

---

# Efficient Deterministic Renewable Energy Forecasting Guided by Multiple-Location Weather Data

Charalampos Symeonidis<sup>1\*</sup> and Nikos Nikolaidis<sup>2</sup>

<sup>1,2</sup>Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece.

## Abstract

Electricity generated from renewable energy sources has been established as an efficient remedy for both energy shortages and the environmental pollution stemming from conventional energy production methods. Solar and wind power are two of the most dominant renewable energy sources. The accurate forecasting of the energy generation of those sources facilitates their integration into electric grids, by minimizing the negative impact of uncertainty regarding their management and operation. This paper proposes a novel methodology for deterministic wind and solar energy generation forecasting for multiple generation sites, utilizing multi-location weather forecasts. The method employs a U-shaped Temporal Convolutional Auto-Encoder (UTCAE) architecture for temporal processing of weather-related and energy-related time-series across each site. The Multi-sized Kernels convolutional Spatio-Temporal Attention (MKST-Attention), inspired by the multi-head scaled-dot product attention mechanism, is also proposed aiming to efficiently transfer temporal patterns from weather data to energy data, without a priori knowledge of the locations of the power stations and the locations of provided weather data. The conducted experimental evaluation on a day-ahead solar and wind energy forecasting scenario on five datasets demonstrated that the proposed method achieves top results, outperforming all competitive time-series forecasting state-of-the-art methods.

**Keywords:** Deterministic RES forecasting, Time-series forecasting, Deep learning

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural Computing and Applications, and is available online at <https://doi.org/10.1007/s00521-024-10607-2>

---

# 1 Introduction

Fossil fuels, including coal, oil and natural gas, have long been the world's dominant sources of energy. Their fast depletion [1] as well as their overall impact in global warming [2], due to the emission of greenhouse gases, has led to the rapid increase in the exploitation of renewable energy sources, such as solar and wind energy, hydropower, geothermal energy and bioenergy. In recent years, solar radiation and wind have emerged as two major Renewable Energy Sources (RES), each accounting for a quarter of the total RES energy output [3]. Two major drawbacks, encountered on both RES types, are that the corresponding power generation is variable and non-dispatchable [4], i.e., the power generation output cannot (or have limited ability to) be adjusted to match the electricity demand. The integration of those RES types into existing power systems is mainly determined by: (i) the temporal variability of the output power from renewable energy generation stations and (ii) the accuracy of forecasts for the variable generation. Fluctuations in total non-dispatchable power generation along with fluctuations in the total power demand, require the in-parallel use of conventional dispatchable power generation methods on different time scales. Accurate forecasting of RES power generation [5] and the electric load demand [6], can effectively mitigate the related uncertainties, contributing positively to the planning, management and operation of energy systems and thus achieving a lower level of use for conventional power generation methods.

Accurate renewable energy generation forecasting is a challenging task due to the intermittent and random nature of renewable energy data. Methods addressing the RES forecasting problem, can be mostly categorized as physical, statistical, machine learning and hybrid. Physical forecasting methods [7][8] focus on energy forecasting based on Numerical Weather Predictions (NWP) and physical principles regarding the installed photovoltaic (PV) cells or wind turbines as well as the corresponding geographic formation. Physical methods do not require historical (past) information regarding energy generation. Statistical forecasting methods, such as Autoregressive (AR) models [9][10] and Markov chain models [11][12], aim to establish a mapping relationship between past-time and future-time energy generation time-series. Machine learning forecasting models, also referred as intelligent forecasting models in the literature, such as neural networks [13][14] and Support Vector Machines (SVMs) [15][16], attempt to establish a mathematical relationship between input and output through curve fitting and parameter optimization techniques [17]. Hybrid methods [18][19] incorporate a combination of different types of methods.

Based on the manifestation of the forecasting results, RES forecasting methods can also be categorized as deterministic [20] [21] or probabilistic [22] [23]. Deterministic methods output single-valued energy generation predictions for each time-step of the forecasting window. Probabilistic methods provide a wider view of possible energy generation outputs expressed as quantiles, prediction intervals (PIs), and distributions.

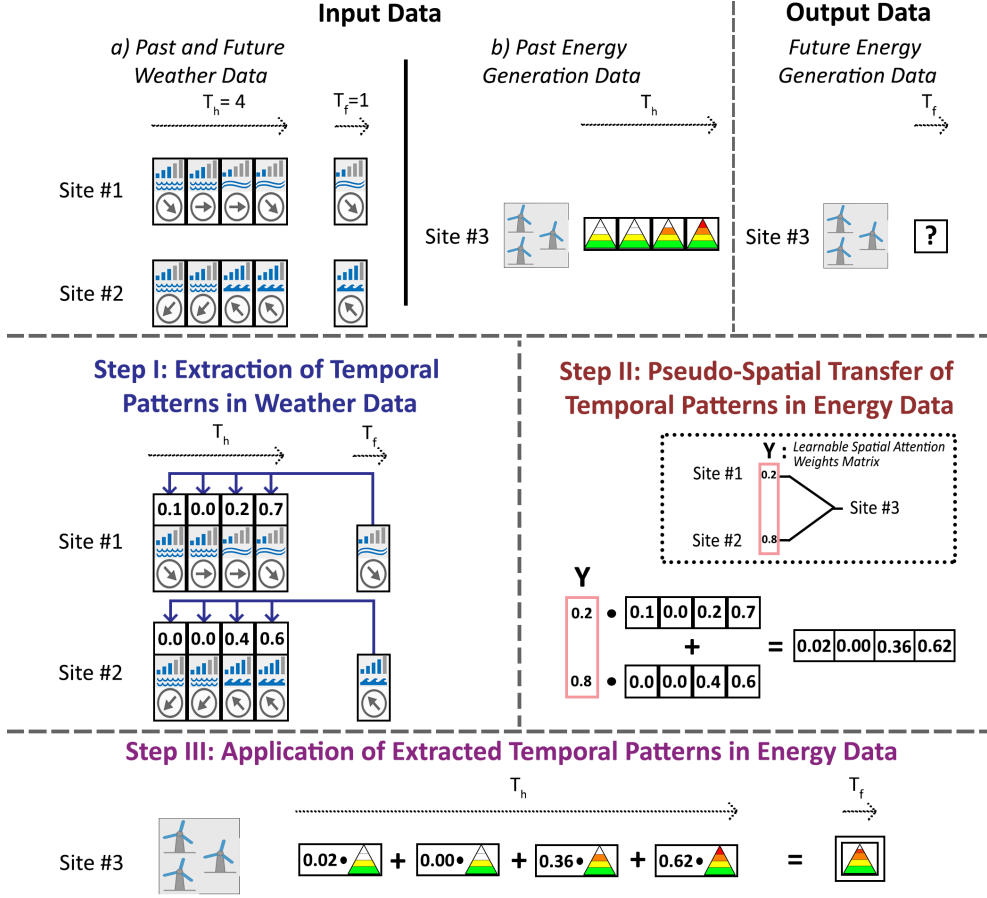
Finally, RES forecasting methods can also be categorized based on the time horizon of the forecasting window. According to the relevant literature, the methods can be split into four categories [24]; ultra-short-term, short-term, medium-term, and long-term forecasting. However, these categories often remain ambiguous since

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural

Computing and Applications, and is available online at

<https://doi.org/10.1007/s00521-024-10607-2>

no universally agreed classification criterion exists. Thus, the terms intra-hour, intra-day, day-ahead and week-ahead forecasting / prediction are more commonly encountered [25] in order to describe the forecast horizon in a more precise way.



**Fig. 1:** Abstract example of the Multi-sized Kernels convolutional Spatio-Temporal Attention (MKST-Attention) mechanism in the inference stage of a wind energy forecasting scenario, for a single wind power station (site #3) and two weather data locations (sites #1 and #2). The size of the future-time temporal window  $T_f$  is 1 and the size of past-time temporal window  $T_h$  is 4.

Our work, focuses mainly on deterministic machine learning-based wind and solar energy forecasting approaches at power station level, that utilize weather data along with past-time energy measurements, as input. Specifically, we introduce a deterministic forecasting method for a such scenario, capable to simultaneously generate multi-step (i.e., for multiple time instances) energy production forecasts for a given set of power stations. In particular, we employ a U-shaped Temporal Convolutional

---

Auto-Encoder (UTCAE) for temporal processing of weather and energy data across each site. The motivation behind this choice is to enrich the representations of a given time-series, by exploring patterns across various temporal resolutions, thus capturing both global and local contextual information. In addition, we propose the use of a Multi-sized Kernels convolutional Spatio-Temporal Attention (MKST-Attention) mechanism that is responsible for transferring temporal patterns from weather data to energy data, without a priori knowledge of the locations of the power stations and the locations where weather data were measured (or forecasted). For each energy generation site, the mechanism is able to discover the weather data locations which are significant for the site (i.e., they affect the site) and transfer temporal patterns from the corresponding weather data subset to the energy generation data. Figure 1 illustrates an instance of MKST-Attention during the inference stage. Overall, the main contributions of this work can be summarized as follows:

- We propose a deterministic multi-step wind and solar energy forecasting methodology for multiple energy generation sites, capable to efficiently utilize weather data from multiple sites, without a priori knowledge of the location of either the energy generation or the weather data sites.
- Inspired by the Multi-head Scaled-dot Product Attention mechanism, we propose the Multi-sized Kernels convolutional Spatio-Temporal Attention mechanism, which can explore spatio-temporal patterns between two sets of time-series data.
- We evaluate the proposed methodology on three wind energy forecasting datasets and two solar energy forecasting datasets, showcasing its effectiveness.

The proposed method is an extension of the method in [26], which was designed specifically for deterministic wind energy forecasting. Both methods utilize a similar convolutional attention mechanism for exploring spatio-temporal relations between energy generation data and weather data. A major difference of the proposed method with the method in [26], lies on the computation of the spatial relations between the energy generation sites and the sites of weather data. In [26], the spatial relations are time-varying and are computed based on the respective energy and weather data fed as input, whereas in the method proposed in this paper the spatial relations explored by the attention mechanism (named Multi-sized Kernels convolutional Spatio-Temporal Attention) remain fixed after the corresponding training process. Apart from the differences associated to the implemented attention mechanism, the method in [26] doesn't incorporate a mechanism to temporally process weather data, adopting a rather shallow architecture. In contrast, the current approach employs the U-shaped Temporal Convolutional Auto-Encoder (UTCAE) and incorporates a scheme of repeated Joint Processing Blocks (JPBs) with the objective of generating contextually informative representations for both the energy-related and weather-related time-series data.

---

## 2 Previous Work

In this Section, we discuss previous work on deterministic wind and solar energy forecasting at power station level, focusing on deep learning methods. A brief summary of the discussed methods is provided in Table 1.

### 2.1 Deterministic Wind Energy Forecasting Methods for Single/Multiple Power Station(s)

In [27], the authors suggested the use of a Temporal Convolutional Neural Network (TCNN), aiming to achieve accurate single station short-term wind energy forecasts. The method's performance was evaluated on a simulated dataset, provided by the National Renewable Energy Laboratory (NREL), achieving top results against the rest of the baseline methods. In addition, the evaluation highlighted the stable learning process and the strong generalization ability of the proposed method. In [13], a Spatio-Temporal Convolutional Neural Network (STCNN) was employed in a multiple-stations short-term wind energy forecasting scenario. According to the authors, the method's performance gains were due to its architecture, which combined the structure of directed Graph Convolutional Network (GCNs) with that of TCNNs, allowing the effective capture of asymmetric spatial correlations at different temporal scales. Similarly, the authors in [28] applied the Transformer model in a short-term multiple-stations wind energy forecasting scenario. The employed model was able to extract different levels of correlation between wind farms and provide accurate wind power forecasting results. In [29], the authors proposed a short-term wind energy forecasting method based on Discrete Wavelet Transform (DWT) and Long Short-Term Memory (LSTM) networks. The method adopts a divide and conquer strategy, in which DWT is used to decompose original wind energy data into sub-signals, while several independent LSTMs are employed to approximate the temporal dynamic behaviors of these sub-signals. The proposed method achieved top prediction accuracy rates against other state-of-the-art methods in a single-farm short-term wind energy forecasting scenario. In [20], the authors designed a sequence-to-sequence model for a multi-step-ahead single-station wind energy forecasting scenario. The model architecture consists of two groups of Attention-based Gated Recurrent Unit (AGRU) blocks, employed in an encoder-decoder architecture. In the proposed model, the correlation between different forecasting tasks was considered in the GRU and an attention mechanism helped select important features. An experimental evaluation, regarding the accuracy, the computational efficiency and the feature selection capabilities of the proposed model demonstrated its superiority against other state-of-the-art methods.

