Energy and AI 14 (2023) 100249

Contents lists available at ScienceDirect

Energy and AI

journal homepage: www.elsevier.com/locate/egyai

Model selection, adaptation, and combination for transfer learning in wind and photovoltaic power forecasts

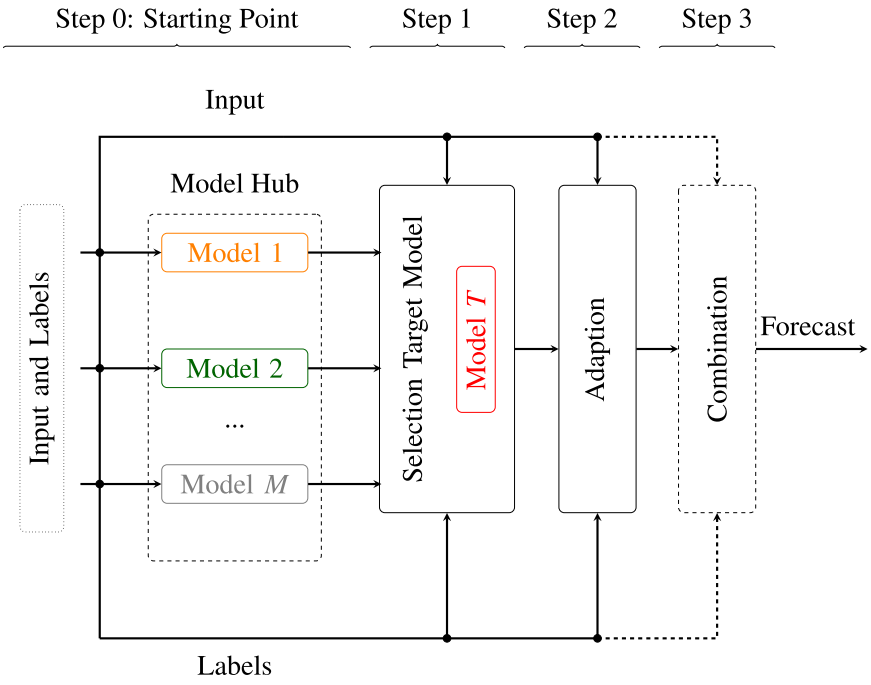
Jens Schreiber *, Bernhard Sick

University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany

HIGHLIGHTS

- With less than 90 days data, fine-tuning a source model is often disadvantageous.
- With less than 30 days data, any adaption of a source model is often disadvantageous.
- With more than 30 days data, an adaption through a linear regression is advantageous.
- Results can be significantly improved through ensemble techniques.
- Ensembles' 30-day training produces mean error akin to a year's training.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Transfer learning
Time series
Renewable energies
Temporal convolutional neural network
Ensembles
Wind and photovoltaic power

ABSTRACT

There is recent interest in using model hubs – a collection of pre-trained models – in computer vision tasks. To employ a model hub, we first select a source model and then adapt the model for the target to compensate for differences. There still needs to be more research on model selection and adaption for renewable power forecasts. In particular, none of the related work examines different model selection and adaptation strategies for neural network architectures. Also, none of the current studies investigates the influence of available training samples and considers seasonality in the evaluation. We close these gaps by conducting the first thorough experiment for model selection and adaptation for transfer learning in renewable power forecast, adopting recent developments from the field of computer vision on 667 wind and photovoltaic parks from six datasets. We simulate different amounts of training samples for each season to calculate informative

* Corresponding author.
E-mail address: j.schreiber@uni-kassel.de (J. Schreiber).

https://doi.org/10.1016/j.egyai.2023.100249
Received 30 September 2022; Received in revised form 28 February 2023; Accepted 28 February 2023
Available online 14 March 2023

forecast errors. We examine the marginal likelihood and forecast error for model selection for those amounts. Furthermore, we study four adaption strategies. As an extension of the current state of the art, we utilize a Bayesian linear regression for forecasting the response based on features extracted from a neural network. This approach outperforms the baseline with only seven days of training data and shows that fine-tuning is not beneficial with less than three months of data. We further show how combining multiple models through ensembles can significantly improve the model selection and adaptation approach such that we have a similar mean error with only 30 days of training data which is otherwise only possible with an entire year of training data. We achieve a mean error of 9.8 and 14 percent for the most realistic dataset for PV and wind with only seven days of training data.

1. Introduction

With the extension of volatile energy resources, such as wind and photovoltaic (PV) parks, one fundamental problem is adding new parks to an operator's portfolio. The historical data for such a new (target) park is often limited. At the same time, reliable forecasts are fundamental to assure grid stability due to weather dependency. They are, however, typically numerous pre-trained models from existing parks that we can utilize for such a forecasting task [1]. Utilizing those pre-trained source models often increases the forecast accuracy and reduces the computational effort for training a new model [2,3]. Now, the question arises: What is the best way to make use of this *model hub* of pre-trained models? The research of inductive transfer learning (ITL) provides methods for this problem [4].

Fig. 1(a) summarizes our proposed strategy. The first step is to select an appropriate source model. Recently, [4,5] showed that selecting an appropriate source model for knowledge transfer for a target substantially influences the test error for computer vision tasks. To select a source model, consider, e.g., we have two source tasks $\mathcal{T}_1$ and $\mathcal{T}_2$ of a wind park with model parameters θ_1 and θ_2. These models have a set of input observations X_1 and X_2 as well as the sets of response values Y_1 and Y_2. Based on this information, we want to select one of the models for knowledge transfer for a target task $\mathcal{T}_T$ with parameters θ_T and its respective sets X_T and Y_T.

The diagram in Fig. 1(b) visualizes this problem for wind power forecasts. In renewable power forecasts, we utilize weather forecasts, such as wind speed or radiation, from a so-called numerical weather prediction (NWP) model. These predicted weather features are the input X to machine learning (ML) models, with parameters θ, predicting the expected power generation $\mathbb{E}[p(Y|X,\theta)]$ in a *day-ahead* forecasting task between 24 and 48 h into the future. In the diagram, we can observe that the similarity S depends on the relation between the input feature wind and the power generated by a wind park, i.e., for different wind speeds and models we expect a different power generation.

Once a model is selected, the second step adapts the source knowledge with the limited target data with an *adaptation strategy*. Often this adaptation strategy is fine-tuning the final layer of a neural network. Only with such an adaptation can we make reliable and task-specific power forecasts for a new target task of a new park with limited data. To the best of our knowledge, different adaptation and selection strategies have so far not been considered for ITL in the field of renewable power forecasts. We close this gap with this article. Selecting and adapting a single source model from a model hub has the disadvantage of neglecting knowledge from other source tasks that are potentially beneficial. However, we can optionally combine models through ensemble techniques in step 3. We initially select and adapt multiple source models to the target in such an ensemble. Afterward, we combine forecasts of those target models through a weighting scheme. Such an ensemble of source models allows us to utilize knowledge from multiple parks for the target.

Since ITL has so far been insufficiently studied for renewable power forecasts [6], especially for day-ahead forecast horizons between 24 and 48 h into the future, we answer the following research questions:

Research Question 1. What is an appropriate similarity measure for model selection for a new target park from a model hub with pre-trained models?

Research Question 2. What is the best adaptation strategy once a model is selected?

Research Question 3. Are ensemble strategies – compared to selecting and adapting a single model – beneficial for combining knowledge?

Each research question directly relates to the different steps for applying transfer learning (TL) for renewable power forecasts. Our contribution lies in providing methods for each step, where each step builds upon the previous one. For step 0, we train a model hub consisting of a Bayesian extreme learning machine (BELM), a multi-layer perceptron (MLP), and a temporal convolution network (TCN) as source models on six datasets, including 667 distinct parks. To select a source model for a target in step 1, we propose using either the marginal likelihood (also known as evidence) or the normalized root mean squared error (nRMSE). Once we select a source model, we adapt the model for the target. For this 2. step, we introduce and evaluate four adaption strategies, such as fine-tuning through weight decay. After this step, we successfully adapted a model for a target with limited data. Note that we adapt each of the three source model types showing their transferability from a source to a target. We compare those models with a gradient boosting regression tree (GBRT) baseline in the first experiment. The GBRT outperforms physical models [7], which are often the fallback option for parks with limited data. We consider an additional optional third step. In this step, we combine models through ensembles and compare them to the best model from the experiment conducted for step 2. Therefore, we adapt the Bayesian model averaging (BMA) and coopetitive soft gating ensemble (CSGE) for ITL in the second experiment.

Based on these datasets and source models, our main contributions can be summarized as follows:

1. By answering the first two research questions, we provide methods and strategies that apply to a wide range of problems in research and industry that have yet to be considered for renewable power forecasts.
2. We also show, against common belief, that fine-tuning the final layer through weight decay of a neural network can be one of the worst choices for ITL in renewable power forecasts with limited data.
3. We propose BMA and the CSGE to show how ensembles outperform single models. These ensembles achieve a mean forecast error with 30 days of training data which is otherwise only possible with an entire year of training data.

The source code is open-accessible.[1] The remainder of this article is structured as follows: Section 2 describes related work and Section 3 introduces relevant definitions and details the proposed approach. We describe the datasets and discuss the experiment's most essential findings in Section 4. In the final Section 5, we summarize our work and provide insights for future work.

[1] https://github.com/scribbler00/deelea.

List of Symbols

D	Size of input features.
θ	The parameters of a linear model.
$\mathbf{X}$	All input features.
$\mathbf{y}$	All response features.
$\mathbf{x}$	A single input vector.
y	A single response respective target.
$\hat{y}$	A single response prediction.
X	The set of input features.
Y	The set of response features.
$\mathcal{X}$	Input feature space.
$\mathcal{Y}$	Output feature space.
N	The number of samples.
n	Index of samples.
$\mathbb{R}$	Set of all real numbers.
$\mathbb{R}^+$	Set of all non-negative real numbers.
$\mathbb{R}_{\geq 1}$	Set of all positive real numbers.
$\mathbb{N}_{\geq 1}$	Set of all positive natural numbers.
$\mathcal{N}$	Normal distribution.
$\mathcal{T}$	A task.
$\mathbb{T}$	The set of all tasks.
$\mathcal{S}$	Similarity measure between two task.
$\mathbf{S}$	Precision matrix of a linear model.
m	Index of a source model.
M	The number of source models.
T	The index of the target task.
$\bar{w}$	CSGE weight not normalized.
w	CSGE weight normalized.

