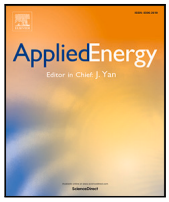
Applied Energy 357 (2024) 122502

Contents lists available at ScienceDirect

Applied Energy

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



Temporal collaborative attention for wind power forecasting

Yue Hu [a], Hanjing Liu [b], Senzhen Wu [b], Yuan Zhao [d], Zhijin Wang [b,*], Xiufeng Liu [c,*]

[a] Chengyi College, Jimei University, Jimei Road 199, 361021 Xiamen, China
[b] College of Computer Engineering, Jimei University, Yinjiang Road 185, 361021 Xiamen, China
[c] Department of Technology, Management and Economics, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark
[d] School of Business Administration, South China University of Technology, Wushan Road 381, 510630, Guangzhou, China

ARTICLE INFO

Keywords:
Wind power forecasting
Temporal collaborative attention
Attention mechanism
Multivariate time series
Data-driven method

ABSTRACT

Wind power serves as a clean and sustainable form of energy. However, its generation is fraught with variability and uncertainty, owing to the stochastic and dynamic characteristics of wind. Accurate forecasting of wind power is indispensable for the efficient planning, operation, and grid integration of wind energy systems. In this paper, we introduce a novel forecasting method termed Temporal Collaborative Attention (TCOAT). This data-driven approach is designed to capture both temporal and spatial dependencies in wind power generation data, as well as discern long-term and short-term patterns. Utilizing attention mechanisms, TCOAT dynamically adjusts the weights of each input variable and time step based on their contextual relevance for forecasting. Furthermore, the method employs collaborative attention units to assimilate directional and global information from the input data. It also explicitly models the interactions and correlations among different variables or time steps through the use of self-attention and cross-attention mechanisms. To integrate long-term and short-term information effectively, TCOAT incorporates a temporal fusion layer that employs concatenation and mapping operations, along with hierarchical feature extraction and aggregation. We validate the efficacy of TCOAT through extensive experiments on a real-world wind power generation dataset from Greece and compare its performance against twenty-two state-of-the-art methods. Experimental results demonstrate that TCOAT outperforms existing methods in terms of both accuracy and robustness in wind power forecasting. Moreover, we conduct a generality study on an additional real-world dataset from a different climate condition and wind power characteristics. The results show that TCOAT can achieve comparable or better performance than the state-of-the-art methods, confirming the generalization ability of TCOAT.

1. Introduction

Wind power is a renewable and clean energy source that has been growing rapidly in recent years, reaching a global capacity of 934 GW in 2022 [1]. However, wind power generation is also fluctuating and intermittent, due to the stochastic and dynamic nature of wind speed and direction, as well as other meteorological factors [2]. These characteristics pose a threat to the stability of electric power systems and hinder the effective application and management of wind energy [3]. Therefore, forecasting wind power accurately is crucial for the planning, operation, and integration of wind energy systems into the power grid [4]. Wind power forecasting (WPF) is the task of estimating the amount of wind power that can be produced by one or more wind turbines or wind farms in a given location and time period [5]. Wind power forecasting can help wind farm operators and grid managers to plan ahead, adjust the power supply and demand, reduce the risk of power outages or curtailments, and optimize the economic benefits. WPF can be categorized into different types according to the time horizon, such as long-term (more than one year), medium-term (one month to one year), short-term (one day to one week), and very short-term (less than one day) [3]. Among these types, short-term and very short-term WPF are the most important and challenging ones, as they have direct implications for the economic and technical aspects of wind power integration, such as scheduling, dispatching, balancing, and market participation [6].

WPF confronts a multitude of challenges that stem from the inherent complexities of wind power generation data. These challenges include nonlinearity, nonstationarity, high-dimensionality, and uncertainty. Nonlinearity in wind power generation is manifested through intricate relationships with a variety of input variables such as wind speed, wind direction, air density, turbine characteristics, and terrain features. As Duan et al. [7] have demonstrated, linear models and

* Corresponding author.
E-mail addresses: yuehu.xm@gmail.com (Y. Hu), hanjingliou@gmail.com (H. Liu), szwbyte@gmail.com (S. Wu), bmyzhao@outlook.com (Y. Zhao), zhijinecnu@gmail.cn (Z. Wang), xiuli@dtu.dk (X. Liu).

https://doi.org/10.1016/j.apenergy.2023.122502
Received 7 September 2023; Received in revised form 20 November 2023; Accepted 12 December 2023

rudimentary statistical methods often fall short in capturing these nonlinear relationships, leading to forecasts that are either suboptimal or biased. To mitigate this issue, the adoption of nonlinear modeling techniques capable of learning complex data patterns is imperative. The temporal variations and seasonal changes in wind power generation introduce another layer of complexity, termed nonstationarity. The study [8] indicates that such variations can alter the data distribution and the underlying dynamics over time, thereby affecting the reliability of models that assume constant parameters or stable conditions. Consequently, there is a pressing need for adaptive modeling approaches that can adjust to these changing conditions to yield consistent forecasts. High-dimensionality is another significant challenge, as wind power generation is influenced by a multitude of factors that operate across different spatial and temporal scales. These include weather conditions, geographical locations, turbine configurations, and grid conditions. As Karamichailidou et al. [9] have noted, the sheer volume of data generated from these multiple sources complicates data processing and analysis. Therefore, models capable of handling high-dimensional data are essential for effective forecasting. Wind power generation is affected by various sources of uncertainty, such as measurement errors, model errors, parameter errors, and forecast errors [10]. These sources of uncertainty may introduce errors or deviations in the forecasts, affecting the decision-making and risk management of wind power integration. To address this difficulty, uncertainty quantification and propagation techniques can be used to provide probabilistic forecasts or confidence intervals, such as [11,12].