### 2.2 Deterministic Solar Energy Forecasting Methods for Single/Multiple Power Station(s)

In [30], the authors studied the use of a TCNN in a day-ahead forecasting scenario with half-hour temporal resolution. According to the authors, the model was able to maintain a much longer effective history, compared to competing methods, by extracting informative features from long sequences using dilated causal convolutional filters. The

authors in [31] proposed a monthly PV power generation forecasting method based on the LSTM model, using meteorological and energy data collected from existing power stations, aiming to predict the energy generation of a potential power station at a new location. Through experiments using historical training data from 134 solar power stations, the proposed approach was able to perform well in a testing set containing time-series of 30 solar power stations not used in the training process. In [21], the authors developed a single-station solar energy forecasting method based on the fusion of a sequence-to-sequence auto-encoder and a gated recurrent unit. While the GRU makes use of the data's time dependencies, the auto-encoder extracts the correlation of significant parameters like solar irradiance. The proposed method was able to perform better than LSTM and CNN-LSTM-based models utilizing a variety of performance metrics, including Mean Square Error (MSE) and Mean Absolute Error (MAE). The authors in [32], proposed a spatio-temporal CNN for solar energy forecasting, which exploits the spatial information of multiple power stations. First, the proposed Greedy Adjoining Algorithm (GAA) constructs a space-time matrix that rearranges the solar energy generation sites based on their geographical proximity. Then, a CNN is applied to learn spatio-temporal relations. The conducted experimental evaluation showed that the proposed STCNN achieves the highest error reduction when data from multiple power stations are aggregated. In [33], the authors introduced a novel spatio-temporal Graph Attention network (GAT) for solar energy forecasting in multiple power stations. The proposed method can capture different dynamical spatio-temporal correlations for different parts of the forecasting horizon. Thus, it is possible to interpret which power stations have the most influence when performing short-, medium- and long-term intra-day forecasts. The proposed method outperformed state-of-the-art methods in 4 to 6 hours ahead solar energy forecasting scenarios in all employed evaluation metric, on real and synthetic datasets.

| Method | Energy Type | Architecture      | Single/Multiple Station(s) Forecasting |
|--------|-------------|-------------------|----------------------------------------|
| [29]   | wind        | RNN, DWT          | single                                 |
| [27]   | wind        | TCNN              | single                                 |
| [20]   | wind        | AGRU              | single                                 |
| [13]   | wind        | STCNN, GCN        | multiple                               |
| [28]   | wind        | Transformer       | multiple                               |
| [26]   | wind        | STAN              | multiple                               |
| [30]   | solar       | TCNN              | single                                 |
| [21]   | solar       | Auto-Encoder, GRU | single                                 |
| [31]   | solar       | LSTM              | multiple                               |
| [32]   | solar       | STCNN             | multiple                               |
| [33]   | solar       | GAT               | multiple                               |

**Table 1:** Brief summary of the discussed deterministic wind and solar energy forecasting methods.

---

### 3 Proposed Method

Our wind/solar energy forecasting approach is inspired by attention-based and temporal convolutional architectures, both commonly encountered in the task of generic time-series forecasting. The core mechanism of the proposed method, named Multi-sized Kernels convolutional Spatio-Temporal Attention (MKST-Attention), is able to spatially transfer temporal patterns from weather data time-series to energy generation time-series. In addition, aiming to temporally process both weather and energy generation data across sites, we employ a U-shaped Temporal Convolutional Auto-Encoder (UTCAE), capable to capture multi-scale contextual information along the temporal dimension. To present the proposed method in an organized and clear way, the rest of the section is structured as follows. Section 3.1 states the corresponding problem and explains the adopted notations. Subsequently, Section 3.2 presents the Multi-sized Kernels convolutional Attention (MK-Attention), a novel variant of multi-head attention. In Sections 3.3 and 3.4 the MKST-Attention mechanism and the architecture of the UTCAE are presented, respectively. Both are employed by the Joint Processing Block (JPB), presented in Section 3.5, aiming to process and refine the latent representations of energy generation and weather data. Finally, the overall architecture of the proposed method is presented in detail in Section 3.6.

#### 3.1 Problem Statement and Notations

The problem of wind and solar energy forecasting, addressed in this paper, can be formulated as:

$$\hat{\mathbf{E}}^f = g(\mathbf{E}^h, \mathbf{W}^h, \mathbf{W}^f) \quad (1)$$

In this equation  $\mathbf{E}^h \in \mathbb{R}^{L_E \times T_h \times 1}$  corresponds to the past/history ( $h$ : history) energy generation measurements from  $L_E$  locations (e.g., locations of wind farms, PV power stations, etc.), where  $T_h$  is the size of the past-time temporal window. Moreover,  $\mathbf{W}^h \in \mathbb{R}^{L_W \times T_h \times D_W}$  corresponds to past-time weather data from  $L_W$  distinct locations, where  $D_W$  is the number of weather data measurements/variables. Also,  $\mathbf{W}^f \in \mathbb{R}^{L_W \times T_f \times D_W}$  corresponds to future-time ( $f$ :future) weather data, i.e. weather forecasts, where  $T_f$  is the size of the future-time temporal window. Finally,  $\hat{\mathbf{E}}^f \in \mathbb{R}^{L_E \times T_f \times 1}$  corresponds to the wind or solar energy predictions that are generated by the method simultaneously for each of the  $L_E$  locations.

In order to ensure clarity and consistency in representing matrix or tensor indexing throughout this work, we adopt the following notation scheme: matrix or tensor elements will be denoted using indices enclosed within angle brackets ( $\langle \cdot \rangle$ ). For instance, elements of a third-order tensor  $\mathbf{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ , will be referenced as  $\mathbf{x}\langle i, j, k \rangle$  where  $i, j$ , and  $k$  represent indices along the first, second, and third dimension, respectively. In addition, the dot symbol ( $\cdot$ ) will represent the selection of all indices within the specified dimension. For instance, the notation  $\mathbf{X}\langle i, \cdot, \cdot \rangle$  denotes the selection of  $\mathbf{X}$  elements with index  $i$  along the first dimension and all index values across the second and third dimensions. These notations will be consistently employed in all subsequent discussions and equations involving matrices or tensors.

In addition, let  $\mathbf{B} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$  and  $\mathbf{D} \in \mathbb{R}^{N_1 \times N_3 \times N_4}$  be two batches of  $N_1$  matrices, where  $\mathbf{B} = [\mathbf{B}_1, \dots, \mathbf{B}_{N_1}]$ , and  $\mathbf{D} = [\mathbf{D}_1, \dots, \mathbf{D}_{N_1}]$ . We can then define Batch

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural

Computing and Applications, and is available online at

<https://doi.org/10.1007/s00521-024-10607-2>

Matrix-Multiplication (BMM) as  $\mathbf{B} \otimes \mathbf{D} = \mathbf{G}$ , where  $\mathbf{G} \in \mathbb{R}^{N_1 \times N_2 \times N_4}$  is a batch of  $N_1$  matrices,  $\mathbf{G} = [\mathbf{G}_1, \dots, \mathbf{G}_{N_1}]$ .

Adopting the typical attention mechanism [34], the energy generation prediction  $\hat{e}^f\langle s, t \rangle$  at location  $s$  and time-step  $t$ , namely one of the elements of  $\hat{\mathbf{E}}^f = [[\hat{e}^f\langle 1, 1 \rangle, \dots, \hat{e}^f\langle 1, T_f \rangle], \dots, [\hat{e}^f\langle L_E, 1 \rangle, \dots, \hat{e}^f\langle L_E, T_f \rangle]]$  can be formulated as:

$$\hat{e}^f\langle s, t \rangle = \left( \sum_{j=1}^{T_h} \alpha\langle s, t, j \rangle \mathbf{E}^{h,r}\langle s, j, \cdot \rangle \right) \mathbf{C} \quad (2)$$

$$\text{where } \sum_{j=1}^{T_h} \alpha\langle s, t, j \rangle = 1$$

$$\forall s \in [1, \dots, L_E], \forall t \in [1, \dots, T_f], \forall j \in [1, \dots, T_h]$$

In the above formulas,  $\mathbf{A} \in \mathbb{R}^{L_E \times T_f \times T_h}$  corresponds to a non-negative attention weights tensor that expresses future-time energy generation latent representations of each site, as linear combinations of past-time energy generation latent representations of the same site. In addition,  $\mathbf{E}^{h,r}\langle s, j, \cdot \rangle \in \mathbb{R}^{D_r}$  corresponds to the latent representation ( $r$ : representations) of the past energy generation measurement at location  $s$  and past time-step  $j$  and  $D_r$  denotes the size of each energy generation latent representation. Moreover,  $\mathbf{C} \in \mathbb{R}^{D_r \times 1}$  are learnable parameters of a linear operator. In this formulation, wind or solar energy generation at  $s$ -th location and  $t$ -th time-step is predicted based on the temporal patterns, imposed by attention weights  $\mathbf{A}\langle s, t, \cdot \rangle$ , and past energy generation measurements (more specifically their latent representations) within the respective temporal window. Our objective is to explore, the previously described, pseudo-spatial and temporal relations between  $\mathbf{E}^h$ ,  $\mathbf{W}^h$  and  $\mathbf{W}^f$  in order to efficiently approximate  $\mathbf{A}$ .

### 3.2 Multi-Sized Kernels Convolutional Scaled Dot-Product Attention

The Scaled Dot-Product Attention was presented in [35] and is formulated as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{M}\mathbf{V}, \quad (3)$$

$$\text{where } \mathbf{M} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_K}}\right) \quad (4)$$

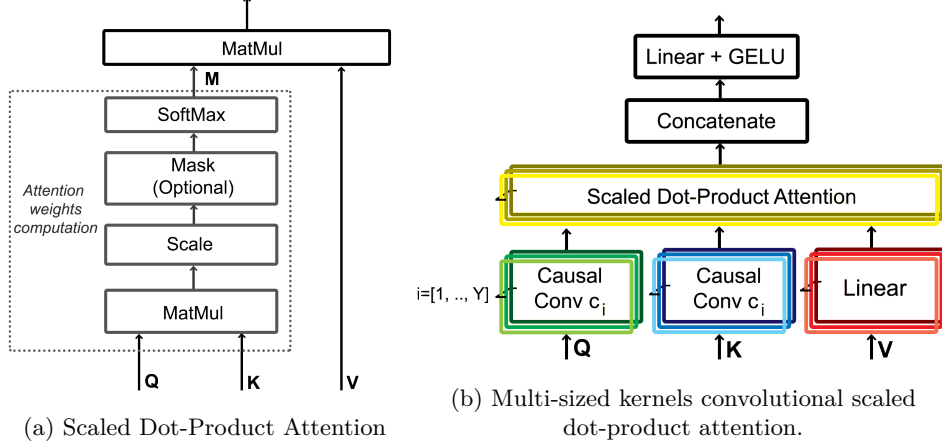
$\mathbf{Q} \in \mathbb{R}^{N_Q \times D_K}$ ,  $\mathbf{K} \in \mathbb{R}^{N_K \times D_K}$  and  $\mathbf{V} \in \mathbb{R}^{N_K \times D_V}$  are the queries, keys and values respectively. Queries and keys have a dimension of  $D_K$ , while values have a dimension of  $D_V$ .  $N_Q$  is the number of queries while  $N_K$  is the number of keys and values. An illustration of the mechanism is depicted in Figure 2a. Multi-head attention was also proposed in [35], allowing various attention mechanisms, including scaled dot-product attention, to run in parallel. To this end, instead of performing a single attention computation on queries, keys, and values with a universal dimension of  $D_r$ , the authors proposed their transformation with  $\Upsilon$  independently learned linear projections. The attention computation is then performed, in parallel, on those  $\Upsilon$  projected queries,

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural

Computing and Applications, and is available online at

<https://doi.org/10.1007/s00521-024-10607-2>





**Fig. 2:** (a): Scaled Dot-Product Attention, (b): the novel Multi-sized kernels convolutional scaled dot-product attention.  $c_i$  denotes the size of the  $i$ -th convolutional kernel, employed in the temporal domain whereas  $\Upsilon$  denotes the number of convolutional kernels.

keys, and values. More specifically, the multi-head attention module can be formulated as:

$$MultiHead(Q, K, V) = [p_1, \dots, p_\Upsilon] S^O, \quad (5)$$

$$\text{where } p_i = Attention(QS_i^Q, KS_i^K, VS_i^V) \quad (6)$$

In this formulation,  $S_i^Q \in \mathbb{R}^{D_r \times D_K}$ ,  $S_i^K \in \mathbb{R}^{D_r \times D_K}$ ,  $S_i^V \in \mathbb{R}^{D_r \times D_V}$ ,  $S^O \in \mathbb{R}^{\Upsilon D_V \times D_r}$  are projection parameter matrices,  $\Upsilon$  is the number of heads,  $D_K = D_V = \frac{D_r}{\Upsilon}$ , and the operator  $[...]$  implies concatenation.