2. Related work

In the following section we overview recent developments for TL and, more specifically, for ITL in computer vision that has not been considered for renewable power forecasts. This review determines relevant techniques that we consider for renewable power forecasts. Afterward, we summarize related work for ITL on deterministic renewable power forecasts. For additional work that utilizes a Bayesian approach in other domains of renewable energy refer to [8–10].

There are two crucial dimensions in ITL; the first is the model selection and the second is the adaptation strategy. The authors of [4,5] provide a study on selection strategies for the field of computer vision. They utilize a Bayesian linear regression (BLR) replacing the final layer of a source model and train it through empirical Bayes, also referred to as evidence approximation, on the target data. The authors repeat this approach for each available model from the model hub. Finally, they determine the similarity through the evidence of a source model on the target. As an adaptation strategy, they proposed Bayesian tuning, which regularizes the fine-tuning process by predictions from multiple sources. These proposed selections and adaptations must be considered and extended for renewable energies. For instance, we can directly forecast through the BLR and compare it with fine-tuning of the final layer. The adaptation through fine-tuning is often regularized by a weight decay regarding zero [11]. However, this regularizer does not consider parameters originating from the source model. Therefore, in [12] a deviation from a source model is penalized by weight decay considering the source model parameters.

There is limited research on ITL for renewable power forecasts compared to research areas like computer vision [6]. There has been some work to learn a transferable representation of the input utilizing autoencoders [13–16]. While transferring an autoencoder for a target is combinable with our approach, considering the conditional distribution of the power forecast is more relevant for model selection and combination. The data-driven TL approaches presented in [17,18] are outside the scope of this article.

Most of the current research on TL in renewable power forecasts focuses on meteorological measurements as input features, see [14,15,17,19–25]. These articles consider forecast horizons between ten seconds and two hours. At the same time, larger forecast horizons, such as day-ahead forecasts, are inherently more difficult as they utilize NWP as input features and forecast errors increase with an increasing forecast horizon [26].

Most of the previously mentioned related work for TL in renewable energies is treating power forecast as a regression problem. At the same time, periodic influences from, e.g., the diurnal cycle, are well-known and are not considered by regression models. Therefore, the article [3] considers time series models in a multi-task learning (MTL) architecture. Additionally, the authors of [27] consider recurrent networks and fine-tuning to achieve good results for an ultra-short-term forecast horizon of PV.

The authors of [28] achieve improvements in day-ahead PV forecasts through multi-target models. The idea of model combination is similar to our ensemble approach. However, the authors of this article do not evaluate it in the context of ITL. The study of [29] proposes an MTL strategy for Gaussian processes to forecast PV targets. By clustering wind parks, a weighting scheme provides predictions for a new park in [30]. This article uses no actual historical power measurements for evaluation; instead, the authors used synthetic data.

A number of articles apply MTL architectures for TL [3,31,32] in day-ahead forecasts. The proposed task embedding in [3,31] for MLPs and convolutional neural networks (CNNs), encodes task-specific information through an embedding to learn latent similarities between tasks. The article [3] is especially interesting as we have a similar experimental set-up.

However, the authors look at errors per season and results can be misleading as an ITL approach should avoid catastrophic forgetting for all seasons. MTL architectures are rare in the industry, e.g., due to their additional data pre-processing and training complexity. It is, therefore, essential to make the best use of existing single-task models for ITL for the extension of renewable energies.

Furthermore, no related work studies different model selection and model adaptation strategies for neural network architectures. While some work has been combining knowledge from multiple sites for PV through ensemble-like strategies, the studies are insufficient as they do not consider the available data. Also, the expected power is solely based on a characteristic curve, or authors only consider PV or wind data. We close these research gaps by providing an extensive study that overcomes those limitations for day-ahead forecasts.

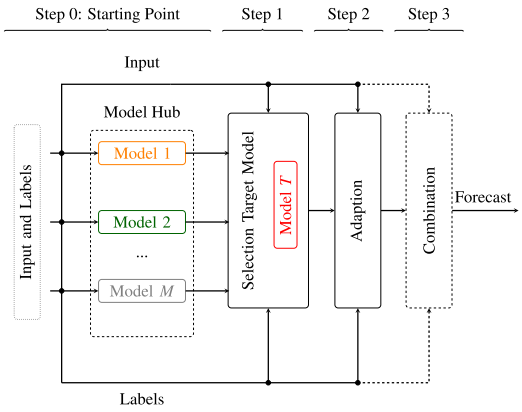
3. Proposed methods

The following sections define the proposed model selection, adaptation, and combination strategies. Beforehand we briefly summarize models for the model hub and introduce the BLR due to its central importance for one model selection and one adaption strategy.

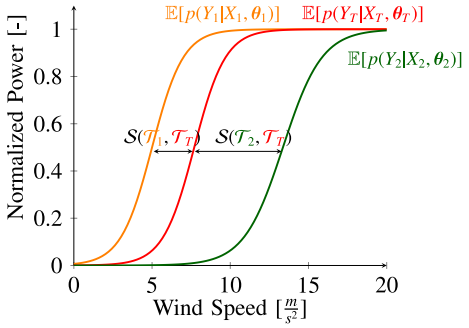
3.1. Step 0: Source models for the model hub

To provide reliable forecasts with limited data through TL, we selected the following ML models for the article: BELM, MLP, and TCN.

A BELM is an extension of linear regression and it exploits that data in a higher-dimensional space are often better linearly separable and thus facilitates prediction [33]. For this purpose, features, for example, from a NWP, are transformed into higher-dimensional space by a randomly initialized vector. At the same time, these transformed features are transformed by a nonlinear function such as Rectified Linear Unit (ReLU). These transformed features are converted to the



(a) Process of model selection and adaption for sequential transfer learning. In step 1, we initially select a source model from the model hub based on historical input and labels from the sources and the target. Afterward, in step 2, the model is adapted for the target based on limited historical data in the target. Optionally in step 3 we can combine multiple selected and adapted models through ensembles.



(b) Similarity between sources $\mathcal{T}_1$ and $\mathcal{T}_2$ with a target task $\mathcal{T}_T$ for mean wind power predictions.

Fig. 1. Diagrams illustrating the knowledge transfer based on a model hub.

corresponding label by a linear combination. In our case, this is done by a BLR, which has the advantage that we can make statements about the goodness of fit of a source model on the target data through the marginal likelihood for model selection in ITL, in comparison to using a deterministic linear regression. Due to their fast training time and convex optimization problem, BELM is a common technique for renewable power forecasts, see e.g. [19,34].

MLPs and, more recently, deep neural networks are a common technique for regression and classification tasks which we train through a gradient descent method. In an MLP, the input features are transformed by matrix multiplication and a following (usually) nonlinear function such as ReLU. The first two operations are grouped as layers, and the successive application of these layers, where the output of one layer is the input to the next layer, makes it possible to find a good representation of the data. A simple linear combination can be used for renewable power prediction in the last layer, the output layer. Due to their ability to find suitable representations of the NWP data that are easily transferable, MLPs are a common technique for renewable energy forecasting in general and TL in particular [35,36].

An extension of MLPs, which consider temporal dependencies in the data for time-series forecasts [37–39], are TCNs. The basis for a TCN is a 1-D CNN layer, which makes a convolution over data in the time dimension over each channel. In time-series problems such a channel corresponds to a feature in the input. In the recent past, 2-D CNNs were the most common technique for computer vision tasks and TL within this domain [40,41]. Recently, 1-D CNNs are of interest for transfer learning in time-series forecasts [42]. CNNs are particularly interesting for TL due to their ability to learn hierarchically, allowing us to adapt only to the last layers during fine-tuning. Within a TCNs, a single layer is replaced by a residual block. The concept of residual blocks is well-known in computer vision [43]. The principle idea behind a residual block is to add a skip connection for input from previous layers to reduce the risk of the vanishing gradient. The input is processed twice in each residual block in the following pattern: dilated convolution, weight norm, ReLU activation, and dropout for regularization. Note that dilated convolutions are special convolutional layers that increase the receptive field, are computationally efficient, and require less memory. The skip connection adds the original input to the output. An optional convolution matches the dimensions in the skip connection if a single layer's input and output dimensions are unequal.

3.2. Bayesian linear regression

Within this article the BLR is a fundamental concept:

- We require it for BELM.
- We require it to measure similarity through the marginal likelihood.
- We require it for model adaption by replacing the final layer of a neural network through a BLR.
- We require it for model combination via BMA.

Due to its central importance, we define it in detail in this section. The following definitions of a BLR make use of the introductions in [44,45]. In contrast to a deterministic perspective to learn the model weights of a linear regression model, a Bayesian approach gives additional insights through the posterior, especially when there is insufficient data [44] as for TL. It helps in measuring task similarity for model selection and allows assessing the quality of the model in terms of its uncertainty, as proposed in [4]. We propose to utilize it also for an adaptation of renewable forecasts. We achieve this by replacing the final layer of a neural network with a BLR, training it with the target data, and making predictions for the target afterward. Additionally, this approach allows combining models through BMA, see Section 3.5. Finally, we can utilize it to learn a BELM.

Let us assume that the following equation details the posterior distribution of such a linear model:

$$\underbrace{p(\boldsymbol{\theta}|X,Y)}_{\text{posterior}} = \frac{\overbrace{p(Y|X,\boldsymbol{\theta})}^{\text{likelihood}}\overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{p(Y|X)}_{\text{marginal likelihood}}}, \tag{1}$$

where $X = \{\mathbf{x}_n\}_{n=1}^{n=N}$ and $Y = \{y_n\}_{n=1}^{n=N}$ are the sets of observed input and response values with $N \in \mathbb{N}_{\geq 1}$ samples from a training dataset. In this setting, a single feature vector $\mathbf{x}_n \in \mathbb{R}^D$ has $D \in \mathbb{N}_{\geq 1}$ features and y_n is of size $\mathbb{R}$. Then the likelihood $p(Y|X,\boldsymbol{\theta})$ describes how well X and the weights $\boldsymbol{\theta} \in \mathbb{R}^D$ describe the response values. Through the prior, we encode our initial beliefs about the model weights. The marginal likelihood normalizes the posterior. Finally, after observing training data, the posterior encodes what we know about the target.

