steadily improves with an increase in the number of training samples. The forecasting error difference with Upper-Bound is also quite minimal. The RMSE of FTRA is only approximately 1.16% larger on average, and the MAE is only about 0.94% larger on average, further illustrating the efficiency and supremacy of the proposed FTRA.

On the one hand, the solar power forecasted curves resulting from different approaches in a specific period are shown in Figure 4. On the one hand, the forecasting performance of the Baseline is clearly subpar in the “valley” of the forecasted curve. The forecasting performance of FTRA is noticeably superior to that of the Reptile algorithm in the “peak” of the forecasting curve, where solar power fluctuates greatly and the training samples are limited. On the other hand, the FTRA approach’s forecasting accuracy is marginally more excellent than the TL-FT-Inp approach when the power is falling or low, which confirms the forecasted outcomes in Tables 2 and 3. The forecasting curves illustrate that the models’ capacity to adapt to changing scenarios is enhanced by FTRA, and it is possible to build an accurate mapping from NWP to solar power, which raises the forecasting accuracy of FSSPF.

**Figure 4.** Solar power curves forecasted by different approaches *. * FSSPF setting: S-1; SPF model: Transformer-based; **Training set:** 1 April 2012–10 April 2012 (20% samples will be randomly selected as the validation set); **Testing set:** 13 May 2012–17 May 2012.

4.8. Computational Costs

This case is implemented in PyTorch’s deep-learning library. The configurations of the simulation computer are Intel Core i7-12700H processor and NVIDIA GeForce RTX 3060 Laptop GPU using Windows 11 Operating System.

The computational costs of neural networks can be divided into two parts: training and inference. Once the neural networks are well trained, their computational time in the inference phase is negligible at less than 1 s. Hence, the main computational costs of the proposed FTRA in this paper are reflected in the training of the SPF models. The specific computation time of each algorithm in FTRA for different scenarios is shown in Table 4. It can be concluded that the computation time of the proposed FTRA is small enough for practical applications.
Table 4. Computation Time of different septs in the proposed FTRA (Unit: s).

<table>
<thead>
<tr>
<th>Settings</th>
<th>Models</th>
<th>Algorithm1</th>
<th>Algorithm2</th>
<th>Algorithm3*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Transformer-based</td>
<td>54</td>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>LSTM-based</td>
<td>75</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>GRU-based</td>
<td>14</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>S-2</td>
<td>Transformer-based</td>
<td>75</td>
<td>17</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>LSTM-based</td>
<td>100</td>
<td>41</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>GRU-based</td>
<td>86</td>
<td>3</td>
<td>18</td>
</tr>
<tr>
<td>S-3</td>
<td>Transformer-based</td>
<td>18</td>
<td>22</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>LSTM-based</td>
<td>12</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>GRU-based</td>
<td>118</td>
<td>13</td>
<td>10</td>
</tr>
</tbody>
</table>

* The computation time of Algorithm 3 is average one-fold in KFCV.

5. Conclusions

For the purpose of developing accurate solar power forecasting models for newly built solar power plants with only a small amount of training data available, an approach to Feature Transfer and Rapid Adaptation (FTRA) is proposed in this paper. Building on the existing TL methods, the contributions of the proposed FTRA are reflected in two aspects:

1. FTRA will divide the adopted deep-learning-based SPF model into the Transferable Learner and the Adaptive Learner, which will take charge of Feature Transfer and Rapid Adaptation, respectively.
2. Through integrating TL and Reptile, the parameters of the Transferable Learner and the Adaptive Learner will be assigned different pre-training and fine-tuning strategies.

By doing so, FTRA can transfer valuable knowledge from the source solar plants, while simultaneously achieving rapid adjustment to the designated target solar plant.

One publicly available solar power dataset (GEFCom2014) and three deep-learning SPF models have been adopted in our case study to validate the proposed FTRA. The results illustrate that the proposed FTRA is able to outperform the other state-of-the-art methods, under different amounts of training data, different SPF models, and different “SPP-TSP” settings.

To further improve the proposed FTRA, some work can be done in the future: Firstly, meteorological factors like solar radiation and air humidity are closely related to the changes in months and seasons. To further enhance the adaptability of the pre-trained models to different meteorological conditions, the segmentation of the original datasets deserves to be improved. Secondly, in the pre-training stage, the correlation of features between the SSPs and TSP should be analyzed and incorporated into the parameters updating process.

Author Contributions: Conceptualization, X.R. and Y.W.; methodology, Z.C. and F.C.; software, F.C., validation, X.R. and J.Y.; formal analysis, Y.W., F.C. and Z.C., investigation, Y.L., F.C. and J.Y.; data curation, Y.L. and F.C.; writing-original draft preparation, Y.L. and F.C., writing-review and editing, F.C., J.Y. and Z.C.; visualization, F.C. and J.Y., supervision, J.Y., project administration, X.R., Y.W. and Z.C.; funding acquisition, X.R. Y.W. and Z.C. All authors have read and agreed to the published version of the manuscript.
Funding: Key technology research and system development for the construction of group-level intelligent operation and maintenance platform, China Huaneng Group Technology Project, Grant/Award Number: HNKJ21-H52.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available with reference number [31]. They can be downloaded through this link: https://www.sciencedirect.com/science/article/abs/pii/S0169207016000133.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSSPF</td>
<td>Few-Shot Solar Power Forecasting</td>
</tr>
<tr>
<td>FTRA</td>
<td>Feature Transfer and Rapid Adaptation</td>
</tr>
<tr>
<td>SP</td>
<td>Solar Power</td>
</tr>
<tr>
<td>SSP</td>
<td>Source Solar Plants</td>
</tr>
<tr>
<td>TSP</td>
<td>Target Solar Plants</td>
</tr>
<tr>
<td>SPF</td>
<td>Solar Power Forecasting</td>
</tr>
<tr>
<td>LSTSM</td>
<td>Long Short-Term Memory Neural Network</td>
</tr>
<tr>
<td>GRU</td>
<td>Gate Recurrent Unit</td>
</tr>
<tr>
<td>TL</td>
<td>Transfer learning</td>
</tr>
<tr>
<td>FSL</td>
<td>Few-Shot Learning</td>
</tr>
<tr>
<td>NWP</td>
<td>Numerical Weather Prediction</td>
</tr>
<tr>
<td>NWPEL</td>
<td>NWP Embedding Layer</td>
</tr>
<tr>
<td>ME</td>
<td>Meteorological Encoder</td>
</tr>
<tr>
<td>POL</td>
<td>Power Output Layer</td>
</tr>
<tr>
<td>FCNN</td>
<td>Fully-Connected Neural Networks</td>
</tr>
<tr>
<td>GEFCom2014</td>
<td>2014 Global Energy Competition</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Square Error</td>
</tr>
<tr>
<td>MAE</td>
<td>Mean Average Error</td>
</tr>
<tr>
<td>KFCV</td>
<td>K-Fold Cross-Validation method</td>
</tr>
</tbody>
</table>

References


Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.