On the original formulation, the scaled dot-product attention was designed to explore point-wise similarities between queries and keys. However, in most time-series analysis tasks, information regarding the context surrounding the observed points is vital for exploring patterns among the series. The authors in [36] were able to employ convolutions of kernel size  $c$  to transform inputs into queries and keys. Thus, local context was exploited in the query-key matching, improving the way temporal patterns among the corresponding time-series are explored. The authors experimented with various values of  $c$  in order to find the optimal one. Instead of selecting one kernel size, we propose the Multi-sized Kernels convolutional scaled dot-product Attention, which can be formulated by modifying Equations 5 and 6 as follows:

$$MultiKernel(Q, K, V) = [p_1, \dots, p_\Upsilon] S^O, \quad (7)$$

$$p_i = Attention(g_Q(Q, c_i), g_K(K, c_i), VS_i^V) \quad (8)$$

where  $g_Q(Q, c)$  and  $g_K(K, c)$  correspond to 1-D convolutional layers of  $D_V$  kernels,  $c$  being the size of 1-D convolutional kernels. In this formulation, convolutions with  $\Upsilon$

different kernel sizes are applied on  $\mathbf{Q}, \mathbf{K}$  resulting in  $\Upsilon$  heads. In addition, a separate linear projection is applied on  $\mathbf{V}$  for each head. In this way, the scaled dot-product attention is computed separately for each head. It should be noted that this process cannot be executed simultaneously but iteratively for all heads. Then, the outputs are concatenated and projected as depicted in Figure 2b.

### 3.3 Multi-Sized Kernels Convolutional Spatio-Temporal Attention

Aiming to formulate the spatio-temporal attention (ST-Attention) we model queries, keys and values in space-time as  $\mathbf{Q} \in \mathbb{R}^{L_K \times T_Q \times D_K}$ ,  $\mathbf{K} \in \mathbb{R}^{L_K \times T_K \times D_K}$ ,  $\mathbf{V} \in \mathbb{R}^{L_V \times T_K \times D_K}$ , respectively, where the first dimension of these tensors is the spatial one, the second dimension is the temporal one and the last corresponds to the size of each vector representation. In addition, we define  $\mathbf{Y} \in \mathbb{R}^{L_V \times L_K}$ , which is a matrix that holds the spatial relations between the  $L_V$  sites of the values and the  $L_K$  sites of queries and keys. Based on Equation 3, we formulate spatio-temporal attention (ST-Attention) as follows:

$$ST-Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{Y}) = \mathbf{B} \otimes \mathbf{V} \quad (9)$$

where

$$\mathbf{B} = [\sum_{i=1}^{L_K} y\langle 1, i \rangle \mathbf{M}\langle i, \cdot, \cdot \rangle, \dots, \sum_{i=1}^{L_K} y\langle L_V, i \rangle \mathbf{M}\langle i, \cdot, \cdot \rangle], \quad (10)$$

$$\mathbf{M}\langle i, \cdot, \cdot \rangle = softmax(\frac{\mathbf{Q}\langle i, \cdot, \cdot \rangle \mathbf{K}\langle i, \cdot, \cdot \rangle^T}{\sqrt{D_K}}), \quad (11)$$

$$\sum_{i=1}^{L_K} y\langle j, i \rangle = 1, \quad (12)$$

$$\forall i \in [1, L_K], \forall j \in [1, L_V]$$

In this formulation  $\mathbf{M} \in \mathbb{R}^{L_K \times T_Q \times T_K}$  holds temporal relations for each of the  $L_K$  sites of keys/values separately. More specifically,  $\mathbf{M}\langle i, \cdot, \cdot \rangle$  holds temporal relations formed between  $\mathbf{Q}\langle i, \cdot, \cdot \rangle$  and  $\mathbf{K}\langle i, \cdot, \cdot \rangle$  for the  $i$ -th site of queries and keys. The temporal relations in  $\mathbf{M}$  can be spatially applied to the sites of the values through  $\mathbf{Y}$ , forming tensor  $\mathbf{B} \in \mathbb{R}^{L_V \times T_Q \times T_K}$ , as shown in Equation 10. Finally, the output of ST-Attention is computed as the batch matrix-multiplication of tensors  $\mathbf{B}$  and  $\mathbf{V}$ .

Combining the MK-Attention, expressed in Equations 7, 8 with ST-Attention, we formulate the Multi-sized Kernels convolutional Spatio-Temporal Attention (MKST-Attention):

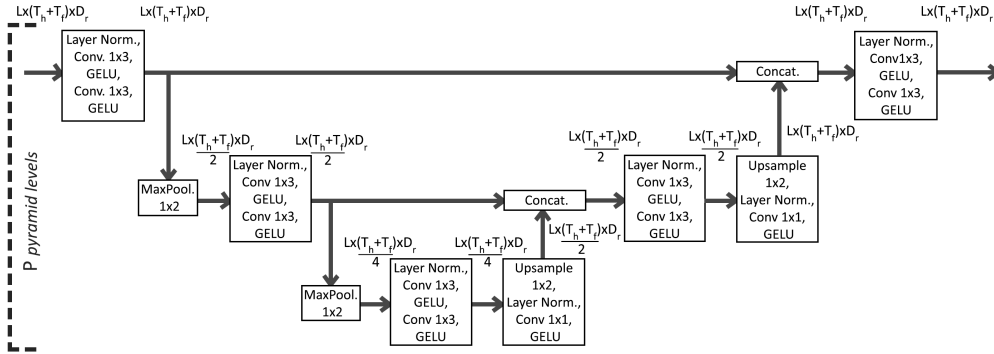
$$MKST-Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{Y}) = [\mathbf{p}_1, \dots, \mathbf{p}_\Upsilon] \otimes \mathbf{S}^A \quad (13)$$

$$\mathbf{p}_i = ST-Attention(g_Q(\mathbf{Q}, c_i), g_K(\mathbf{K}, c_i), \mathbf{V}\mathbf{S}_i^V, \mathbf{Y}) \quad (14)$$

where  $\mathbf{S}^A \in \mathbb{R}^{L_V \times \Upsilon D_V \times D_r}$  is a tensor formed as a batch of  $L_V$  repetitions of the  $\mathbf{S}^O \in \mathbb{R}^{\Upsilon D_V \times D_r}$  projection parameter matrix.

### 3.4 U-Shaped Temporal Convolutional Auto-Encoder

High temporal resolution weather and energy generation data (e.g. per minute or hour measurements/estimations) might contain fluctuations or noise resulting from sensor inaccuracies or due to their processing procedure. As a result, high-frequency changes in weather data might not coincide with equivalent changes in energy generation data and vice versa. This has a very negative impact on identifying temporal patterns between those two modalities. Aiming to diminish such data inaccuracies as well as to extract essential contextual information from multiple temporal scales we employ a U-shaped Temporal Convolutional Auto-Encoder (UTCAE). The corresponding architecture, depicted in Figure 3, shares similarities with the architecture of U-Net [37], a neural network originally proposed for the task of biomedical image segmentation. It should be noted that UNet-inspired architectures [38] [39] have also been applied in time-series related tasks. The architecture of UTCAE, has a pyramid structure of  $P$  levels, and consists of two parts. The encoding part comprises mostly of convolutional layers followed by max pooling operations, aiming to progressively reduce the temporal dimension of the time-series data while extracting latent representations with more contextual information. The decoding part, at its core, consists of upsampling operations, thus increasing gradually the temporal dimension back to the original size. Skip connections, enable information from the layers of the encoder to be delivered directly to the decoder, facilitating the gradients to propagate smoother during training by providing shortcuts through the network.



**Fig. 3:** Illustration of the architecture of the U-shaped Temporal Convolutional Auto-Encoder, where the number of pyramid levels is 3.

### 3.5 Joint Processing Block

The Joint Processing Block (JPB) can be formulated as:

$$\mathbf{E}^{out}, \mathbf{W}^{out} = JPB(\mathbf{E}^{in}, \mathbf{W}^{in}, \mathbf{Y}) \quad (15)$$

where

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural Computing and Applications, and is available online at <https://doi.org/10.1007/s00521-024-10607-2>

$$\mathbf{E}^{out} = MKST-Attention(\mathbf{W}^{out}, \mathbf{W}^{out}, \mathbf{E}^u, \mathbf{Y}) + \mathbf{E}^u, \quad (16)$$

$$\mathbf{W}^{out} = U_W(\mathbf{W}^{in}), \quad (17)$$

$$\mathbf{E}^u = U_E(\mathbf{E}^{in}). \quad (18)$$

As input, the block receives spatio-temporal energy generation data representations  $\mathbf{E}^{in} \in \mathbb{R}^{L_E \times (T_h + T_f) \times D_r}$  and spatio-temporal weather data representations  $\mathbf{W}^{in} \in \mathbb{R}^{L_W \times (T_h + T_f) \times D_r}$ . Matrix  $\mathbf{Y} \in \mathbb{R}^{L_E \times L_W}$ , which holds the spatial relations between the  $L_E$  energy generation sites and the  $L_W$  weather data sites, is also provided to the JPB. At first stage, the energy generation data and weather data are separately processed by the U-shaped Temporal Convolutional Auto-Encoders, denoted as  $U_E(\cdot)$  and  $U_W(\cdot)$ , resulting in refined energy- and weather-data latent representations  $\mathbf{E}^u \in \mathbb{R}^{L_E \times (T_h + T_f) \times D_r}$  and  $\mathbf{W}^{out} \in \mathbb{R}^{L_W \times (T_h + T_f) \times D_r}$ . Then, those latent representations, along with  $\mathbf{Y} \in \mathbb{R}^{L_E \times L_W}$  are fed to MKST-Attention, which outputs the final energy generation representation  $\mathbf{E}^{out} \in \mathbb{R}^{L_E \times (T_h + T_f) \times D_r}$ . A residual connection is also applied between  $\mathbf{E}^{out}$  and  $\mathbf{E}^u$ . The architecture of JPB is depicted in Figure 4.

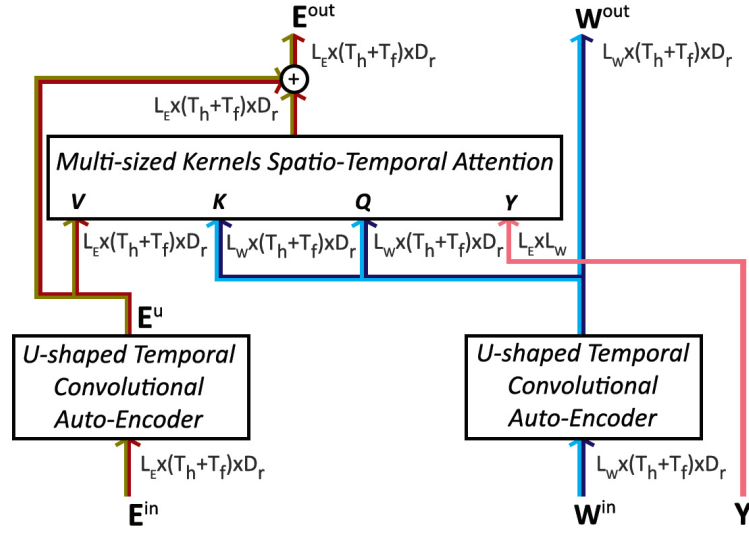


Fig. 4: Architecture of the Joint Processing Block.

### 3.6 Overall Architecture

The overall architecture of the proposed method is depicted on Figure 5. The method receives as input  $\mathbf{E}^h$ ,  $\mathbf{W}^h$  and  $\mathbf{W}^f$  and feeds them into respective linear projection layers, forming  $\mathbf{E}^{h, rin} \in \mathbb{R}^{L_E \times T_h \times D_r}$ ,  $\mathbf{W}^{h, rin} \in \mathbb{R}^{L_W \times T_h \times D_r}$  and  $\mathbf{W}^{f, rin} \in \mathbb{R}^{L_W \times T_f \times D_r}$ , respectively. In this formulation,  $L_E$  is the number of energy generation sites,  $L_W$  is the number of the weather data sites,  $T_h$  is the number of past time-steps. In addition, the method receives spatial encodings  $\mathbf{S} \in \mathbb{R}^{(L_E + L_W) \times D_S}$ , where  $D_S$  is the size of the

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural

Computing and Applications, and is available online at

<https://doi.org/10.1007/s00521-024-10607-2>

encodings. Although these encodings can be generated based on the geolocation coordinates of the respective sites, if these are not known, which is the case studied in this paper, one can use the one-hot encoding scheme for generating discrete pseudo-spatial encodings for all sites. The encodings are fed to a linear layer, in order to linearly project them into a  $D_r$ -dimensional space. Since the method processes energy generation data from all sites using the same neural layers, the projected pseudo-spatial encodings of the energy generation sites are added to  $\mathbf{E}^{h,rin}$ , so as to inject the spatial information into the neural network. Similarly, the pseudo-spatial encodings of weather data are added to  $\mathbf{W}^{h,rin}$  and  $\mathbf{W}^{f,rin}$ . In addition, those spatial encodings are utilized for the generation of matrix  $\mathbf{Y} \in \mathbb{R}^{L_E \times L_W}$ , which holds the spatial relations between the  $L_E$  energy generation locations and the  $L_W$  weather data locations. The spatial attention weights matrix can be defined as:

$$\mathbf{Y} = softmax(\frac{\mathbf{\Lambda}^E \mathbf{\Lambda}^W T}{\sqrt{D_\Lambda}}) \quad (19)$$

$$\text{where } \mathbf{\Lambda}^E = \mathbf{S}^{E,r} \mathbf{C}^E \text{ and } \mathbf{\Lambda}^W = \mathbf{S}^{W,r} \mathbf{C}^W \quad (20)$$

In this formulation, the attention weights in  $\mathbf{Y}$  are computed similar to Equation 4, through  $\mathbf{\Lambda}^E \in \mathbb{R}^{L_E \times D_\Lambda}$  and  $\mathbf{\Lambda}^W \in \mathbb{R}^{L_W \times D_\Lambda}$  matching.  $\mathbf{S}^{E,r} \in \mathbb{R}^{L_E \times D_r}$  and  $\mathbf{S}^{W,r} \in \mathbb{R}^{L_W \times D_r}$  are the projected spatial encodings of energy generation sites and weather data sites, respectively. In addition,  $\mathbf{C}^E \in \mathbb{R}^{D_r \times D_\Lambda}$  and  $\mathbf{C}^W \in \mathbb{R}^{D_r \times D_\Lambda}$  are learnable parameters of the linear projection layers.