To calculate the distributions of the posterior of a linear regression model consider that we have a prior over the weights $\boldsymbol{\theta}$ with $p(\boldsymbol{\theta}|\alpha) = \mathcal{N}(\boldsymbol{\theta}|0,\alpha^{-1}\mathbf{I})$, where $\alpha \in \mathbb{R}^+$ is the precision of the zero mean isotropic Gaussian distribution. Note: Choosing an isotropic Gaussian distribution for the prior allows deriving a closed-form solution that reduces the computational effort for calculating the mean and covariance matrix.

Consider that we have a target $\mathbf{y}$ of size $1 \times N$ and $\mathbf{X}$ is the $N \times D$ design matrix, where each row corresponds to the nth observation. For multivariate problems, we can train one model per response. In our

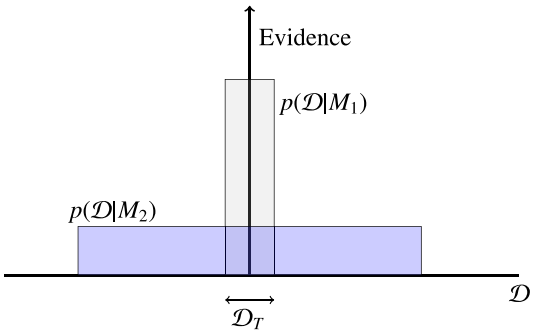


Fig. 2. Bayesian model selection.
Source: Adapted from [45].

case, $\mathbf{X}$ are either weather predictions from an NWP such as wind speed or radiation, random features from a BELM, or features extracted from a neural network at the second last layer. In the last two cases, the weather predictions are transformed through the neural network or the BELM.

Finally, the posterior distribution is given by $\mathbf{y}$, $\mathbf{X}$, and the noise precision parameter $\beta \in \mathbb{R}^+$ through $p(\boldsymbol{\theta}|\mathbf{X},\mathbf{y}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{m}_N, \mathbf{S}_N)$, where

$$\mathbf{m}_N = \beta \mathbf{S}_N \mathbf{X}^T \mathbf{y} \text{ and } \mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \mathbf{X}^T \mathbf{X}. \tag{2}$$

In this setting, N indicates the number of training samples used to update our prior beliefs of the model weights. In most cases, we are interested in predicting an unknown response $\mathbf{y}_*$ based on input $\mathbf{X}_*$ from a (test) dataset not seen during the training of the model. Therefore, the predictive posterior is defined by:

$$\begin{aligned} p(\mathbf{X}_*|\mathbf{y},\alpha,\beta) &= \int p(\mathbf{X}_*|\mathbf{y},\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{X},\mathbf{y}) \mathrm{d}\boldsymbol{\theta} \\ &= \mathcal{N}(y_*|\mathbf{X}_*\mathbf{m}_N, \sigma_N^2(\mathbf{X}_*)), \end{aligned} \tag{3}$$

where $\mathbf{y}$ and $\mathbf{X}$ are from the training set and the posterior variance is given by

$$\sigma_N^2(\mathbf{X}_*) = \beta^{-1} + \mathbf{X}_*^T \mathbf{S}_N \mathbf{X}_*. \tag{4}$$

3.3. Step 1: Model selection for inductive transfer learning

Measuring task similarity between a target task and multiple source tasks is a critical challenge in ITL [4] as it allows selecting an appropriate source task for knowledge transfer. Ideally, a valid model selection avoids negative transfer, so utilizing knowledge from the source model has a smaller error than training a target model from scratch.

Before we formalize the concept of model selection in the context of TL we will provide intuition behind (Bayesian) model selection in a broader sense. In [44] it is argued that a model selection (outside the context of TL) approach should find a trade-off between model complexity and the fit for the data. This trade-off is visualized from a Bayesian perspective in Fig. 2. On the horizontal axis, the space of all possible datasets is given. The evidence of a model for a given dataset D_T is given on the vertical axis. In this case, consider that model M_2 is a larger model with more parameters than model M_1 and, therefore, can express a larger number of datasets. We can see that with the model evidence $p(D|M_m)$ we would favor the simpler model for dataset D_T through the Bayesian perspective. The concept – that a Bayesian perspective on model selection favors the simpler model – is also known as *Occam's razor*.

The general concept of model selection is also valid in the context of TL. We aim to find a source model, from $m \in \{1,\dots,M\}$ source models with $M \in \mathbb{N}_{>1}$, that explains the limited target data T best. Consider two tasks $\mathcal{T}_1 = \{\mathcal{Y},\ P_1(Y_1 \mid X_1)\}$ and $\mathcal{T}_2 = \{\mathcal{Y},\ P_2(Y_2 \mid X_2)\}$, where the tasks $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{T}$ and $\mathbb{T}$ is the set of all possible tasks. The sets Y_m and X_m are from the response space $\mathcal{Y}$ and feature space $\mathcal{X}$. By defining a similarity measure S with $S : \mathbb{T} \times \mathbb{T} \to \mathbb{R}_{\geq 1}$, the mapping into a scalar allows making quantitative statements. For instance, given two source tasks $\mathcal{T}_1, \mathcal{T}_2$ and a target task $\mathcal{T}_T$; if $S(\mathcal{T}_1, \mathcal{T}_T) > S(\mathcal{T}_2, \mathcal{T}_T)$ then $\mathcal{T}_1$ is more similar to the target $\mathcal{T}_T$ compared to $\mathcal{T}_2$, which means that a high value implicates a high similarity. Respectively, we define dissimilarity by the inverse of a similarity measure.

The question now arises: What (similarity) measure and what kind of data should be considered to select a source model from a model hub for a specific target. One choice would be to measure similarity exclusively based on the input feature space. However, the input feature space contains limited information on the expected power generation, the response variable, in renewable power generation. For example, different amounts of energy will be produced with the same radiation for different solar modules. Consequently, we need to take the response variable into account.

3.3.1. Evidence or marginal likelihood

The authors of [4] utilize the marginal likelihood or evidence as a similarity measure S. For that purpose, the final layer of a (source) neural network $\mathcal{T}_m$ is replaced by a BLR, where the priors α and β of this model are optimized through empirical Bayes [5,44] on limited target data. In this way, the source model acts as a feature extractor. The marginal likelihood is then given by

$$\begin{aligned} S(\mathcal{T}_T, \mathcal{T}_m) &= \ln p(\mathbf{y}|\alpha,\beta) \\ &= \frac{D}{2}\ln\alpha + \frac{N}{2}\ln\beta - E(\mathbf{m}_N) - \frac{1}{2}\ln|\mathbf{S}_N^{-1}| - \frac{N}{2}\ln 2\pi, \end{aligned} \tag{5}$$

where D is the number of features, e.g., defined by the dimension of the second last layer of the neural network m with N samples and $E(\mathbf{m}_n) = \frac{\beta}{2}\cdot\|\mathbf{y}-\mathbf{X}\mathbf{m}_n\| + \frac{\alpha}{2}\mathbf{m}_n^T\mathbf{m}_n$ [5,44]. This way, we consider features extracted from the source neural network of task $\mathcal{T}_m$ and the response feature from the target $\mathcal{T}_T$. If we do this for each source model m, we can calculate the marginal likelihood of each source model on the target to calculate $S(\mathcal{T}_m, \mathcal{T}_T)$. We then select the model with the most extensive evidence as the appropriate source model. We repeat this for each dimension for multivariate problems and average the results [4].

While this approach is theoretically appealing and generalizes a broad number of problems, it has one drawback in the context of ITL: It does not consider already learned weights from the final layer of a model. This consideration is essential, as we often do not need to remove the final layer to assure compatibility between a source and a target task in renewable energies. At the same time, in contrast to a randomly initialized layer, a pre-trained layer is usually beneficial.

3.3.2. Normalized root mean-squared error

Respectively, we propose to directly measure the similarity through the nRMSE based on the pre-trained layer of a source model given by Eqs. (6) and (7).

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{i=N}(y_i^{(T)} - \hat{y}_i^{(m)})^2} \tag{6}$$

$$S(\mathcal{T}_T, \mathcal{T}_m)^{-1} = \mathrm{nRMSE} = \frac{\mathrm{RMSE} - y_{\min}}{y_{\max} - y_{\min}} \tag{7}$$

In those equations $y_i^{(T)}$ is the ith response from the target and $\hat{y}_i^{(m)}$ is the prediction from source model m on the target, $N \in \mathbb{N}_{\geq 1}$ is the number of samples, and $y_{\max}$ and $y_{\min}$ are the maximum and minimum values of the response. As a low nRMSE indicates a good similarity, we must calculate the inverse such that a large value corresponds to a large similarity. Note: For normalization of the nRMSE in all datasets, $y_{\max}$ is given by the nominal power and $y_{\min}$ is zero. We can directly measure how well a source model performs on the available target data to measure the similarity $S(\mathcal{T}_m, \mathcal{T}_T)$. Consequently, we can select the source model with a lower nRMSE on, e.g., a validation error from the target data.

Table 1
Overview of different combinations for models, selections, and adaptations. RM abbreviates the RMSE selection strategy, EV the selection through evidence, DI stands for directly applying the model, WD for fine-tuning through weight decay regarding the origin, WDS for a fine-tuning through weight decay regarding the source parameters, BT for fine-tuning with Bayesian tuning.

Model type	Selection strategy	Adaptation strategy	Abbreviation
MLP/TCN	RMSE [ours]	Direct [ours]	MLP-/TCN-RM-DI
MLP/TCN	RMSE [ours]	Weight decay [11]	MLP-/TCN-RM-WD
MLP/TCN	RMSE [ours]	Weight decay source [12]	MLP-/TCN-RM-WDS
MLP/TCN	EVIDENCE [5]	Direct [ours]	MLP-/TCN-EV-DI
MLP/TCN	EVIDENCE [5]	Direct linear [ours]	MLP-/TCN-EV-DILI
MLP/TCN	EVIDENCE [5]	Weight decay [11]	MLP-/TCN-EV-WD
MLP/TCN	EVIDENCE [5]	Weight decay source [12]	MLP-/TCN-EV-WDS
MLP/TCN	EVIDENCE [5]	Bayesian tuning [5]	MLP-/TCN-EV-BT
BELM	RMSE [ours]	Online [ours]	BELM-RM
BELM	EVIDENCE [5]	Online [ours]	BELM-EV

3.4. Step 2: Adaptation strategies for inductive transfer learning

Table 1 outlines all 18 combinations of models and adaptation strategies. As a simple TL model, we consider an *online* update of the posterior of the BELM. Therefore, the posterior from a source model acts as a prior for the target. Additionally, we evaluate *directly* applying a selected source model on the target without adapting a source model's parameter.