Aiming to generate future-time energy generation latent representations  $\mathbf{E}^{f,rin} \in \mathbb{R}^{L_E \times T_f \times D_r}$  we employ the MKST-Attention mechanism described in Section 3.3 as follows:

$$\mathbf{E}^{f,rin} = MKST\text{-}Attention(\mathbf{W}^{f,rin}, \mathbf{W}^{h,rin}, \mathbf{E}^{h,rin}, \mathbf{Y}) \quad (21)$$

Then,  $\mathbf{E}^{rin} \in \mathbb{R}^{L_E \times (T_h + T_f) \times D_r}$ , which results from the concatenation of  $\mathbf{E}^{h,rin}$  and  $\mathbf{E}^{f,rin}$  in the temporal dimension and  $\mathbf{W}^{rin} \in \mathbb{R}^{L_W \times (T_h + T_f) \times D_r}$ , which results from the concatenation of  $\mathbf{W}^{h,rin}$  and  $\mathbf{W}^{f,rin}$  also in the temporal dimension, are sequentially processed by  $\phi$  stacked Joint Processing Blocks, resulting in refined latent energy generation and weather data representations, namely  $\mathbf{E}^{rJPB} \in \mathbb{R}^{L_E \times (T_h + T_f) \times D_r}$  and  $\mathbf{W}^{rJPB} \in \mathbb{R}^{L_W \times (T_h + T_f) \times D_r}$  ( $rJPB$ : representations derived through Joint Processing Blocks). Thereafter, the final representation of energy generation predictions  $\mathbf{E}^{f,rout} \in \mathbb{R}^{L_E \times T_f \times D_r}$  are computed as follows:

$$\mathbf{E}^{f,rout} = MKST\text{-}Attention(\mathbf{W}^{f,rJPB}, \mathbf{W}^{h,rJPB}, \mathbf{E}^{h,rJPB}, \mathbf{Y}) + \mathbf{E}^{f,rJPB} \quad (22)$$

where  $\mathbf{E}^{h,rJPB} \in \mathbb{R}^{L_E \times T_h \times D_r}$  and  $\mathbf{E}^{f,rJPB} \in \mathbb{R}^{L_E \times T_f \times D_r}$  are past- and future-time energy generation representations and  $\mathbf{W}^{h,rJPB} \in \mathbb{R}^{L_W \times T_h \times D_r}$ ,  $\mathbf{W}^{f,rJPB} \in \mathbb{R}^{L_W \times T_f \times D_r}$  are past- and future-time weather data representations, all generated through the  $\phi$  stacked Joint Processing Blocks. Finally,  $\mathbf{E}^{f,rout}$  is fed into a linear layer, thus obtaining the energy generation predictions  $\hat{\mathbf{E}}^f$ .

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural

Computing and Applications, and is available online at

<https://doi.org/10.1007/s00521-024-10607-2>



---

## 4 Experimental Evaluation

The experimental evaluation of the proposed method was conducted in a day-ahead (forecasting horizon of 24 hours) solar/wind energy generation forecasting scenario, utilizing weather data (predictions) for the corresponding forecasting window, as well as past energy generation data and weather data for the last 14 days (past-time measurements of 336 hours). Specifically for the solar energy forecasting task, time-related features, such as the hour of the day, the month of the year and the season, were treated as cyclical features and were encoded in polar coordinates. Then, they were concatenated along with weather data before being fed to the proposed method. It shall be noted that the time-related features were not employed in the wind energy forecasting task due to the lack of a noticeable correlation between them and the wind energy generation.

The rest of the section is structured as follows. First, details of the utilized datasets are provided, followed by the description of the proposed method implementation details. Next, the selected baseline methods are described and, finally, the experimental results are provided and discussed.

### 4.1 Dataset Description

Five datasets were used in the experiments, as detailed below.

**GEFCom2014 wind/solar:** The complete dataset, named *GEFCom2014 dataset* [40], contains four parts, each for a specific energy-related forecasting task. These tasks are: (i) electric load forecasting, (ii) electricity price forecasting, (iii) wind energy forecasting and (iv) solar energy forecasting. In our experimental evaluation, only the latter two were employed. According to the authors in [40], *GEFCom2014-wind* contains wind energy data and wind-related weather data obtained from ten power stations in Australia, featuring an hourly temporal resolution. The locations of the stations are not disclosed. Both the energy data and the wind-related weather data span the period from 2012 to 2013. The weather data contain the zonal component  $u$  and meridional component  $v$  of the wind velocity vector estimated at 10m and 100m above ground level. In a similar manner, *GEFCom2014-solar* contains solar energy generation data as well as weather data consisting of 12 variables, such as Total Cloud Cover (TCC), Surface Solar Rad Down (SSRD), Surface Thermal Rad Down (STRD), Total Column Liquid Water (TCLW) and Surface Pressure (SP), from three solar power stations in Australia, whose exact locations are not provided. Both the energy data and the wind-related weather data span the period from 2012 to 2014. The weather data of both sub-datasets were obtained from the European Centre for Medium-range Weather Forecasts (ECMWF) and were to be issued each day at midnight. The solar energy data, along with its associated weather data, maintains a temporal resolution of one hour. Both *GEFCom2014-wind* and *GEFCom2014-solar* were proposed for day-ahead probabilistic energy prediction, with an hourly resolution. However, in our experiments we employed them for deterministic day-ahead wind and solar energy generation forecasting at an hour-level temporal resolution. Details regarding the dataset’s split are provided in Table 2.

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural

Computing and Applications, and is available online at

<https://doi.org/10.1007/s00521-024-10607-2>

| Set            | Forecasting Period     |   |            |                         |   |            |
|----------------|------------------------|---|------------|-------------------------|---|------------|
|                | <i>GEFCom2014-wind</i> |   |            | <i>GEFCom2014-solar</i> |   |            |
| Training set   | 15/01/2012             | - | 31/10/2013 | 16/04/2012              | - | 30/04/2014 |
| Validation set | 01/11/2013             | - | 30/11/2013 | 01/05/2014              | - | 31/05/2014 |
| Testing set    | 01/12/2013             | - | 30/12/2013 | 01/06/2014              | - | 30/06/2014 |

**Table 2:** Training/Validation/Testing split of *GEFCom2014-wind/solar*

**AEMO-H:** The dataset was originally proposed in [23] for very short-term probabilistic wind energy forecasting. It contains 5-minute resolution wind energy data from 22 wind farms in south-east Australia from the period 2012-2013 and it was made publicly available by the Australian Electricity Market Operator (AEMO). In our experimental evaluation, two modifications / enhancements were applied. First, 5-minute resolution weather data, corresponding to the locations of the wind farms, were retrieved from the Solcast<sup>1</sup> online service. Details regarding the retrieved weather data are presented in Table 3a. Second, the dataset’s temporal resolution was modified from 5 minutes to 1 hour, aiming to employ the dataset in an 1-hour resolution, day-ahead wind energy generation forecasting scenario. The temporal resolution was modified by averaging the wind energy generation values as well as the average of weather data variables within each hour interval. Details about the dataset’s split are presented in Table 3b.

| Weather Data Variables          |                              |
|---------------------------------|------------------------------|
| Air temperature (°C)            | Dew point temperature (°C)   |
| Precipitable water ( $kg/m^2$ ) | Precipitable rate ( $mm/h$ ) |
| Wind speed 10m/100m ( $m/s$ )   | Wind direction 10m/100m (°)  |
| Surface pressure ( $hPa$ )      |                              |

(a) Weather data variables

| Set            | Forecasting Period      |
|----------------|-------------------------|
| Training set   | 15/01/2012 - 30/11/2012 |
| Validation set | 01/12/2012 - 31/12/2012 |
| Testing set    | 01/01/2013 - 30/12/2013 |

(b) Training/Validation/Testing split

**Table 3:** Details regarding AEMO-H.

**ENTSO-E wind/solar:** The datasets consist of (i) hourly wind/solar energy generation data for Greece (the entire country), collected by the European Network of Transmission System Operators for Electricity<sup>2</sup> (ENTSO-E), and (ii) hourly weather data for 29 locations in Greece, retrieved by the OpenWeather<sup>3</sup> online service. The coordinates of these locations are depicted in Table 4a. These locations coincide with locations of large solar and wind power stations in Greece. The weather data variables that were employed in the experiments are listed in Table 4b. The dataset spans

<sup>1</sup><https://solcast.com/>

<sup>2</sup><https://transparency.entsoe.eu>

<sup>3</sup><https://openweathermap.org/>



the period 2017-2023. Details about the dataset’s split are depicted in Table 4c. Preliminary or modified versions of the datasets were originally employed in [26] and [41].

| ID | Lat.    | Lon.    | ID | Lat.    | Lon.    | ID | Lat.    | Lon.    |
|----|---------|---------|----|---------|---------|----|---------|---------|
| 1  | 37.5411 | 22.5951 | 11 | 41.1655 | 25.9326 | 21 | 38.2119 | 23.4278 |
| 2  | 37.7662 | 22.8489 | 12 | 41.1528 | 25.8356 | 22 | 41.2958 | 25.8879 |
| 3  | 41.0472 | 25.9823 | 13 | 38.1886 | 23.2295 | 23 | 40.4103 | 22.0868 |
| 4  | 37.4915 | 23.1672 | 14 | 38.4691 | 23.2353 | 24 | 41.1553 | 25.9094 |
| 5  | 37.4313 | 22.3016 | 15 | 38.4066 | 21.8195 | 25 | 41.2986 | 23.3747 |
| 6  | 37.5007 | 23.3038 | 16 | 36.8331 | 22.9094 | 26 | 38.2305 | 23.5026 |
| 7  | 38.1395 | 22.1279 | 17 | 38.2889 | 20.4995 | 27 | 38.3603 | 23.2384 |
| 8  | 40.8091 | 21.2575 | 18 | 41.2038 | 24.2845 | 28 | 38.4308 | 22.2748 |
| 9  | 37.4779 | 23.2937 | 19 | 40.9651 | 25.9702 | 29 | 36.8554 | 22.9517 |
| 10 | 37.5286 | 23.2460 | 20 | 38.2226 | 31.8676 | -  | -       | -       |

(a) Coordinates of the 29 weather locations.

| Weather Data Variables                     |                    |                           |
|--------------------------------------------|--------------------|---------------------------|
| Min/max/feels like/actual temperature (°C) |                    |                           |
| Atmospheric Pressure ( <i>hPa</i> )        |                    | Humidity (%)              |
| Cloudiness (%)                             | Wind direction (%) | Wind speed ( <i>m/s</i> ) |

(b) Types of NLPs

| Set            | Forecasting Period      |
|----------------|-------------------------|
| Training set   | 15/01/2017 - 31/12/2020 |
| Validation set | 01/01/2021 - 31/12/2021 |
| Testing set    | 01/01/2022 - 30/01/2023 |

(c) Training/Validation/Testing split

**Table 4:** Details regarding ENTSO-E wind/solar.

## 4.2 Implementation details

The majority of parameters related to the proposed method architecture were set universally for all datasets. More specifically, the size of latent representations  $D_r$ , for both energy generation data and weather data, was set to 48. In MKST-Attention,  $D_K$  and  $D_V$  were set to 16. The selected kernel sizes were set to [3, 5, 7]. Thus,  $\Upsilon$ , namely the number of the kernel sizes, was set to 3. Regarding UTCAE, the number of pyramid levels  $P$  was set to 4. Finally, the number  $\phi$  of JPB blocks was set to 3. To prevent over-fitting, dropout regularization was implemented with a rate between 0.1 and 0.2. In addition, Reversible Instance Normalization [42] (RevIN) was implemented for addressing the distribution shift problem.

The method was trained by utilizing the Mean Squared Error (MSE) as loss function:

$$\mathcal{L} = \frac{1}{L_E} \frac{1}{T_f} \sum_{s=1}^{L_E} \sum_{t=1}^{T_f} (\hat{e}\langle s, t \rangle - e\langle s, t \rangle)^2 \quad (23)$$

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural

Computing and Applications, and is available online at

<https://doi.org/10.1007/s00521-024-10607-2>

---

The training parameters were also set universally for all datasets. Adam optimizer was used during training, with the initial learning rate set to  $5 \times 10^{-4}$ . The ReduceLROnPlateau<sup>4</sup> scheduler, was employed for reducing the learning accordingly. The early stopping technique was employed for finding the optimal number of training epochs, based on the performance of the method on the validation set of each dataset. The implemented model, which supports batch-training, was trained using batches of 8 samples.

### 4.3 Baseline Methods

Nine neural and two non-neural SoA time-series forecasting methods were selected as baselines for comparison against the proposed method in the wind/solar energy forecasting tasks. All methods, except [26], are time-series forecasting methods that haven't been explicitly proposed for wind/solar energy forecasting. The rationale for this selection stemmed from the fact that none of the wind/solar energy forecasting methods outlined in Section 2 is provided in open source. It shall be noted however that the majority of these papers follow a similar experimental evaluation methodology wherein the respective proposed method is compared against generic time-series forecasting methods.

Table 5 lists all baseline methods, and indicates the type (past-time, future-time, or both) of covariates supported by each method. In all methods, past energy generation measurements were employed as lagged input variables, while weather data were employed as covariates. Specifically for the solar energy forecasting task, we utilized as covariates the concatenation of weather data and time-related features.

Implementations of all baseline methods, except [26], were retrieved by the *Darts*[43] library. Thus, details about any referred implementation modifications against the originally proposed version of each baseline method, can be found in *Darts* official GitHub repository<sup>5</sup>. The Asynchronous Successive Halving Algorithm [44] (ASHA) scheduler, was utilized for hyperparameter tuning and for selecting the initial learning rate. Similar to our method, the ReduceLROnPlateau scheduler was employed for reducing the learning accordingly and the early stopping technique was employed for finding the optimal number of training epochs, based on the performance of each method on the validation set of each dataset. In all methods, MSE was used as the corresponding loss function. Finally, all baseline methods, except [26], were employed in a single-site manner. In other words, a separate model was trained and evaluated for each energy generation site, and only weather data for the corresponding site were used. In ENTSO-E wind and solar datasets, the weather data were spatially aggregated before being fed as covariates, instead of being fed separately for each of the 29 locations. This choice was made based on preliminary experiments, where the baseline methods, using weather data from all 29 locations, achieved better results in training set but performed poorly in validation and testing sets, demonstrating the inability of the methods to select weather data from relevant locations, leading thus to over-fitting.

---

<sup>4</sup>[https://pytorch.org/docs/stable/generated/torch.optim.lr\\_scheduler.ReduceLROnPlateau.html](https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html)

<sup>5</sup><https://github.com/unit8co/darts>

---

The two selected non-neural time-series forecasting methods, both utilizing future-time covariates as input, are Random Forest regression [45] and Kalman Filtering [46]. Among the selected neural time-series forecasting methods, *N-HiTS* [47] combines two complementary techniques, multi-rate input sampling and hierarchical interpolation, to produce drastically improved, interpretable, and computationally efficient long-horizon time-series predictions. N-HiTS is a univariate method, accepting as input only the predicted time-series history. In the conducted evaluation, a modified version of N-HiTS, presented in [43], was employed, supporting the use of past-time weather data as past-time covariates. In addition, an implementation of a dilated Temporal Convolutional Neural Network (TCNN), suitable for time-series forecasting inspired from [48], was employed. The utilized implementation also supports the use of past-time covariates. An implementation of the Transformer [35] model, supporting past-time weather-data as covariates, was also included in the conducted evaluation. Furthermore, a deterministic autoregressive recurrent neural network, inspired by *DeepAR*[49], was employed for solar/wind energy generation forecasting. The network is capable of efficiently receiving and processing future-time covariates as input. DLinear and NLinear are two simple one-layer linear models, proposed in [50] for long-term time-series forecasting tasks. Both utilize past- and future-time covariates as input. Moreover, Time-series Dense Encoder (TiDE) [51] is a Multi-layer Perceptron (MLP)-based encoder-decoder model, suitable for long-term time-series forecasting tasks, that enjoys the simplicity and speed of linear models while also being able to handle past- and future-time covariates and non linear dependencies. Additionally, the Temporal Fusion Transformer (TFT) [52] method adopts an attention-based architecture that is able to learn temporal relationships at different scales, using recurrent layers for local processing and interpretable self-attention layers for long-term dependencies. TFT allows the use of both past- and future-time covariates, while utilizing specialized components to select relevant features and a series of gating layers to suppress unnecessary components. Finally, the method in [26], designed specifically for wind energy forecasting based on weather data from multiple locations, is also employed in the experimental evaluation process on both wind and solar energy forecasting tasks. As already mentioned in Section 1, this method is a preliminary version of the one proposed in this paper. The key differences between the two methods are highlighted in the same Section.

## 4.4 Results

Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) were utilized to measure the performance of the methods in each energy generation site. To achieve a fair comparison, each method was trained four times from scratch with the same setup, and the mean MAE and RMSE are reported. For each dataset, both metrics are reported separately for each energy generation site. The average MAE and RMSE values, computed across all sites, are also provided.

**GEFCom2014-wind:** Tables 6 and 7 report MAE and RMSE values achieved by the employed methods, respectively. Our proposed method attained the best average results, computed across the ten wind energy generation sites, in both evaluation metrics. More specifically it achieved MAE and RMSE values of 0.108 and 0.160

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural

Computing and Applications, and is available online at

<https://doi.org/10.1007/s00521-024-10607-2>

| Method                | Covariates |             |
|-----------------------|------------|-------------|
|                       | Past-time  | Future-time |
| N-HITS [47]           | ✓          |             |
| TCNN [48]             | ✓          |             |
| Transformer [35]      | ✓          |             |
| Kalman Filtering [46] |            | ✓           |
| Random Forest [45]    |            | ✓           |
| DeepAR-based RNN [49] |            | ✓           |
| DLinear [50]          | ✓          | ✓           |
| NLinear [50]          | ✓          | ✓           |
| TiDE [51]             | ✓          | ✓           |
| TFT [52]              | ✓          | ✓           |
| [26]                  | ✓          | ✓           |
| Ours                  | ✓          | ✓           |

**Table 5:** Comparison of the employed time-series forecasting methods in deterministic RES forecasting.

respectively. Among the rest of the methods, TFT, the DeepAR-based RNN and the method proposed in [26], which, as already mentioned, is a preliminary version of the one proposed in this paper, achieved favourable performance results. In particular, TFT achieved top MAE, being on par with our proposed method, in sites III and IV. In addition, TFT attained top RMSE in sites IV, V. The method proposed in [26] achieved top RMSE in site III.

| Method           | Energy Stations |              |              |              |              |              |              |              |              |              |              |
|------------------|-----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                  | I               | II           | III          | IV           | V            | VI           | VII          | VIII         | IX           | X            | All          |
| TCNN             | 0.206           | 0.180        | 0.231        | 0.252        | 0.263        | 0.263        | 0.203        | 0.201        | 0.190        | 0.265        | 0.225        |
| N-HITS           | 0.188           | 0.185        | 0.205        | 0.227        | 0.225        | 0.225        | 0.182        | 0.183        | 0.185        | 0.243        | 0.205        |
| Transformer      | 0.181           | 0.166        | 0.203        | 0.211        | 0.216        | 0.229        | 0.168        | 0.176        | 0.182        | 0.223        | 0.195        |
| Kalman Filtering | 0.167           | 0.144        | 0.193        | 0.172        | 0.195        | 0.202        | 0.162        | 0.170        | 0.154        | 0.207        | 0.177        |
| Random Forest    | 0.152           | 0.151        | 0.187        | 0.178        | 0.181        | 0.194        | 0.138        | 0.155        | 0.136        | 0.215        | 0.169        |
| DeepAR-based RNN | 0.125           | 0.134        | 0.121        | 0.126        | 0.141        | 0.151        | 0.082        | 0.105        | 0.110        | 0.164        | 0.126        |
| DLinear          | 0.177           | 0.181        | 0.208        | 0.216        | 0.227        | 0.224        | 0.177        | 0.176        | 0.170        | 0.250        | 0.201        |
| NLinear          | 0.201           | 0.188        | 0.211        | 0.209        | 0.222        | 0.226        | 0.204        | 0.205        | 0.194        | 0.259        | 0.212        |
| TiDE             | 0.135           | 0.122        | 0.130        | 0.138        | 0.152        | 0.156        | 0.084        | 0.107        | 0.107        | 0.185        | 0.132        |
| TFT              | 0.111           | 0.110        | <b>0.121</b> | <b>0.118</b> | 0.134        | 0.141        | 0.074        | 0.092        | 0.093        | 0.159        | 0.115        |
| [26]             | 0.099           | 0.115        | 0.123        | 0.140        | 0.152        | 0.150        | 0.084        | 0.099        | 0.100        | 0.164        | 0.122        |
| Ours             | <b>0.077</b>    | <b>0.099</b> | <b>0.121</b> | <b>0.118</b> | <b>0.132</b> | <b>0.133</b> | <b>0.070</b> | <b>0.090</b> | <b>0.090</b> | <b>0.150</b> | <b>0.108</b> |

**Table 6:** Mean Absolute Error (MAE) of the compared methods in GEFCom2014-wind dataset.

**GEFCom2014-solar:** Table 8 reports the achieved MAE and RMSE for the proposed and the baseline methods in GEFCom2014-solar dataset. Our method attained the best-average results, across the three solar energy generation sites, in both evaluation metrics. More specifically it achieved MAE and RMSE values of 0.0362 and 0.0792 respectively. Similar to GEFCom2014-wind dataset, TFT, the DeepAR-based RNN and the method in [26] achieved good results in terms of MAE and RMSE. In

| Method           | Energy Stations |              |              |              |              |              |              |              |              |              |              |
|------------------|-----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                  | I               | II           | III          | IV           | V            | VI           | VII          | VIII         | IX           | X            | All          |
| TCNN             | 0.267           | 0.222        | 0.277        | 0.303        | 0.305        | 0.310        | 0.250        | 0.257        | 0.238        | 0.317        | 0.275        |
| N-HITS           | 0.254           | 0.233        | 0.251        | 0.275        | 0.268        | 0.269        | 0.232        | 0.230        | 0.228        | 0.296        | 0.254        |
| Transformer      | 0.245           | 0.212        | 0.249        | 0.261        | 0.278        | 0.292        | 0.213        | 0.225        | 0.225        | 0.284        | 0.248        |
| Kalman Filtering | 0.223           | 0.189        | 0.237        | 0.215        | 0.250        | 0.256        | 0.204        | 0.214        | 0.198        | 0.254        | 0.224        |
| Random Forest    | 0.206           | 0.195        | 0.238        | 0.226        | 0.228        | 0.239        | 0.176        | 0.201        | 0.180        | 0.266        | 0.215        |
| DeepAR-based RNN | 0.176           | 0.187        | 0.162        | 0.183        | 0.197        | 0.221        | 0.115        | 0.157        | 0.151        | 0.234        | 0.178        |
| DLinear          | 0.245           | 0.230        | 0.256        | 0.263        | 0.271        | 0.272        | 0.227        | 0.228        | 0.217        | 0.302        | 0.251        |
| NLinear          | 0.254           | 0.239        | 0.252        | 0.258        | 0.266        | 0.270        | 0.237        | 0.237        | 0.230        | 0.311        | 0.255        |
| TiDE             | 0.183           | 0.166        | 0.173        | 0.182        | 0.201        | 0.212        | 0.115        | 0.160        | 0.149        | 0.254        | 0.180        |
| TFT              | 0.158           | 0.159        | 0.166        | <b>0.179</b> | <b>0.191</b> | 0.208        | 0.105        | 0.147        | 0.134        | 0.230        | 0.168        |
| [26]             | 0.132           | 0.161        | <b>0.161</b> | 0.193        | 0.202        | 0.202        | 0.111        | 0.148        | 0.137        | 0.225        | 0.167        |
| Ours             | <b>0.110</b>    | <b>0.148</b> | 0.163        | 0.180        | 0.198        | <b>0.201</b> | <b>0.097</b> | <b>0.144</b> | <b>0.133</b> | <b>0.224</b> | <b>0.160</b> |

**Table 7:** Root Mean Square Error (RMSE) of the compared methods in GEFCom2014-wind dataset.

particular, the DeepAR-based RNN managed to achieve the lowest MAE of 0.0386 in station II. In the same station, the method proposed in [26] achieved top RMSE of 0.830, being on par with our proposed method.