We also consider two standard fine-tuning methods from the field of computer vision. The first one is *weight decay* which penalizes the deviation of weights from zero and *weight decay source*, which penalizes a deviation from the source model's weights. Additionally, we examine *Bayesian tuning* as introduced in [5].

The last three adaption strategies are a type of regularization. In general, this means that we add an additional penalty term L_{pen} to the loss function L_{task} of a task through

$$L = L_{task} + \lambda \cdot L_{pen}, \tag{8}$$

where $\lambda \in \mathcal{R}$ is a hyper-parameter for the regularization typically selected by hyper-parameter optimization. L_{task} is given by

$$L_{task} = \frac{1}{N}\sum_{n=1}^{N} l(f(\mathbf{x}_n, \boldsymbol{\theta}), y_n), \tag{9}$$

where $\boldsymbol{\theta} \in \mathbb{R}^p$ and $p \in \mathbb{N}_{\geq 1}$ is a vector of the parameters we update, $\mathbf{x}_n$ is the nth input vector with $n \in N$ and $N \in \mathbb{N}_{\geq 1}$, and y_n is the respective response. For simplicity, we consider a uni-variate response here. For weight decay (WD) with respect to the origin [11], L_{pen} is then given by

$$L_{WD} = \frac{1}{2}\|\boldsymbol{\theta}\|_2^2 \tag{10}$$

To penalize a deviation from the source model, [12] proposes a weight decay w.r.t. to source weights (WDS) given by

$$L_{WDS} = \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}^0\|_2^2, \tag{11}$$

where $\boldsymbol{\theta}^0 \in \mathbb{R}^p$ is the vector of parameters from the source model before fine-tuning. Finally, in Bayesian tuning L_{pen} is given by [5]:

$$L_{Bayesian} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{M}\sum_{m=1}^{M}\mathbf{x}_{m,n}^T\boldsymbol{\theta}_{m,k} - \mathbf{x}_{t,n}^T\boldsymbol{\theta}_{t,k}\right)^2, \tag{12}$$

where $n \in N$ is the nth data sample, m is the mth source model adapted with BLR, k is the kth dimension of the response for example for different forecast horizons. $\mathbf{x}_{m,k}$ are features extracted from the mth source model, $\mathbf{x}_{t,n}$ are the respective features extracted from the target model t. $\boldsymbol{\theta}_{k,c}$ and $\boldsymbol{\theta}_{t,c}$ are the mean vectors calculated by the BLR.

3.5. Step 3: Model combination for inductive transfer learning

We discussed the model selection and adaptation strategies for a single source model for a target. However, a single model might lead to overfitting with limited data. Combining source models through an ensemble reduces this risk.

3.5.1. Bayesian model averaging

We extend the concept of [4] so that instead of choosing a single model based on the evidence, we combine models adapted through BLR by BMA. BMA is theoretically appealing as it considers the predictive posterior [46] and therefore considers the uncertainty of a model through

$$p(\mathbf{y}_{T*}|\mathbf{y}_T) = \sum_{i=1}^{i=M} p(\mathbf{y}_{T*}|\mathbf{y}_T, \boldsymbol{\theta}_{M_i})p(\boldsymbol{\theta}_{M_i}|\mathbf{y}_T). \tag{13}$$

The prior probability $p(\boldsymbol{\theta}_{M_i}|\mathbf{y}_T)$ encodes our prior belief of how similar a model M_i is to the target data set. For simplicity, we consider an equal prior for all source models. Note that we have omitted the input here to simplify notations. $p(\mathbf{y}_{T*}|\mathbf{y}_T, \boldsymbol{\theta}_{M_i})$ is the predictive posterior of a model M_i given by Eq. (3), where, e.g., the model results from the proposed *direct linear* adaptation strategy.

3.5.2. Coopetitive soft gating ensemble

We also propose to utilize the CSGE for model combination in the context of ITL. Since the CSGE can work in ensemble selection- or weighting mode, the name coopetitive is a suitcase word combining cooperation and competition. The CSGE was initially introduced for renewable power forecast in [47]. The idea of the CSGE is to link the weights to the ensemble members' performance, i.e., good source models are weighted stronger than weaker ones.

The CSGE characterizes the overall weight of a source model using three aspects:

- The *global weight* is defined by how well a source model performs with the available training data on the target task.
- The *local weight* is defined by how well a source model performs on the target tasks for different areas in the feature space. In the case of wind, for example, one model might perform well for low wind speeds, while another source model might perform well for larger wind speeds on the target.g, we laugh.
- The *forecast horizon-dependent weight* is defined by how well a source model performs for different lead times on the target task. In this case, between 24 and 48 h into the future.

Fig. 3 provides an overview of the CSGE. The CSGE includes M ensemble members, with $m \in \{1, \ldots, M\}$. Each ensemble member is a source model with a predictive function f_m. Each source model forecasts an univariate estimate $\hat{y}_{t+k|t}^{(m)} \in \mathbb{R}$ for the input $\mathbf{x}_{t+k|t} \in \mathbb{R}^D$ of a target task T. We omit the subscript T for reasons of clarity and comprehensibility. Let D be the dimension of the input feature vector $\mathbf{x}$. Then, k denotes the forecast horizon, denoted by the subscript, for the forecast origin t. For each prediction of each source model, we compute an aggregated weight $w_{t+k|t}^{(m)}$.

The weight incorporating the global w_g, local w_l, and forecast horizon-dependent weight w_h for a single source model and lead time is given by

$$\bar{w}_{t+k|t}^{(m)} = w_g^{(m)} \cdot w_l^{(m,t)} \cdot w_h^{(m,k)}, \tag{14}$$

where $\bar{w}_{t+k|t}^{(m)}$ is normalized to sum up to one to calculate $w_{t+k|t}^{(m)}$.

To calculate the weights $w_{t+k|t}^{(m)}$, we utilize the definition of the inverse similarity measurement $\mathcal{S}^{-1}$ from Section 3.3 and the coopetitive soft gating principle from Eq. (15).

$$\varsigma_\eta'(\boldsymbol{\Phi}, \phi) = \frac{\sum_{j=1}^{J}\boldsymbol{\Phi}_j}{\phi^\eta + \epsilon} \tag{15}$$

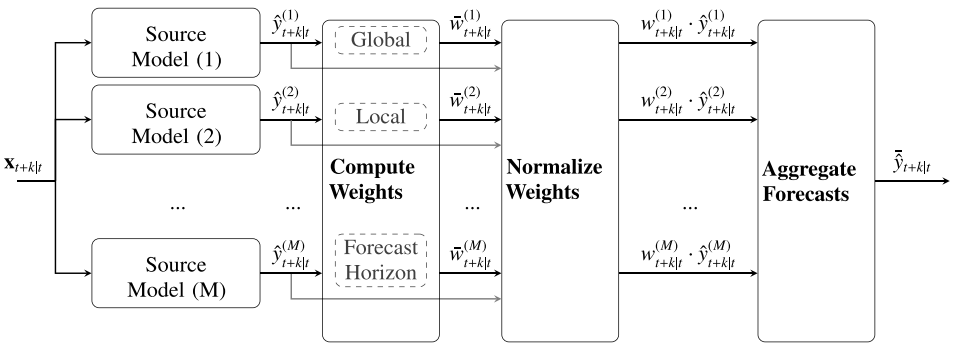


Fig. 3. The architecture of the CSGE. The source models' predictions $\hat{y}_{t+k|t}^{(m)}$ for the input $\mathbf{x}$ are passed to the CSGE. The ensemble member's weights are given by aggregating the respective global-, local- and forecast horizon-dependent weights. The weights are normalized. The source models' predictions are weighted and aggregated in the final step.

By calculating the weighting through the inverse S^{-1} (here the nRMSE), we estimate how well a source model performs on the target. Let us assume that $\mathbf{\Phi} \in \mathbb{R}^J$ contains all $J \in \mathbb{N}_{\geq 1}$ estimates based on the nRMSE and $\phi \in \mathbf{\Phi}$. Then, $\eta \geq 0$ depicts the amount of exponential weighting and the small constant $\epsilon > 0$ avoids division by zero. For greater η, the CSGE tends to work as a gating ensemble, thereby considering only a few source models. For smaller η result in a weighting ensemble. After calculating all weights from $\mathbf{\Phi}$ through Eq. (15), we normalize the results to sum up to one estimating the final weights $w_{t+k|t}^{(m)}$. This approach is repeated for each of the three weighting aspects as detailed in [47] and Appendix B.

4. Experimental evaluation

In the following Section 4.1, we summarize the experimental setup. We conduct experiments on six datasets with a total of 667 parks. Due to the utilized cross-validation, each park is once a target park. Thereby, we provide the most extensive study for ITL for renewable power forecasts.

We evaluate models through the mean performance rank, calculated across parks within a dataset, to show significant improvements against the baseline. Section 4.2 provides the details of our first experiment to answer research questions one and two. The second experiment in Section 4.3 details our findings for research question three.

4.1. Overall experimental setup

The pre-processing of the data is aligned with [3,7] to assure comparability with the current state of the art. We considered a BELM, MLP, and a TCN as source models. To have a robust baseline that generalizes well with a limited amount of data and which is known to mitigate the effects of overfitting, we trained a GBRT for each target task identical to [7].

4.1.1. Datasets

We conducted all experiments for day-ahead forecasts, between 24 to 48 h into the future. All datasets, summarized in Table 2, have NWP features as inputs, e.g., wind speed, wind direction, air pressure, or radiation. We align those weather forecasts with the historical power measurements as the response for day-ahead predictions for all datasets. These input features are weather forecasts from the European centre for medium-range weather forecasts (ECMWF) or the Icosahedral Nonhydrostatic-European Union (ICON-EU) weather model.

We have varying amounts of input features, resolutions, and different numbers of samples for training and testing in all datasets. For instance, the PVOPEN has 47 features, where various manually engineered features take seasonal patterns of the sun into account. In contrast, these manually engineered features are not included in other datasets.