**Table 8:** Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) of the compared methods in GEFCom2014-solar dataset.

| Method           | Energy Stations |               |               |               |               |               |               |               |
|------------------|-----------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
|                  | I               |               | II            |               | III           |               | All           |               |
|                  | MAE             | RMSE          | MAE           | RMSE          | MAE           | RMSE          | MAE           | RMSE          |
| TCNN             | 0.0506          | 0.1036        | 0.0514        | 0.0990        | 0.0532        | 0.1004        | 0.0517        | 0.1010        |
| N-HITS           | 0.0534          | 0.1038        | 0.0517        | 0.0956        | 0.0555        | 0.1013        | 0.0535        | 0.1002        |
| Transformer      | 0.0571          | 0.1091        | 0.0557        | 0.1036        | 0.0566        | 0.10127       | 0.0564        | 0.1051        |
| Kalman Filtering | 0.0635          | 0.0994        | 0.0737        | 0.1092        | 0.0617        | 0.0998        | 0.0663        | 0.1028        |
| Random Forest    | 0.0571          | 0.1091        | 0.0557        | 0.1036        | 0.0566        | 0.1027        | 0.0564        | 0.1051        |
| DeepAR-based RNN | 0.0348          | 0.0825        | <b>0.0386</b> | 0.0871        | 0.0368        | 0.0842        | 0.0367        | 0.0846        |
| DLinear          | 0.0543          | 0.0990        | 0.0551        | 0.0941        | 0.0573        | 0.0965        | 0.0556        | 0.0966        |
| NLinear          | 0.0547          | 0.0991        | 0.0554        | 0.0940        | 0.0574        | 0.0961        | 0.0558        | 0.0964        |
| TiDE             | 0.0448          | 0.0885        | 0.0479        | 0.0908        | 0.0430        | 0.0912        | 0.0452        | 0.0902        |
| TFT              | 0.0375          | 0.0823        | 0.0434        | 0.0894        | 0.0395        | 0.0823        | 0.0401        | 0.0847        |
| [26]             | 0.0367          | 0.0810        | 0.0395        | <b>0.0830</b> | 0.0397        | 0.0819        | 0.0386        | 0.0820        |
| Ours             | <b>0.0331</b>   | <b>0.0755</b> | 0.0388        | <b>0.0830</b> | <b>0.0366</b> | <b>0.0791</b> | <b>0.0362</b> | <b>0.0792</b> |

**AEMO-H:** Tables 9 and 10 report the achieved MAE and RMSE for all employed methods in AEMO-H dataset. Our method attained the best results, for each of the twenty-two wind energy generation sites, on both evaluation metrics. More specifically, it achieved MAE and RMSE of 0.098 and 0.138 respectively. TFT and the DeepAR-based RNN also achieved good results in terms of MAE and RMSE. In particular, the DeepAR-based RNN attained MAE of 0.089 in energy station VII, thereby aligning its performance with that of our proposed method. Figure 6 depicts concatenated day-ahead energy predictions for seven days and station VI. The predictions were

---

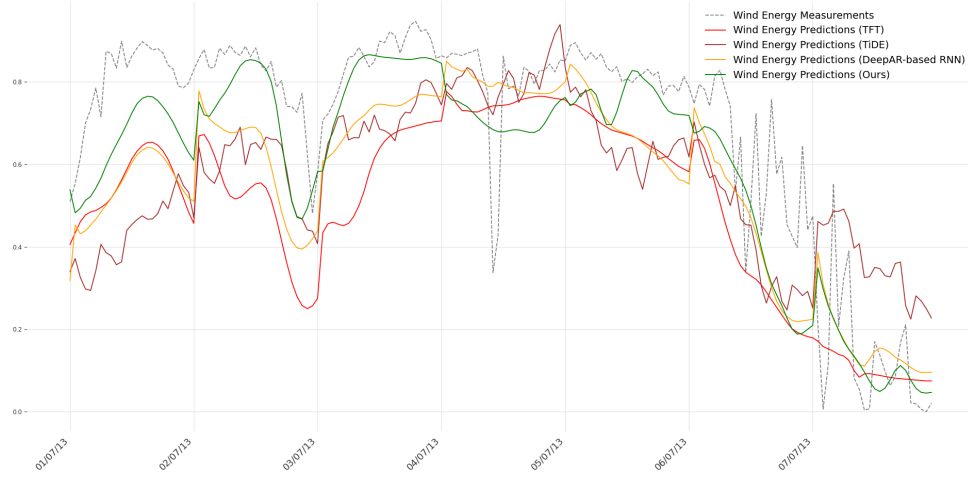
generated by the best performing methods, namely DeepAR-based RNN, TFT, TiDE, and the proposed method. These predictions are compared against the actual energy generation measurements (ground truth). It is obvious that the predictions time-series generated by the proposed method follow in general, more faithfully, the ground truth.

| Method           | Energy Stations |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|------------------|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                  | I               | II    | III   | IV    | V     | VI    | VII   | VIII  | IX    | X     | XI    | XII   | XIII  | XIV   | XV    | XVI   | XVII  | XVIII | XIX   | XX    | XXI   | XXII  | All   |
| TCNN             | 0.217           | 0.201 | 0.243 | 0.207 | 0.204 | 0.231 | 0.227 | 0.232 | 0.234 | 0.226 | 0.233 | 0.236 | 0.235 | 0.245 | 0.249 | 0.195 | 0.203 | 0.245 | 0.206 | 0.264 | 0.217 | 0.237 | 0.227 |
| N-HiTS           | 0.168           | 0.171 | 0.184 | 0.155 | 0.171 | 0.188 | 0.170 | 0.180 | 0.189 | 0.183 | 0.199 | 0.190 | 0.182 | 0.199 | 0.204 | 0.149 | 0.162 | 0.192 | 0.171 | 0.210 | 0.184 | 0.193 | 0.182 |
| Transformer      | 0.162           | 0.146 | 0.172 | 0.141 | 0.147 | 0.176 | 0.154 | 0.157 | 0.164 | 0.161 | 0.169 | 0.174 | 0.168 | 0.177 | 0.182 | 0.138 | 0.151 | 0.171 | 0.152 | 0.180 | 0.164 | 0.174 | 0.163 |
| Kalman Filtering | 0.114           | 0.120 | 0.124 | 0.102 | 0.117 | 0.156 | 0.107 | 0.120 | 0.123 | 0.123 | 0.127 | 0.140 | 0.118 | 0.128 | 0.135 | 0.102 | 0.117 | 0.122 | 0.111 | 0.143 | 0.126 | 0.125 | 0.123 |
| Random Forest    | 0.139           | 0.139 | 0.156 | 0.128 | 0.140 | 0.169 | 0.145 | 0.142 | 0.154 | 0.154 | 0.157 | 0.171 | 0.153 | 0.160 | 0.161 | 0.122 | 0.139 | 0.159 | 0.137 | 0.177 | 0.147 | 0.154 | 0.150 |
| DeepAR-based RNN | 0.090           | 0.095 | 0.101 | 0.080 | 0.104 | 0.129 | 0.089 | 0.097 | 0.108 | 0.106 | 0.117 | 0.124 | 0.100 | 0.111 | 0.113 | 0.085 | 0.105 | 0.107 | 0.104 | 0.118 | 0.105 | 0.099 | 0.104 |
| DLinear          | 0.155           | 0.159 | 0.180 | 0.136 | 0.157 | 0.179 | 0.156 | 0.157 | 0.168 | 0.164 | 0.178 | 0.189 | 0.171 | 0.185 | 0.193 | 0.134 | 0.148 | 0.178 | 0.155 | 0.197 | 0.169 | 0.177 | 0.168 |
| NLinear          | 0.165           | 0.174 | 0.186 | 0.145 | 0.160 | 0.179 | 0.162 | 0.168 | 0.174 | 0.182 | 0.196 | 0.191 | 0.177 | 0.203 | 0.211 | 0.144 | 0.155 | 0.197 | 0.161 | 0.204 | 0.186 | 0.195 | 0.178 |
| TiDE             | 0.111           | 0.123 | 0.127 | 0.096 | 0.116 | 0.146 | 0.110 | 0.112 | 0.126 | 0.133 | 0.130 | 0.149 | 0.123 | 0.136 | 0.139 | 0.100 | 0.125 | 0.133 | 0.128 | 0.148 | 0.126 | 0.123 | 0.125 |
| TFT              | 0.100           | 0.101 | 0.105 | 0.082 | 0.106 | 0.141 | 0.092 | 0.099 | 0.119 | 0.114 | 0.115 | 0.129 | 0.102 | 0.114 | 0.112 | 0.089 | 0.110 | 0.114 | 0.105 | 0.123 | 0.107 | 0.104 | 0.108 |
| [26]             | 0.125           | 0.128 | 0.136 | 0.112 | 0.125 | 0.141 | 0.127 | 0.125 | 0.139 | 0.130 | 0.144 | 0.152 | 0.136 | 0.141 | 0.144 | 0.105 | 0.120 | 0.142 | 0.128 | 0.157 | 0.134 | 0.133 | 0.133 |
| Ours             | 0.082           | 0.093 | 0.096 | 0.079 | 0.096 | 0.105 | 0.089 | 0.092 | 0.105 | 0.102 | 0.110 | 0.116 | 0.095 | 0.108 | 0.107 | 0.077 | 0.093 | 0.105 | 0.099 | 0.115 | 0.098 | 0.093 | 0.098 |

**Table 9:** Mean Absolute Error (MAE) of the compared methods in AEMO-H dataset.

| Method           | Energy Stations |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|------------------|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                  | I               | II    | III   | IV    | V     | VI    | VII   | VIII  | IX    | X     | XI    | XII   | XIII  | XIV   | XV    | XVI   | XVII  | XVIII | XIX   | XX    | XXI   | XXII  | All   |
| TCNN             | 0.273           | 0.241 | 0.294 | 0.264 | 0.247 | 0.272 | 0.275 | 0.285 | 0.280 | 0.277 | 0.290 | 0.279 | 0.284 | 0.295 | 0.300 | 0.243 | 0.255 | 0.294 | 0.256 | 0.307 | 0.270 | 0.293 | 0.276 |
| N-HiTS           | 0.230           | 0.217 | 0.238 | 0.209 | 0.221 | 0.240 | 0.219 | 0.231 | 0.240 | 0.237 | 0.255 | 0.241 | 0.231 | 0.255 | 0.264 | 0.198 | 0.215 | 0.246 | 0.224 | 0.265 | 0.239 | 0.253 | 0.235 |
| Transformer      | 0.218           | 0.189 | 0.220 | 0.192 | 0.195 | 0.226 | 0.203 | 0.209 | 0.212 | 0.211 | 0.224 | 0.219 | 0.215 | 0.227 | 0.236 | 0.182 | 0.198 | 0.221 | 0.206 | 0.233 | 0.215 | 0.226 | 0.213 |
| Kalman Filtering | 0.163           | 0.160 | 0.162 | 0.138 | 0.160 | 0.205 | 0.137 | 0.161 | 0.161 | 0.161 | 0.170 | 0.179 | 0.152 | 0.166 | 0.178 | 0.136 | 0.163 | 0.160 | 0.157 | 0.185 | 0.172 | 0.171 | 0.164 |
| Random Forest    | 0.188           | 0.179 | 0.201 | 0.174 | 0.187 | 0.220 | 0.186 | 0.185 | 0.201 | 0.200 | 0.207 | 0.212 | 0.196 | 0.203 | 0.207 | 0.163 | 0.186 | 0.201 | 0.185 | 0.223 | 0.195 | 0.206 | 0.196 |
| DeepAR-based RNN | 0.136           | 0.137 | 0.142 | 0.116 | 0.149 | 0.177 | 0.120 | 0.136 | 0.148 | 0.148 | 0.163 | 0.168 | 0.134 | 0.153 | 0.161 | 0.120 | 0.156 | 0.151 | 0.153 | 0.166 | 0.156 | 0.143 | 0.147 |
| DLinear          | 0.212           | 0.205 | 0.235 | 0.188 | 0.209 | 0.233 | 0.205 | 0.207 | 0.220 | 0.217 | 0.236 | 0.242 | 0.222 | 0.240 | 0.252 | 0.181 | 0.202 | 0.233 | 0.208 | 0.256 | 0.226 | 0.235 | 0.221 |
| NLinear          | 0.221           | 0.219 | 0.238 | 0.196 | 0.210 | 0.230 | 0.208 | 0.217 | 0.225 | 0.233 | 0.251 | 0.241 | 0.226 | 0.256 | 0.266 | 0.191 | 0.208 | 0.248 | 0.214 | 0.260 | 0.241 | 0.253 | 0.230 |
| TiDE             | 0.160           | 0.165 | 0.171 | 0.134 | 0.159 | 0.191 | 0.142 | 0.152 | 0.169 | 0.177 | 0.180 | 0.196 | 0.163 | 0.181 | 0.189 | 0.136 | 0.175 | 0.177 | 0.177 | 0.198 | 0.178 | 0.172 | 0.170 |
| TFT              | 0.148           | 0.140 | 0.146 | 0.117 | 0.152 | 0.186 | 0.122 | 0.138 | 0.166 | 0.153 | 0.163 | 0.171 | 0.137 | 0.156 | 0.160 | 0.123 | 0.157 | 0.155 | 0.155 | 0.171 | 0.159 | 0.146 | 0.151 |
| [26]             | 0.179           | 0.172 | 0.185 | 0.155 | 0.173 | 0.189 | 0.165 | 0.168 | 0.185 | 0.176 | 0.194 | 0.199 | 0.182 | 0.188 | 0.194 | 0.145 | 0.170 | 0.188 | 0.175 | 0.208 | 0.180 | 0.188 | 0.180 |
| Ours             | 0.124           | 0.130 | 0.133 | 0.113 | 0.137 | 0.146 | 0.119 | 0.129 | 0.145 | 0.142 | 0.155 | 0.159 | 0.129 | 0.148 | 0.152 | 0.110 | 0.141 | 0.145 | 0.146 | 0.160 | 0.141 | 0.134 | 0.138 |

**Table 10:** Root Mean Square Error (RMSE) of the compared methods in AEMO-H dataset.



**Fig. 6:** Day-ahead wind energy generation measurements (ground truth) and predictions for station VI of the AEMO-H dataset. Time period: 01-07/07/13.