Table 2
Overview of the evaluated datasets.

Dataset	#parks	#features	#train samples	#mean samples	resolution	NWP model
PVOPEN [3]	21	47	6336	8424	Hourly	ECMWF
PVSYN [7]	114	20	30 385	14 920	15-min	ICON-EU
PVREAL	42	25	58 052	19 344	15-min	ICON-EU
WINDOPEN [3]	45	13	27 724	26 636	15-min	ECMWF
WINDSYN [7]	260	29	33 714	16 678	15-min	ICON-EU
WINDREAL	185	33	36 129	12 092	15-min	ICON-EU

Note that four datasets, the PVOPEN, WINDOPEN, WINDSYN, and PVSYN have already been investigated, see e.g. [3,7]. This is not the case for WINDREAL and PVREAL. These two datasets are not publicly available. They are, however, the most realistic datasets due to their diversity. The WINDREAL dataset comprises 99 nominal capacities, 13 turbine manufacturers, and six hub heights. All parks are located in Germany. PV power plants in the PVREAL dataset have 31 different nominal capacities, ten tilt orientations, and nine azimuth orientations and are also located in Germany. It is also important to note that forecasting the expected power generation from wind parks is more challenging than for PV parks. For additional insights on the challenges and the datasets refer to Appendix A.

Each dataset was split through five-fold cross-validation so that *each park is once a target task* and four times a source park. We trained source models and their hyperparameters on the training and validation data. We split the training into the four seasons for training target models and limited the training data to $7, 14, 30, 60$ or 90 days of training data, respectively. The presented results are mean values for all tasks and seasons. This setup assures that results are not biased by seasonality [26]. All input features of all datasets are normalized. We normalized the historical power by the nominal power to make errors comparable. We resampled all datasets to have a 15-minute resolution except the PVOPEN dataset, which we resampled for an hourly resolution due to the low initial resolution. A predefined test set is given for the WINDSYN and PVSYN datasets. In the case of the WINDOPEN and PVOPEN, we used the first year's data as training data and the remaining data as test data, identical to [3]. Due to this diversity in the number of historical power measurements for the PVREAL and WINDREAL datasets, 25% randomly sampled days are considered test data. As each day is based on an independent day ahead NWP forecasts, no information is leaked from the future to the past [34]. We use 25% of the remaining days for validation and the rest for training.

4.1.2. Source models

As pointed out earlier, due to the weather dependency for renewable power forecasts, the input features of the models are themselves forecasts from the NWP model. Respectively, we can directly utilize those to train, e.g., an MLP to forecast the expected power of the next

Table 3
Rank summary for all models, selections, and adaptation strategies on the PV datasets, cf. Table 1. Only those within the top ranks for a dataset are included. GBRT is the baseline and all models are tested if the forecast error is significantly ($\alpha = 0,01$) better (∨), worse (∧), or not significantly different (⋄). We conduct this hypothesis test for all parks within a dataset for the given number of days of training data. The colors denote the respective rank. Blue indicates a smaller (better) rank and red a higher (worse) rank.

Data type	#Days	Baseline	BELM-EV	BELM-RM	MLP-EV-DILI	MLP-RM-DI	TCN-EV-BT	TCN-EV-DI	TCN-EV-DILI	TCN-EV-WD	TCN-RM-DI
PVOPEN	7	6.631	**3.524**∨	4.107∨	6.036⋄	4.107∨	6.845⋄	5.679∨	5.179∨	6.81⋄	4.369∨
PVREAL	7	8.786	5.173∨	4.821∨	6.012∨	4.423∨	5.518∨	5.048∨	4.464∨	5.25∨	**3.548**∨
PVSYN	7	8.636	4.649∨	4.263∨	6.928∨	3.866∨	6.015∨	4.002∨	5.189∨	5.719∨	**3.482**∨
PVOPEN	14	6.143	3.738∨	**3.357**∨	5.298∨	4.155∨	7.524∧	5.857⋄	5.298⋄	7.417∧	4.726∨
PVREAL	14	8.619	5.125∨	4.744∨	5.101∨	4.137∨	6.268∨	5.351∨	4.119∨	6.19∨	**3.345**∨
PVSYN	14	8.252	4.691∨	4.408∨	6.171∨	3.75∨	6.904∨	3.936∨	4.329∨	6.785∨	**3.217**∨
PVOPEN	30	5.619	**3.464**∨	3.583∨	5.702⋄	4.536∨	7.0∧	6.452⋄	4.952⋄	6.881∧	5.286⋄
PVREAL	30	8.417	5.423∨	5.095∨	3.875∨	4.25∨	5.875∨	6.214∨	**3.631**∨	6.048∨	4.018∨
PVSYN	30	8.015	4.93∨	4.544∨	6.14∨	4.05∨	6.432∨	4.268∨	4.042∨	6.279∨	**3.542**∨
PVOPEN	60	5.44	**3.726**∨	3.845∨	5.321⋄	4.952⋄	6.762∧	6.94∧	4.679⋄	6.655∧	5.179⋄
PVREAL	60	8.095	5.726∨	5.244∨	4.286∨	4.565∨	5.786∨	5.732∨	**3.452**∨	5.613∨	3.792∨
PVSYN	60	7.509	4.886∨	4.733∨	6.401∨	4.22∨	6.312∨	4.13∨	3.978∨	6.267∨	**3.608**∨
PVOPEN	90	5.0	**3.512**∨	4.167⋄	5.393⋄	4.524⋄	6.976∧	6.69∧	5.345⋄	6.869∧	5.095⋄
PVREAL	90	8.077	5.815∨	5.173∨	4.107∨	4.833∨	5.452∨	5.661∨	**3.685**∨	5.661∨	3.905∨
PVSYN	90	7.192	5.1∨	5.076∨	6.508∨	4.327∨	5.986∨	4.516∨	3.8∨	6.092∨	**3.749**∨

Table 4
Rank summary for all source models, selections, and adaptation strategies on the wind datasets. Cf. Tables 1 and 3.

WINDOPEN	7	7.387	4.012∨	5.526∨	6.734∨	4.861∨	5.382∨	6.399∨	4.532∨	5.133∨	**3.832**∨
WINDREAL	7	8.476	4.478∨	4.305∨	6.807∨	4.252∨	5.469∨	6.152∨	5.103∨	5.378∨	**3.407**∨
WINDSYN	7	7.863	5.165∨	4.096∨	6.644∨	4.264∨	5.382∨	6.305∨	5.183∨	5.544∨	**3.688**∨
WINDOPEN	14	6.671	4.422∨	5.85∨	6.306⋄	4.919∨	5.566∨	6.387⋄	3.965∨	5.468∨	**3.78**∨
WINDREAL	14	7.892	4.661∨	4.872∨	6.223∨	4.368∨	5.933∨	6.248∨	4.048∨	5.905∨	**3.386**∨
WINDSYN	14	7.601	4.847∨	4.516∨	5.892∨	4.29∨	5.997∨	6.472∨	4.365∨	6.062∨	**3.684**∨
WINDOPEN	30	5.156	5.527⋄	6.365∧	6.892∧	5.048⋄	5.407⋄	6.246∧	**3.856**∨	5.186⋄	3.862∨
WINDREAL	30	6.848	5.352∨	5.779∨	5.918∨	4.806∨	5.525∨	6.514∨	**3.566**∨	5.574∨	3.589∨
WINDSYN	30	6.669	4.812∨	4.926∨	5.447∨	4.833∨	5.864∨	6.827⋄	4.081∨	6.108∨	**4.049**∨
WINDOPEN	60	4.25	5.974∧	6.467∧	6.711∧	5.414∧	5.421∧	6.789∧	**3.289**∨	5.309∧	4.0⋄
WINDREAL	60	5.665	5.804⋄	6.052∧	5.819⋄	4.958∨	5.646⋄	6.758∧	**3.58**∨	5.628⋄	3.801∨
WINDSYN	60	5.947	5.116∨	5.304∨	5.614∨	4.973∨	5.59∨	7.203∧	**3.891**∨	5.724⋄	4.351∨
WINDOPEN	90	3.992	6.289∧	6.969∧	7.07∧	5.719∧	4.648⋄	6.406∧	**3.516**⋄	4.805∧	3.969⋄
WINDREAL	90	5.017	5.876∧	6.212∧	6.071∧	5.143⋄	5.569∧	6.944∧	**3.463**∨	5.604∧	3.836∨
WINDSYN	90	5.008	5.212⋄	5.394∧	5.478∧	5.281∧	5.701∧	7.359∧	**3.88**∨	5.879∧	4.518∨

day. To optimize the hyperparameters of those models, we utilize a tree-structured Parzen sampler for 200 samples on the validation data. Details of the chosen hyperparameters are provided in Appendix C.

We train four kinds of models in total. The trained BELM is particularly interesting as a source model because it can directly measure similarity by the evidence and has a linear increase in time for updating the model. We train an MLP as it is common practice in the renewable power forecast industry [3]. To account for cyclic behavior within the forecast, we also train a TCN architecture, similar to [3]. To have a strong baseline that generalizes well we trained a GBRT [7].

4.1.3. Evaluation method

We calculated the error on the test dataset through the nRMSE through Eq. (7) for all combinations of seasons and available training data. For a given dataset, season, and the number of days of training data, we calculated the mean performance rank based on the nRMSE. We test for a significant improvement compared to the baseline by the Wilcoxon test ($\alpha = 0.01$) across all parks within a dataset.

4.2. Experiment on model selection and model adaptation

This section conducts an experiment to answer research questions one and two simultaneously as a model selection technique can only be evaluated after the adaptation:

Research Question 1. What is an appropriate similarity measure for model selection for a new target park from a model hub with pre-trained models?

Research Question 2. What is the best adaptation strategy once a model is selected?

4.2.1. Findings questions 1 & 2

Model selection and adaptation strategies highly influence each other. With limited training data (between 7 and 30 days), selecting a model based on the forecast error with no adaptation has one of the best results. Replacing the final layer with a BLR is superior with additional data. None of the fine-tuning methods are among the best models.

4.2.2. Experimental setup

The source models are those detailed in Section 4.1.2. As adaptation strategies, we consider those mentioned in Section 3.4. For fine-tuning, we train for a single epoch and optimize hyperparameters through grid search on 30% of the available target data.