**ENTSO-E wind/solar:** The MAE and RMSE values achieved by the proposed and the baseline methods in ENTSO-E wind and solar datasets are reported in Table 11. The proposed method attained the best results, for both evaluation metrics. In ENTSO-E wind, it achieved top MAE and RMSE values in normalized data, namely 0.065 and 0.087 respectively. TiDE and the method in [26] achieved also good results in both metrics. Similarly in ENTSO-E solar, our proposed method achieved the best performance (0.035 MAE and 0.072 RMSE) in normalized data. Again, TiDE and the method proposed in [26] attained favourable results.

## 4.5 Discussion

Overall, the proposed wind and solar energy forecasting method achieved the best performance in terms of average - across the energy generation stations- MAE and RMSE. This superiority with respect to the competing methods was manifested in all five datasets. In addition, when measuring the performance on each energy generation site separately, the proposed method achieved top results on most of the cases. The obtained results highlight the ability of the method to generate enriched latent representations for energy generation data and weather data by exploring spatio-temporal patterns among them. It shall be also noted that the temporal processing of weather data in the proposed method, facilitated by the UTCAE, consistently demonstrates enhanced performance with respects to our preliminary approach presented in [26].

Aiming to demonstrate the efficacy of the proposed methodology in establishing meaningful spatial relationships between the locations of energy generation stations and the locations of weather data, we compared for the AEMO-H dataset the spatial attention weights matrix illustrated in Figure 7 with the corresponding geographical locations depicted in Figure 8. Indeed, the method forms significant spatial relations between neighboring stations such as stations a) I, III, XIII, XXII b) VI, IX, c) XI,

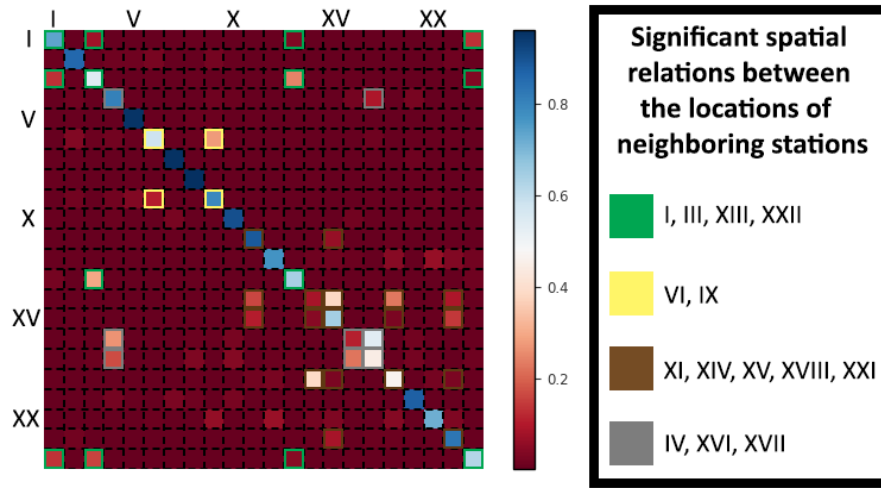


| Method           | Wind                  |                       | Solar                |                       |
|------------------|-----------------------|-----------------------|----------------------|-----------------------|
|                  | MAE<br>(nrm. / MW)    | RMSE<br>(nrm. / MW)   | MAE<br>(nrm. / MW)   | RMSE<br>(nrm. / MW)   |
| TCNN             | 0.203 / 532.60        | 0.251 / 660.21        | 0.063 / 134.15       | 0.123 / 260.94        |
| N-HITS           | 0.141 / 370.53        | 0.190 / 498.62        | 0.060 / 126.67       | 0.112 / 237.44        |
| Transformer      | 0.135 / 354.47        | 0.180 / 474.12        | 0.058 / 123.36       | 0.111 / 235.31        |
| Kalman Filtering | 0.108 / 284.37        | 0.155 / 408.19        | 0.143 / 301.87       | 0.237 / 502.24        |
| Random Forest    | 0.116 / 305.34        | 0.157 / 413.61        | 0.114 / 240.74       | 0.209 / 442.07        |
| DeepAR-based RNN | 0.093 / 245.81        | 0.132 / 346.13        | 0.095 / 201.10       | 0.175 / 370.48        |
| DLinear          | 0.107 / 281.29        | 0.145 / 381.33        | 0.062 / 130.47       | 0.105 / 222.74        |
| NLinear          | 0.110 / 289.70        | 0.147 / 387.54        | 0.061 / 128.10       | 0.104 / 219.45        |
| TiDE             | 0.078 / 206.14        | 0.105 / 275.28        | 0.040 / 83.83        | 0.078 / 164.49        |
| TFT              | 0.084 / 221.25        | 0.116 / 304.29        | 0.063 / 134.15       | 0.123 / 260.94        |
| [26]             | 0.076 / 200.71        | 0.102 / 267.83        | 0.040 / 84.64        | 0.080 / 168.53        |
| Ours             | <b>0.065 / 170.37</b> | <b>0.087 / 229.04</b> | <b>0.035 / 73.70</b> | <b>0.072 / 151.47</b> |

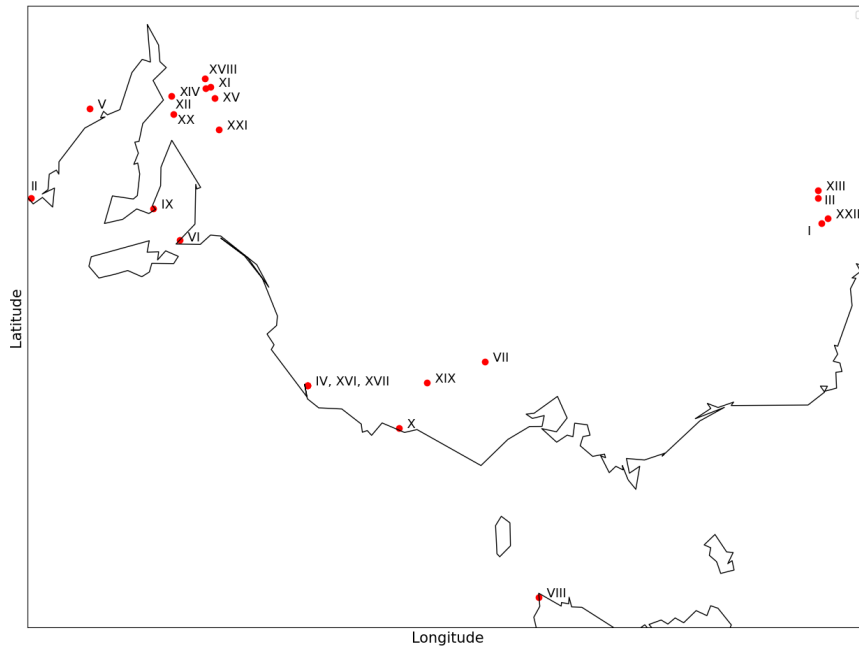
**Table 11:** Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) of the compared methods in ENTSO-E wind/solar dataset, for both normalized and raw data.

XIV, XV, XVIII, XXI and d) IV, XVI, XVII. To aid the reader, the attention weights corresponding to these stations are marked in the figure with outlines of different colors.

Moreover, in order to thoroughly evaluate the importance of spatial relations, we conducted a comprehensive examination of the proposed methodology across three distinct scenarios, using data from the AEMO-H dataset. In all these scenarios, the proposed method is applied for day-ahead energy generation forecasting at eleven stations, namely stations IV, XIII-XXII, of this datasets. The first scenario aimed at analysing the method’s performance in the case where weather data for the locations of the energy stations are available. Thus, in this scenario, weather data for the locations of stations IV, XII-XXII are provided. The competing methods, namely Kalman Filtering, DeepAR-based RNN, TiDE and TFT, were also applied under this scenario. The second scenario was designed so as to assess the proposed method’s ability to discover spatial relations when the locations of the weather data don’t align with these of the energy stations. Thus, in this scenario, weather data for locations corresponding to the positions of stations I-III and V-XII (namely stations that are not included in the experiment) are provided. In the third scenario, the examination involves weather data for the locations of the energy stations, as well as supplementary weather data, for additional locations. Thus, weather data for locations corresponding to the positions of all AEMO-H stations, namely stations I-XXII are provided.



**Fig. 7:** Spatial attention weights matrix, computed by the proposed method for the AEMO-H dataset. The locations of the energy stations, represented as rows, coincide with the weather data locations, represented as columns.



**Fig. 8:** Locations of the 22 energy generation stations in AEMO-H (south-east Australia). The weather data locations coincide with the locations of the stations.

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural Computing and Applications, and is available online at <https://doi.org/10.1007/s00521-024-10607-2>

As illustrated in Table 12, the proposed method attains top MAE (0.099) and RMSE (0.142) when weather data are available for the positions of the energy stations. A similar performance was also achieved when utilizing weather data corresponding to the positions of the aforementioned stations, as well as from locations corresponding to stations not involved in the associated evaluation (scenario 3). In the case where weather data were provided for locations which do not coincide with the positions of the energy stations (scenario 2), the proposed method yielded MAE and RMSE of 0.109 and 0.152, respectively. Thus, in this challenging case, the method managed to attain a decent performance, being third behind the DeepAR-based RNN and TFT, which, as mentioned above, utilized weather data from the locations of the stations. These findings highlight the significance of weather data, especially when their locations coincide with the positions of the energy stations. Results also show that the proposed method can perform well when provided with weather data for locations in close proximity to the energy stations.

| Method           | Weather Data for Specified Stations | MAE          | RMSE         |
|------------------|-------------------------------------|--------------|--------------|
| Kalman Filtering | IV, XII-XXII (scenario 1)           | 0.121        | 0.162        |
| DeepAR-based RNN | IV, XII-XXII (scenario 1)           | 0.102        | 0.146        |
| TiDE             | IV, XII-XXII (scenario 1)           | 0.125        | 0.171        |
| TFT              | IV, XII-XXII (scenario 1)           | 0.106        | 0.149        |
| Ours             | IV, XII-XXII (scenario 1)           | <b>0.099</b> | 0.142        |
|                  | I-III, V-XII (scenario 2)           | 0.109        | 0.152        |
|                  | I-XXII (scenario 3)                 | <b>0.099</b> | <b>0.141</b> |

**Table 12:** Average Mean Absolute Error (MAE) and average Root Mean Square Error (RMSE) for the top-five compared methods, in day-ahead energy forecasting for stations IV and XII-XXII of the AEMO-H dataset. The proposed methodology undergoes evaluation in three distinct scenarios, considering the spatial availability of the provided weather data (second column).

## 5 Conclusions

The efficient integration of Renewable Energy Sources (RES) into existing power systems highly depends on the accuracy of energy generation forecasting. This paper proposes a novel, deterministic multi-site and multi-step (i.e., for multiple time instances) wind and solar energy generation forecasting methodology, which utilizes past-time energy generation measurements as well as past- and future-time weather data from multiple locations. The weather data locations and the locations of the energy generation stations may not coincide with each other and no information, such as geolocation coordinates, needs to be provided. The proposed method utilizes a U-shaped Temporal Convolutional Auto-Encoder architecture to process time-series data related to weather and energy generation across each site. Furthermore, inspired

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural Computing and Applications, and is available online at <https://doi.org/10.1007/s00521-024-10607-2>

---

by the multi-head scaled-dot product attention mechanism, we introduce the Multi-sized Kernels convolutional Spatio-Temporal Attention, in order to effectively transfer temporal patterns from weather data to energy data. The results of the conducted experimental evaluation, in a day-ahead forecasting scenario, on three wind and two solar energy generation forecasting datasets, highlight the potential of the proposed method and its superiority against eleven state-of-the-art time-series forecasting methods. Future extensions will focus on validating the effectiveness of the method on various forecasting horizons, temporal resolutions, and datasets where the locations of the weather data don't correspond with the locations of the energy stations.

**Acknowledgments.** This work is co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE (project code: T2EDK-03048).

## Declarations

**Code availability.** The code used for the conducted experiments can be found at: <https://github.com/charsyme/Efficient-Deterministic-Renewable-Energy-Forecasting-Guided-by-Multiple-Location-Weather-Data>.

**Data availability.** The following datasets are available:  
GEFCom2014-wind and GEFCom2014-solar datasets are available, at <https://dx.doi.org/10.1016/j.ijforecast.2016.02.001>.

AEMO dataset is publicly available at <https://pureportal.strath.ac.uk/en/publications/very-short-term-probabilistic-wind-power-forecasts-by-sparse-vect>. The weather data necessary for reproducing the AEMO-H dataset can be obtained from the the Solcast online service at <https://solcast.com/>, utilizing the geolocation coordinates of the stations featured in the AEMO dataset.

The energy generation data of ENTSO-E wind/solar datasets can be obtained from the European Network of Transmission System Operators for Electricity at <https://transparency.entsoe.eu>. The weather data necessary for reproducing the aforementioned datasets can be obtained from the OpenWeather online service at <https://openweathermap.org/>, utilizing the geolocation coordinates of the 29 locations.

## References

- [1] Höök, M., Tang, X.: Depletion of fossil fuels and anthropogenic climate change—a review. *Energy Policy* **52**, 797–809 (2013)
- [2] Yoro, K.O., Daramola, M.O.: CO<sub>2</sub> emission sources, greenhouse gases, and the global warming effect. *Advances in Carbon Capture*, 3–28 (2020)
- [3] Rahman, A., Farrok, O., Haque, M.M.: Environmental impact of renewable energy source based electrical power plants: Solar, wind, hydroelectric, biomass,

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in *Neural Computing and Applications*, and is available online at <https://doi.org/10.1007/s00521-024-10607-2>

- 
- geothermal, tidal, ocean, and osmotic. *Renewable and Sustainable Energy Reviews* **161**, 112279 (2022)
- [4] Widén, J., Carpmann, N., Castellucci, V., Lingfors, D., Olauson, J., Remouit, F., Bergkvist, M., Grabbe, M., Waters, R.: Variability assessment and forecasting of renewables: A review for solar, wind, wave and tidal resources. *Renewable and Sustainable Energy Reviews* **44**, 356–375 (2015)
  - [5] Tzelepi, M., Symeonidis, C., Nousi, P., Kakaletsis, E., Manousis, T., Tosidis, P., Nikolaidis, N., Tefas, A.: Deep learning for energy time-series analysis and forecasting. *arXiv preprint arXiv:2306.09129* (2023)
  - [6] Stentoumi, O., Nousi, P., Tzelepi, M., Tefas, A.: AnIO: anchored input–output learning for time-series forecasting. *Neural Computing and Applications*, 1–11 (2023)
  - [7] Lorenz, E., Scheidsteger, T., Hurka, J., Heinemann, D., Kurz, C.: Regional PV power prediction for improved grid integration. *Progress in Photovoltaics: Research and Applications* **19**(7), 757–771 (2011)
  - [8] Jørgensen, J., Möhrle, C.: HONEYMOON-A high resolution numerical wind energy model for on-and offshore forecasting using ensemble predictions. *Citeseer* (2005)
  - [9] Rajagopalan, S., Santoso, S.: Wind power forecasting and error analysis using the autoregressive moving average modeling. In: *IEEE Power & Energy Society General Meeting*, pp. 1–6 (2009). IEEE
  - [10] Atique, S., Noureen, S., Roy, V., Subburaj, V., Bayne, S., Macfie, J.: Forecasting of total daily solar energy generation using arima: A case study. In: *IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0114–0119 (2019)
  - [11] Carpinone, A., Langella, A. R. and Testa, Giorgio, M.: Very short-term probabilistic wind power forecasting based on markov chain models. In: *2010 IEEE 11th International Conference on Probabilistic Methods Applied to Power Systems*, pp. 107–112 (2010)
  - [12] Bai, X., Liang, X. L. and Zhu: Improved markov-chain-based ultra-short-term PV forecasting method for enhancing power system resilience. *The Journal of Engineering* **2021**(2), 114–124 (2021)
  - [13] Dong, X., Sun, Y., Li, Y., Wang, X., Pu, T.: Spatio-temporal convolutional network based power forecasting of multiple wind farms. *Journal of Modern Power Systems and Clean Energy* **10**(2), 388–398 (2022)
  - [14] Hossain, M.S., Mahmood, H.: Short-term photovoltaic power forecasting using an

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in *Neural Computing and Applications*, and is available online at <https://doi.org/10.1007/s00521-024-10607-2>

---

LSTM neural network and synthetic weather forecast. *IEEE Access* **8**, 172524–172533 (2020)

- [15] Li, L.-L., Zhao, X., Tseng, M.-L., Tan, R.R.: Short-term wind power forecasting based on support vector machine with improved dragonfly algorithm. *Journal of Cleaner Production* **242**, 118447 (2020)
- [16] Fan, J., Wang, X., Wu, L., Zhou, H., Zhang, F., Yu, X., Lu, X., Xiang, Y.: Comparison of support vector machine and extreme gradient boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: A case study in china. *Energy conversion and management* **164**, 102–111 (2018)
- [17] Ahmed, A., Khalid, M.: A review on the selected applications of forecasting models in renewable power systems. *Renewable and Sustainable Energy Reviews* **100**, 9–21 (2019)
- [18] Aghajani, A., Kazemzadeh, R., Ebrahimi, A.: A novel hybrid approach for predicting wind farm power production based on wavelet transform, hybrid neural networks and imperialist competitive algorithm. *Energy conversion and management* **121**, 232–240 (2016)
- [19] Vrettos, E., Gehbauer, C.: A hybrid approach for short-term PV power forecasting in predictive control applications. In: 2019 IEEE Milan PowerTech, pp. 1–6 (2019)
- [20] Zhewen, N., Zeyuan, Y., Wenhui, T., Qinghua, W., R., M.: Wind power forecasting using attention-based gated recurrent unit network. *Energy* **196**, 117081 (2020)
- [21] Rai, A., Shrivastava, A., Jana, K.C.: A robust auto encoder-gated recurrent unit (AE-GRU) based deep learning approach for short term solar power forecasting. *Optik* **252**, 168515 (2022)
- [22] Hagan, K.E., Oyebanjo, O.O., Masaud, T.M., Challoo, R.: A probabilistic forecasting model for accurate estimation of PV solar and wind power generation. In: IEEE Power and Energy Conference at Illinois (PECI), pp. 1–5 (2016)
- [23] Dowell, J., Pinson, P.: Very-short-term probabilistic wind power forecasts by sparse vector autoregression. *IEEE Transactions on Smart Grid* **7**(2), 763–770 (2016)
- [24] Hanifi, S., Liu, X., Linand, Z., Lotfian, S.: A critical review of wind power forecasting methods—past, present and future. *Energies* **13** (2020)
- [25] Hong, T., Pinson, P., Wang, Y., Weron, R., Yang, D., Zareipour, H.: Energy forecasting: A review and outlook. *IEEE Open Access Journal of Power and Energy* **7**, 376–388 (2020)

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in *Neural Computing and Applications*, and is available online at <https://doi.org/10.1007/s00521-024-10607-2>

- 
- [26] Symeonidis, C., Nikolaidis, N.: Wind energy prediction guided by multiple-location weather forecasts. In: *Engineering Applications of Neural Networks*, pp. 445–457 (2023)
  - [27] Ruijin, Z., Wenlong, L., Yusen, W.: Short-term prediction for wind power based on temporal convolutional network. *Energy Reports* **6**, 424–429 (2020)
  - [28] Qu, K., Si, G., Shan, Z., X.Kong, Yang, X.: Short-term forecasting for multiple wind farms based on transformer model. *Energy Reports* **8**, 483–490 (2022)
  - [29] Liu, Y., Guan, L., Hou, C., Han, H., Liu, Z., Sun, Y., Zheng, M.: Wind power short-term prediction based on LSTM and discrete wavelet transform. *Applied Sciences* **9**(6) (2019)
  - [30] Lin, Y., Koprinska, I., Rana, M.: Temporal convolutional neural networks for solar power forecasting. In: *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8 (2020)
  - [31] Jung, Y., Jung, J., Kim, B., Han, S.: Long short-term memory recurrent neural network for modeling temporal patterns in long-term power forecasting for solar PV facilities: Case study of south korea. *Journal of Cleaner Production* **250**, 119476 (2020)
  - [32] Jeong, J., Kim, H.: Multi-site photovoltaic forecasting exploiting space-time convolutional neural network. *Energies* **12**(23), 4490 (2019)
  - [33] Simeunović, J., Schubnel, B., Alet, P.-J., Carrillo, R.E., Frossard, P.: Interpretable temporal-spatial graph attention network for multi-site PV power forecasting. *Applied Energy* **327**, 120127 (2022)
  - [34] Shih, S.-Y., Sun, F.-K., Lee, H.-Y.: Temporal pattern attention for multivariate time series forecasting. *Machine Learning* **108**, 1421–1441 (2019)
  - [35] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, .L., Polosukhin, .I.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
  - [36] Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., Yan, X.: Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: *Advances in Neural Information Processing Systems*, vol. 32 (2019)
  - [37] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18, pp. 234–241 (2015)

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in *Neural Computing and Applications*, and is available online at <https://doi.org/10.1007/s00521-024-10607-2>

- 
- [38] Perslev, M., Jensen, M., Darkner, S., Jennum, P.J., Igel, C.: U-time: A fully convolutional network for time series segmentation applied to sleep staging. *Advances in Neural Information Processing Systems* **32** (2019)
- [39] Madhusudhanan, K., Burchert, J., Duong-Trung, N., Born, S., Schmidt-Thieme, L.: U-Net inspired transformer architecture for far horizon time series forecasting. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 36–52 (2022)
- [40] Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., Hyndman, R.J.: Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. *International Journal of Forecasting* **32**(3), 896–913 (2016)
- [41] Vartholomaos, A., Karlos, S., Kouloumpis, E., Tsoumakas, G.: Short-term renewable energy forecasting in greece using prophet decomposition and tree-based ensembles. In: *International Conference on Database and Expert Systems Applications*, pp. 227–238 (2021)
- [42] Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., Choo, J.: Reversible instance normalization for accurate time-series forecasting against distribution shift. In: *International Conference on Learning Representations* (2021)
- [43] Herzen, J., Lässig, F., Piazzetta, S.G., Neuer, T., Tafti, L., Raille, G., Van Pottelbergh, T., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., Kościsz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., Grosch, G.: Darts: User-friendly modern machine learning for time series. *The Journal of Machine Learning Research* **23**(1), 5442–5447 (2022)
- [44] Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Ben-Tzur, J., Hardt, M., Recht, B., Talwalkar, A.: A system for massively parallel hyperparameter tuning. *Proceedings of Machine Learning and Systems* **2**, 230–246 (2020)
- [45] Breiman, L.: Random forests. *Machine learning* **45**, 5–32 (2001)
- [46] Morrison, G.W., Pike, D.H.: Kalman filtering applied to statistical forecasting. *Management Science* **23**(7), 768–774 (1977)
- [47] Challu, C., Olivares, K.G., Oreshkin, B.N., Garza Ramirez, F., Mergenthaler Canseco, M., Dubrawski, A.: NHITS: Neural hierarchical interpolation for time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* **37**(6), 6989–6997 (2023)
- [48] Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271* (2018)
- [49] Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T.: Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of*

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in *Neural Computing and Applications*, and is available online at <https://doi.org/10.1007/s00521-024-10607-2>



---

Forecasting **36**(3), 1181–1191 (2020)

- [50] Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? Proceedings of the AAAI Conference on Artificial Intelligence **37**(9), 11121–11128 (2023)
- [51] Das, A., Kong, W., Leach, A., Mathur, S.K., Sen, R., Yu, R.: Long-term forecasting with TiDE: Time-series dense encoder. Transactions on Machine Learning Research (2023)
- [52] Lim, B., Arık, S.O., Loeff, N., Pfister, T.: Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting **37**(4), 1748–1764 (2021)

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this article is published in Neural Computing and Applications, and is available online at  
<https://doi.org/10.1007/s00521-024-10607-2>