For the weight decay adaptation, we optimize seven logarithmically spaced learning rates between 10^{-1} and 10^{-4}, similar to [4]. We take seven grid points for the amount of penalty λ in the logarithmic space between 10^{-6} to 10^{-3}, similar to [4]. We use the same learning rate for the Bayesian tuning and weight decay source. The amount of penalty λ for the L_{Bayesian} loss is one of $[0.1, 0.25, 0.5, 1, 2, 4, 8]$. λ is one of $[1, 0.1]$ for the weight decay source. Note that we shuffle the data during training. Hyperparameter optimization is not required for other approaches.

4.2.3. Detailed findings

Results of the best techniques are summarized in Tables 3 and 4. We only show models appearing at least once within the top four ranks for a dataset. The BELM is among the best models and outperforms the baseline up to 30 days of training data. With less or equal to 14 days of training data, it seems beneficial to directly utilize a model without any model adaptation. Starting with 30 days of training data utilizing a BLR trained on extracted features from the source model and the historical power from the target is beneficial, especially for

Table 5
Rank summary of ensembles on the PV datasets. The best model, the TCN-EV-DILI, from the experiment in Section 4.2 is the baseline. Cf. Tables 1 and 3.

Data type	#Days	Baseline	BMA-BELM	BMA-MLP	BMA-TCN	CSGE-MLP-DI	CSGE-MLP-DILI	CSGE-MLP-DILI-GBRT	CSGE-TCN-DI	CSGE-TCN-DILI	CSGE-TCN-DILI-GBRT
PVOPEN	7	7.381	**3.143**∨	3.643∨	4.167∨	4.429∨	5.702∨	7.571⋄	4.952∨	5.667∨	7.476⋄
PVREAL	7	6.524	5.084∨	4.187∨	**3.807**∨	4.506∨	5.91∨	8.367∧	4.066∨	4.602∨	7.904∧
PVSYN	7	5.053	6.007∧	6.031∧	5.921∧	3.739∨	5.254⋄	7.805∧	**3.596**∨	4.447∨	7.031∧
PVOPEN	14	7.512	3.643∨	**3.512**∨	4.333∨	4.857∨	5.262∨	7.298⋄	5.071∨	5.5∨	7.571⋄
PVREAL	14	6.405	5.869⋄	4.149∨	**3.905**∨	5.155∨	5.161∨	7.827∧	4.714∨	4.536∨	7.268∧
PVSYN	14	4.353	6.943∧	6.417∧	6.566∧	4.566⋄	4.542⋄	6.969∧	4.246⋄	**3.998**⋄	6.346∧
PVOPEN	30	7.607	3.964∨	**3.476**∨	4.31∨	5.655∨	4.94∨	6.238∨	6.274∨	5.69∨	6.393∨
PVREAL	30	6.476	7.071⋄	3.929∨	**3.899**∨	6.208⋄	4.369∨	6.643⋄	5.673∨	4.214∨	6.518⋄
PVSYN	30	**3.789**	7.625∧	7.138∧	7.478∧	5.548∧	4.075⋄	5.226∧	5.213∧	3.958⋄	4.943∧
PVOPEN	60	7.548	4.214∨	**3.607**∨	4.679∨	5.929∨	4.881∨	5.833∨	6.464∨	5.488∨	6.262∨
PVREAL	60	6.464	7.565∧	4.131∨	4.542∨	6.44⋄	4.077∨	5.988⋄	6.125⋄	**4.065**∨	5.601∨
PVSYN	60	**3.623**	7.401∧	7.554∧	7.879∧	5.776∧	4.426∧	4.455∧	5.404∧	4.229∧	4.253∧
PVOPEN	90	8.131	3.964∨	**3.571**∨	4.833∨	6.119∨	4.833∨	4.94∨	6.631∨	6.048∨	5.857∨
PVREAL	90	6.631	7.768∧	4.31∨	4.81∨	6.482⋄	**4.024**∨	5.113∨	6.244⋄	4.458∨	5.161∨
PVSYN	90	**3.714**	7.451∧	7.8∧	8.281∧	5.719∧	4.116∧	4.116⋄	5.495∧	4.181∧	4.127∧

Table 6
Rank summary of ensembles on the wind datasets. Cf. Table 5.

Data type	#Days	Baseline	BMA-BELM	BMA-MLP	BMA-TCN	CSGE-MLP-DI	CSGE-MLP-DILI	CSGE-MLP-DILI-GBRT	CSGE-TCN-DI	CSGE-TCN-DILI	CSGE-TCN-DILI-GBRT
WINDOPEN	7	5.817	4.787∨	4.065∨	**3.598**∨	5.657⋄	6.254⋄	7.751∧	5.296⋄	4.586∨	7.006∧
WINDREAL	7	6.363	5.756∨	4.975∨	**3.71**∨	4.724∨	5.36∨	7.926∧	5.068∨	4.079∨	6.932∧
WINDSYN	7	5.81	5.54⋄	5.046∨	**3.848**∨	5.193∨	5.88⋄	7.595∧	5.222∨	4.313∨	6.401∧
WINDOPEN	14	5.633	5.521⋄	3.917∨	**3.74**∨	6.639∧	5.373⋄	6.976∧	6.734∧	4.148∨	6.195⋄
WINDREAL	14	6.184	7.077∧	4.705∨	3.919∨	6.066⋄	4.504∨	6.582∧	6.44⋄	**3.807**∨	5.686∨
WINDSYN	14	5.218	6.681∧	4.7∨	4.232∨	6.637∧	4.957∨	6.2∧	6.881∧	**4.085**∨	5.302⋄
WINDOPEN	30	5.221	6.209∧	**4.074**∨	4.368∨	7.669∧	5.184⋄	5.301⋄	7.65∧	4.595⋄	4.669⋄
WINDREAL	30	5.699	7.646∧	4.927∨	4.512∨	6.753∧	4.275∨	5.186∨	7.17∧	**4.113**∨	4.71∨
WINDSYN	30	5.061	7.478∧	4.683∨	4.596∨	7.477∧	4.365∨	4.877⋄	7.683∧	**4.219**∨	4.53∨
WINDOPEN	60	4.966	6.527∧	4.791⋄	5.014⋄	7.912∧	4.831⋄	4.493⋄	8.182∧	4.446⋄	**3.838**∨
WINDREAL	60	5.757	7.894∧	5.225∨	4.882∨	7.295∧	4.182∨	4.052∨	7.613∧	4.234∨	**3.865**∨
WINDSYN	60	4.468	7.216∧	5.569∧	5.212∧	7.791∧	4.453⋄	4.177∨	7.766∧	4.401⋄	**3.945**∨
WINDOPEN	90	4.984	6.685∧	4.694⋄	5.161⋄	8.282∧	4.919⋄	4.089∨	8.185∧	4.484⋄	**3.516**∨
WINDREAL	90	5.656	7.979∧	5.307⋄	5.097∨	7.376∧	4.181∨	**3.633**∨	7.697∧	4.403∨	3.66∨
WINDSYN	90	4.627	7.405∧	5.602∧	5.599∧	8.087∧	4.131∨	**3.514**∨	8.009∧	4.437⋄	3.589∨

Table 7
Mean nRMSE of ensembles on the PV datasets. The best model, the TCN-EV-DILI, from the experiment in Section 4.2 is the baseline. Cf. Tables 1 and 3.

Data type	#Days	Base-line	BMA-BELM	BMA-MLP	BMA-TCN	CSGE-MLP-DI	CSGE-MLP-DILI	CSGE-MLP -DILI-GBRT	CSGE-TCN-DI	CSGE-TCN-DILI	CSGE-TCN-DILI-GBRT
PVOPEN	7	0.087	**0.07**∨	0.075∨	0.074∨	0.073∨	0.079∨	0.083⋄	0.073∨	0.079∨	0.084⋄
PVREAL	7	0.109	0.1∨	0.101∨	0.1∨	0.099∨	0.106∨	0.119∧	**0.098**∨	0.102∨	0.117∧
PVSYN	7	0.1	0.096∧	0.101∧	0.098∧	**0.09**∨	0.101⋄	0.109∧	**0.09**∨	0.097∨	0.106∧
PVOPEN	14	0.084	**0.07**∨	0.073∨	0.073∨	0.072∨	0.075∨	0.081⋄	0.072∨	0.075∨	0.081⋄
PVREAL	14	0.103	0.1⋄	0.099∨	0.098∨	**0.097**∨	0.1∨	0.107∧	**0.097**∨	0.099∨	0.106∧
PVSYN	14	0.089	0.095∧	0.094∧	0.094∧	0.088⋄	0.089⋄	0.097∧	**0.087**⋄	0.088⋄	0.096∧
PVOPEN	30	0.077	**0.069**∨	0.071∨	0.072∨	0.072∨	0.072∨	0.075∨	0.072∨	0.073∨	0.075∨
PVREAL	30	0.097	0.1⋄	**0.094**∨	**0.094**∨	0.097⋄	**0.094**∨	0.099⋄	0.096∨	**0.094**∨	0.099⋄
PVSYN	30	**0.084**	0.094∧	0.091∧	0.092∧	0.086∧	**0.084**⋄	0.087∧	0.085∧	**0.084**⋄	0.087∧
PVOPEN	60	0.076	**0.068**∨	0.07∨	0.07∨	0.071∨	0.071∨	0.072∨	0.072∨	0.071∨	0.073∨
PVREAL	60	0.095	0.099∧	**0.093**∨	0.094∨	0.096⋄	**0.093**∨	0.096⋄	0.096⋄	**0.093**∨	0.096∨
PVSYN	60	**0.082**	0.092∧	0.091∧	0.092∧	0.085∧	0.084∧	0.084∧	0.085∧	0.084∧	0.084∧
PVOPEN	90	0.077	**0.068**∨	0.07∨	0.07∨	0.071∨	0.071∨	0.071∨	0.072∨	0.072∨	0.072∨
PVREAL	90	0.096	0.098∧	**0.093**∨	0.094∨	0.096⋄	**0.093**∨	0.094∨	0.096⋄	0.094∨	0.095∨
PVSYN	90	**0.081**	0.091∧	0.091∧	0.091∧	0.084∧	0.082∧	0.083⋄	0.084∧	0.082∧	0.083∧

WINDREAL and PVREAL. This effect occurs as features extracted from a single model from a single prediction task do not generalize well enough for other parks. Therefore, sufficient data is required to train the BLR to compensate for differences between a source and a target park.

We conclude from these observations with two critical considerations for real-world applications. First, due to the learning procedure of gradient descent, there is a high risk of catastrophic forgetting that should be avoided for model hubs in safety-critical areas such as renewable power forecasts. Second, the BLR gives rise to optimal training due to the convex optimization problem, which reduces the risk of catastrophic forgetting. Neither a weight decay nor the Bayesian tuning adaptation strategy is within the best models in the evaluated scenarios. This observation is surprising as this fine-tuning approach is common in various domains. However, due to the source model's training on a single park approach, there is a high risk that even the best-selected source model causes catastrophic forgetting as the model is too specific. For instance, catastrophic forgetting may appear due to slightly different weather conditions or physical characteristics such as the turbine type.

An additional study in Appendix C shows that for fine-tuning techniques, the evidence selection strategy is superior for the TCN model regardless of the adaptation strategy. A selection through the MLP is preferable for the nRMSE. Most likely, the probabilistic approach of the evidence and, therefore, the more comprehensive treatment of similarity better captures the correlations between source and target

Table 8
Mean nRMSE of ensembles on the wind datasets. Cf. Table 7.

Data type	#Days	Base-line	BMA-BELM	BMA-MLP	BMA-TCN	CSGE-MLP-DI	CSGE-MLP-DILI	CSGE-MLP-DILI-GBRT	CSGE-TCN-DI	CSGE-TCN-DILI	CSGE-TCN-DILI-GBRT
WINDOPEN	7	0.176	0.17∨	0.166∨	**0.163**∨	0.167⋄	0.172⋄	0.187∧	0.167⋄	0.166∨	0.184∧
WINDREAL	7	0.152	0.151∨	0.147∨	**0.14**∨	0.142∨	0.148∨	0.166∧	0.142∨	0.141∨	0.161∧
WINDSYN	7	0.184	0.191⋄	0.174∨	**0.167**∨	0.174∨	0.178⋄	0.193∧	0.174∨	0.169∨	0.186∧
WINDOPEN	14	0.165	0.167⋄	0.16∨	**0.157**∨	0.166∧	0.162⋄	0.173∧	0.166∧	0.158∨	0.169⋄
WINDREAL	14	0.14	0.149∧	0.136∨	**0.133**∨	0.14⋄	0.137∨	0.145∧	0.14⋄	**0.133**∨	0.141∨
WINDSYN	14	0.163	0.187∧	0.161∨	0.158∨	0.17∧	0.162∨	0.167∧	0.171∧	**0.157**∨	0.163⋄
WINDOPEN	30	0.158	0.164∧	**0.153**∨	0.154∨	0.165∧	0.154⋄	0.156⋄	0.165∧	**0.153**⋄	0.155⋄
WINDREAL	30	0.134	0.146∧	0.133∨	**0.131**∨	0.139∧	0.132∨	0.135∨	0.139∧	**0.131**∨	0.133∨
WINDSYN	30	0.153	0.182∧	0.153∨	0.152∨	0.166∧	0.151∨	0.154⋄	0.166∧	**0.15**∨	0.152∨
WINDOPEN	60	0.146	0.156∧	0.146⋄	0.146⋄	0.158∧	0.145⋄	0.145⋄	0.158∧	0.145⋄	**0.144**∨
WINDREAL	60	0.133	0.143∧	0.132∨	0.131∨	0.139∧	0.131∨	0.131∨	0.139∧	**0.13**∨	**0.13**∨
WINDSYN	60	0.15	0.176∧	0.153∧	0.153∧	0.165∧	0.15⋄	0.15∨	0.165∧	0.15⋄	**0.149**∨
WINDOPEN	90	0.141	0.15∧	0.14⋄	0.141⋄	0.152∧	0.14⋄	0.139∨	0.152∧	0.14⋄	**0.138**∨
WINDREAL	90	0.132	0.14∧	0.13⋄	0.13∨	0.138∧	**0.129**∨	**0.129**∨	0.138∧	**0.129**∨	**0.129**∨
WINDSYN	90	0.147	0.173∧	0.15∧	0.151∧	0.162∧	0.147∨	**0.146**∨	0.161∧	0.147⋄	**0.146**∨

for the convolutional layers in the TCN. To update the weights of the final layer of the MLP a selection through the nRMSE is sufficient.

4.3. Experiment on model combination

In this section, we conduct an experiment to answer research question three.

Research Question 3. Are ensemble strategies – compared to selecting and adapting a single model – beneficial for combining knowledge?

4.3.1. Findings research question 3

Ensembles improve results from the previous experiment significantly. An approach utilizing BMA is preferable for more straightforward problems. An approach by the CSGE is superior for more complex scenarios.

4.3.2. Experimental setup

TCN-EV-DILI from the experiment in Section 4.2 is the baseline. For the BMA, we first update all source models based on available target data as previously described. For the MLP and TCN source models we replace the final layer through BLR model(s), as described for the direct linear adaptation. After this adaptation for the target, each model provides a predictive posterior distribution according to Eq. (3) that is combined by BMA with Eq. (13). We consider three variants for the BMA, one for each of the three source model types.

For the CSGE, we calculate the global and forecast horizon-dependent error based on the nRMSE. We estimate the local error through a k-nearest neighbor approach. Therefore, we first reduce the dimension of the feature space through principal component analysis (PCA) to two components. We consider three neighbors within this reduced feature space to estimate the local error in the feature space. The hyperparameter η is selected as either one or two through grid search. We also optimize the learning rate from the set {0.5, 0.1, 1×10^{-3}, 1×10^{-5}}. In total, we consider six variants of the CSGE: Two for the MLP and TCN model where the source models are not updated for the target (CSGE-MLP-DI/CSGE-TCN-DI), two variants, where the final layer of the MLP and TCN source models are updated through BLR (CSGE-MLP-DILI/CSGE-TCN-DILI), and these two variants are extended, where we utilize the GBRT as an additional source model (CSGE-MLP-DILI-GBRT/CSGE-TCN-DILI-GBRT).

4.3.3. Detailed findings

Results are summarized in Tables 5 and 6. For the PV datasets, the best CSGE variants outperform the baseline in almost all cases. At the same time, the BMA achieves excellent results for the PVOPEN dataset and the PVREAL dataset for up to 30 days of training data. With minimal data (less than 30 days) the BMA is among the best for the wind datasets. With more training data, the CSGE with TCN source models, where the final layer is replaced by BLR, is the best. For these datasets we can also observe that additionally considering the GBRT as the source model improves the results.

This observation also shows the flexibility of the CSGE. Due to the combination through the forecast error, we can combine arbitrary models. This flexibility is not given by the proposed BMA approach. However, the BMA has the advantage that probabilistic forecasts are provided, which is not this article's focus.

Another important consideration is that in almost all cases, the ensemble techniques outperform the baseline, which is the best model from the previous experiment. These results show that a single source model's selection and adaptation process is highly uncertain because the model may be too specific for the target. Selecting and adapting a single source model for the target is challenging due to specific characteristics of a single model — the weather at the location or technical factors, for example. In contrast, combining several models balances individual properties and improves the error significantly.

Besides the previous statistical discussion through the mean performance ranking, we must also include an analysis of the forecast error for real-world implications. Therefore, the mean nRMSE is summarized for this experiment in Tables 7 and 8. The best model from the previous experiment is again the baseline.

In these tables, the error of the models decreases with increasing training data amount for all six datasets. For the PV datasets, the best model has the largest error for the PVREAL dataset. The best forecast error for this dataset is only 9.8 percent with seven days of training data. For the PVOPEN with seven days of training data, the error is with 7 percent error rate lower than results from [3].

Also, for the WINDOPEN dataset, the best models have similar error rates, between 16.3 for seven days and 13.8 percent for 90 days of training data, similar to the result in [3]. The WINDSYN dataset has the largest errors, between 16.7 and 14.6 percent, for the wind datasets.

Based on the analysis of the nRMSE, we can observe that even with a small amount of training data, good up to excellent prediction quality can be achieved. Furthermore, the mean nRMSE with more than 30 days often corresponds to error rates with a whole year of training data [3,7].

5. Conclusion and future work

We successfully evaluated several combinations of models, model selection, adaptation strategies, and two combination strategies on six datasets. Our study's exhaustive evaluation is the most extensive for transfer learning utilizing a model hub in renewable power forecasts on real-world datasets.

We found that fine-tuning the final layer of a neural network, a well-known strategy, does not lead to convincing results in this setting.

Instead, replacing the layer with a Bayesian linear regression model trained with features extracted from the source and limited power measurements from the target task yields one of the best results, especially for a temporal convolutional neural network. This result is best explained in comparison to computer vision tasks, where tasks are typically trained on many variations, e.g., various classification tasks, which helps in generalization. In contrast, renewable energy models are often trained on a single forecasting task. This approach with limited variations generalizes insufficiently for fine-tuning.

We suggest utilizing the forecast error with less than 30 days of training data for source model selection; the evidence is recommended with additional data. We also showed how combining models leads to further significant improvements compared to considering a single model. The proposed coopetitive soft-gating ensemble combines source models based on the error of the target. Our suggestion to utilize the Bayesian model averaging as an ensemble strategy is beneficial for minimal historical data.

To overcome the shortcomings of fine-tuning caused by limited data, we aim to augment the target data with synthetic data in the future. Likewise, we will expand our analysis for multi-task problems.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jens Schreiber reports financial support was provided by German Federal Ministry of Education and Research.

Data availability

Four out of six datasets are openly accessible. Two cannot be made publicly available. Source code is provided on GitHub.

Acknowledgments

This work results from the project TRANSFER (01IS20020B) funded by BMBF (German Federal Ministry of Education and Research). We thank enercast GmbH for providing the PVREAL and WINDREAL datasets. We also thank Marek Herde, Mohammad Wazed Ali, and David Meier for their valuable input.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.egyai.2023.100249.

References

[1] Schreiber J. Transfer learning in the field of renewable energies - a transfer learning framework providing power forecasts throughout the lifecycle of wind farms after initial connection to the electrical grid. In: Organic computing - doctoral dissertation colloquium. kassel university press GmbH; 2019, p. 75–87.

[2] Schwartz R, Dodge J, Smith NA, et al. Green AI. 2019, p. 1–12, CoRR arXiv:1907.10597.

[3] Schreiber J, Vogt S, Sick B. Task embedding temporal convolution networks for transfer learning problems in renewable power time-series forecast. In: ECML. 2021, p. 1–16. http://dx.doi.org/10.1007/978-3-030-86514-6_8.

[4] You K, Liu Y, Wang J, et al. Logme: Practical assessment of pre-trained models for transfer learning. In: ICML. 2021, p. 12133–43, arXiv:2102.11005.

[5] You K, Liu Y, Wang J, et al. Ranking and tuning pre-trained models: A new paradigm of exploiting model hubs. 2021, p. 1–45. http://dx.doi.org/10.48550/arXiv.2110.10545, CoRR arXiv:2110.10545.

[6] Alkhayat G, Mehmood R. A review and taxonomy of wind and solar energy forecasting methods based on deep learning. Energy AI 2021;4:100060. http://dx.doi.org/10.1016/j.egyai.2021.100060.

[7] Vogt S, Schreiber J. Synthetic photovoltaic and wind power forecasting data. 2022, CoRR arXiv:2204.00411.

[8] Zheng X-W, Li H-N, Gardoni P. Hybrid Bayesian-copula-based risk assessment for tall buildings subject to wind loads considering various uncertainties. Reliab Eng Syst Saf 2023;233:109100. http://dx.doi.org/10.1016/j.ress.2023.109100.

[9] Alruqi M, Sharma P, Deepanraj B, Shaik F. Renewable energy approach towards powering the CI engine with ternary blends of algal biodiesel-diesel-diethyl ether: Bayesian optimized Gaussian process regression for modeling-optimization. Fuel 2023;334:126827. http://dx.doi.org/10.1016/j.fuel.2022.126827.

[10] Said Z, Sharma P, Syam Sundar L, Nguyen VG, Tran VD, Le VV. Using Bayesian optimization and ensemble boosted regression trees for optimizing thermal performance of solar flat plate collector under thermosyphon condition employing MWCNT-Fe3o4/water hybrid nanofluids. Sustain Energy Technol Assess 2022;53:102708. http://dx.doi.org/10.1016/j.seta.2022.102708.

[11] Li H, Chaudhari P, Yang H, et al. Rethinking the hyperparameters for fine-tuning. In: ICLR. 2020, p. 1–20, URL http://arxiv.org/abs/2002.11770. arXiv:2002.11770.

[12] Li X, Grandvalet Y, Davoine F. Explicit inductive bias for transfer learning with convolutional networks. In: ICML, Vol. 6. 2018, p. 4408–19, arXiv:1802.01483.

[13] Qureshi AS, Khan A. Adaptive transfer learning in deep neural networks: Wind power prediction using knowledge transfer from region to region and between different task domains. Comput Intell 2019;35(4):1088–112. http://dx.doi.org/10.1111/coin.12236.

[14] Liu X, Cao Z, Zhang Z. Short-term predictions of multiple wind turbine power outputs based on deep neural networks with transfer learning. Energy 2021;217:119356. http://dx.doi.org/10.1016/j.energy.2020.119356.

[15] Ju Y, Li J, Sun G. Ultra-short-term photovoltaic power prediction based on self-attention mechanism and multi-task learning. IEEE Access 2020;8:44821–9. http://dx.doi.org/10.1109/access.2020.2978635.

[16] Henze J, Schreiber J, Sick B. Representation learning in power time series forecasting. In: Deep learning: algorithms and applications. Springer, Cham; 2020, p. 67–101. http://dx.doi.org/10.1007/978-3-030-31760-7_3.

[17] Cao L, Wang L, Huang C, Luo X, Wang J-H. A transfer learning strategy for short-term wind power forecasting. In: Chinese automation congress. 2018, p. 3070–5. http://dx.doi.org/10.1016/j.renene.2015.06.034.

[18] Cai L, Gu J, Ma J, et al. Probabilistic wind power forecasting approach via instance-based transfer learning embedded gradient boosting decision trees. Energies 2019;12(1):159. http://dx.doi.org/10.3390/en12010159.

[19] Liu Y, Wang J. Transfer learning based multi-layer extreme learning machine for probabilistic wind power forecasting. Appl Energy 2022;312:118729. http://dx.doi.org/10.1016/J.APENERGY.2022.118729.

[20] Chen J, Zhu Q, Li H, Zhu L, Shi D, Li Y, Duan X, Liu Y. Learning heterogeneous features jointly: A deep end-to-end framework for multi-step short-term wind power prediction. IEEE Trans Sustain Energy 2020;11(3):1761–72. http://dx.doi.org/10.1109/TSTE.2019.2940590.

[21] Sheng H, Ray B, Shao J, Lasantha D, Das N. Generalization of solar power yield modelling using knowledge transfer. Expert Syst Appl 2022;116992. http://dx.doi.org/10.1016/J.ESWA.2022.116992.

[22] Yin H, Ou Z, Fu J, Cai Y, Chen S, Meng A. A novel transfer learning approach for wind power prediction based on a serio-parallel deep learning architecture. Energy 2021;234:121271. http://dx.doi.org/10.1016/J.ENERGY.2021.121271.

[23] Khan M, Naeem MR, Al-Ammar EA, Ko W, Vettikalladi H, Ahmad I. Power forecasting of regional wind farms via variational auto-encoder and deep hybrid transfer learning. Electronics 2022;11(2):206. http://dx.doi.org/10.3390/ELECTRONICS11020206.

[24] Almonacid-Olleros G, Almonacid G, Gil D, Medina-Quero J. Evaluation of transfer learning and fine-tuning to nowcast energy generation of photovoltaic systems in different climates. Sustainability 2022;14(5):3092. http://dx.doi.org/10.3390/SU14053092.

[25] Yan C, Pan Y, Archer CL. A general method to estimate wind farm power using artificial neural networks. Wind Energy 2019;22(11):1421–32. http://dx.doi.org/10.1002/WE.2379.

[26] Schreiber J, Buschin A, Sick B. Influences in forecast errors for wind and photovoltaic power: A study on machine learning models. In: INFORMATIK 2019. GI e.V.; 2019, p. 585–98. http://dx.doi.org/10.18420/inf2019_74.

[27] Zhou S, Zhou L, Mao M, et al. Transfer learning for photovoltaic power forecasting with long short-term memory neural network. In: International conference on big data and smart computing (BigComp). 2020, p. 125–32. http://dx.doi.org/10.1109/BigComp48618.2020.00-87.

[28] Ceci M, Corizzo R, Fumarola F, et al. Predictive modeling of PV energy production: How to set up the learning task for a better prediction? IEEE TIL 2017;13(3):956–66. http://dx.doi.org/10.1109/TII.2016.2604758.

[29] Shireen T, Shao C, Wang H, et al. Iterative multi-task learning for time-series modeling of solar panel PV outputs. Appl Energy 2018;212:654–62. http://dx.doi.org/10.1016/j.apenergy.2017.12.058.

[30] Tasnim S, Rahman A, Oo A, et al. Wind power prediction in new stations based on knowledge of existing stations: A cluster based multi source domain adaptation approach. Knowl-Based Syst 2018;145:15–24. http://dx.doi.org/10.1016/j.knosys.2017.12.036.

[31] Vogt S, Braun A, Dobschinski J, et al. Wind power forecasting based on deep neural networks and transfer learning. In: Wind integration workshop, Vol. 18. 2019, p. 8.

[32] Schreiber J, Sick B. Emerging relation network and task embedding for multi-task regression problems. In: ICPR. 2020, p. 2663–70. http://dx.doi.org/10.1109/ICPR48806.2021.9412476.

[33] Vapnik VN. The nature of statistical learning theory. Springer-Verlag; 2000.
[34] Gensler A. Wind power ensemble forecasting (Ph.D. thesis), University of Kassel; 2018, p. 204.
[35] Zhang C, Bin J, Liu Z. Wind turbine ice assessment through inductive transfer learning. In: International instrumentation and measurement technology conference. IEEE; 2018, p. 1–6. http://dx.doi.org/10.1109/I2MTC.2018.8409794.
[36] Guariso G, Nunnari G, Sangiorgio M. Multi-step solar irradiance forecasting and domain adaptation of deep neural networks. Energies 2020;13(15):1–18. http://dx.doi.org/10.3390/en13153987.
[37] Thill M, Konen W, Bäck T. Time series encodings with temporal convolutional networks. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). LNCS, vol. 12438, 2020, p. 161–73. http://dx.doi.org/10.1007/978-3-030-63710-1_13/COVER/.
[38] Zhu R, Liao W, Wang Y. Short-term prediction for wind power based on temporal convolutional network. Energy Rep 2020;6:424–9. http://dx.doi.org/10.1016/j.egyr.2020.11.219.
[39] Yan J, Mu L, Wang L, Ranjan R, Zomaya AY. Temporal convolutional networks for the advance prediction of ENSO. Sci Rep 2020;10(1). http://dx.doi.org/10.1038/s41598-020-65070-5.
[40] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016, p. 775.
[41] Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C. A survey on deep transfer learning. Lecture Notes in Comput Sci 2018;11141:270–9. http://dx.doi.org/10.1007/978-3-030-01424-7_27/COVER.
[42] Fawaz HI, Forestier G, Weber J, et al. Transfer learning for time series classification. In: 2018 IEEE bigdata. 2019, p. 1367–76. http://dx.doi.org/10.1109/BigData.2018.8621990.
[43] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. CVPR, 2016, p. 1–112. http://dx.doi.org/10.48550/arXiv.1512.03385.
[44] Bishop CM. Pattern recognition and machine learning. Springer; 2006, p. 738.
[45] Borthwick M. Math for machine learning. Cambridge University; 2019, p. 411. http://dx.doi.org/10.1515/9781400843909-001.
[46] Hoeting JA, Madigan D, Raftery AE, et al. Bayesian model averaging: a tutorial. MathSciNet 1999;14(4):382–417. http://dx.doi.org/10.1214/SS/1009212519.
[47] Gensler A, Sick B. A multi-scheme ensemble using coopetitive soft gating with application to power forecasting for renewable energy generation. 2018, CoRR arXiv:1803.06344.