In recent decades, a plethora of methods have been developed to tackle the challenges associated with WPF, as evidenced by a range of seminal works [13–16]. These works can be broadly categorized into three categories: physical methods, statistical methods, and hybrid methods. Physical methods are based on numerical weather prediction (NWP) models that simulate atmospheric dynamics and physics using mathematical equations [8]. Physical methods primarily rely on numerical weather prediction (NWP) models, which employ mathematical equations to simulate atmospheric dynamics and physics [8]. While these physically-based models, such as NWP systems [17,18], are adept at providing detailed long-term wind forecasts, they are computationally intensive and may suffer from inaccuracies due to model simplifications. Moreover, the computational burden becomes particularly pronounced during the downscaling process [19]. Statistical methods, on the other hand, employ data-driven models to learn empirical relationships between input variables, such as historical wind and meteorological data, and output variables like future wind power generation [5]. These models are generally efficient for short-term forecasts but are susceptible to overfitting and may exhibit reduced accuracy over extended prediction horizons [20]. Traditional statistical models like the Persistence Model (PM) and various Autoregressive (AR) types, including ARMA and ARIMA, have been adapted to account for the non-stationary and complex nature of wind speeds. For example, Fractional ARIMA models have demonstrated significant improvements in 24-hour and 48-hour wind speed forecasts compared to the PM [21]. However, these models often make the simplifying assumption of Gaussian distributions in wind speed data, which may not always hold true [22]. Hybrid methods aim to amalgamate the strengths of both physical and statistical approaches, thereby providing robust and reliable forecasts across various time horizons [8]. Despite their potential, these methods may encounter challenges in effectively integrating disparate models or data sources [23].

Nevertheless, existing methods still have some limitations and drawbacks. One of them is that most of the methods do not consider the importance and relevance of each input variable or time step. They treat all the input data equally or use fixed weights, which may lead to poor or biased forecasts. For example, ARIMA [2], ELM [3] and SVR models [5], do not differentiate the importance of different wind fields, time steps, or meteorological factors, respectively. Another notable limitation is the inadequate exploitation of collaborative and directional information present in the input data. Many existing methods overlook the interactions and correlations among different variables or time steps, leading to forecasts that may be either incomplete or redundant. For instance, both the ARIMA and SVR models fail to capture the directional nuances of wind speed and direction, as well as the interactions among different input variables or geographical locations. Moreover, a majority of the methods exhibit a lack of effective integration between long-term and short-term information from the input data. They either focus on one of these aspects to the exclusion of the other or resort to simplistic concatenation techniques, potentially resulting in information loss or inconsistencies in the forecasts. For example, ELM networks and SVR methods primarily concentrate on long-term information derived from NWP equations or short-term information from Multi-Temporal Scale (MTS) models, or they attempt to combine both types of information through rudimentary concatenation. Furthermore, most of existing methods do not consider the spatial dependencies among different wind turbines or wind farms, which may affect the accuracy and reliability of the forecasts. These methods often rely on fixed or predefined weights for each input variable or time step, which may not reflect their contextual relevance for forecasting. Therefore, there is a need for a novel method that can address these challenges and provide accurate and robust forecasts for WPF.

In this paper, we aim to answer the following research problem statement: How to accurately forecast wind power generation using a data-driven method that can capture the temporal and spatial dependencies, as well as the long-term and short-term patterns in the data, and that can dynamically adjust the weights of each input variable and time step based on their contextual relevance for forecasting? The research objectives of this paper are: (1) To develop a novel data-driven method for WPF that can accurately forecast wind power generation by capturing the temporal and spatial dependencies, as well as the long-term and short-term patterns in the data. (2) To explore how the proposed method can dynamically adjust the weights of each input variable and time step based on their contextual relevance for forecasting, and how this can improve the representation and interpretation of the data.

To address these objectives, we propose Temporal Collaborative Attention (TCOAT), a novel method for WPF that advances the state-of-the-art in several ways. First, it is the first method that integrates temporal collaborative attention with temporal fusion for WPF, which can capture both the temporal and spatial dependencies, as well as the long-term and short-term patterns in the data. Existing methods either use fixed or predefined weights for each input variable or time step, or use simple attention mechanisms that do not consider the directional information or the global information in the data. Second, it introduces collaborative attention units (CAUs), which can transform the input data into a tensorial representation capable of capturing directional dependencies, and computing attention scores and memory weights for each tensor direction. CAUs can model the interactions and correlations among different variables or time steps using self-attention and cross-attention, and can enhance the representation and interpretation of the data. Existing methods either do not use attention mechanisms, or use single-directional or single-dimensional attention mechanisms that do not capture the complex relationships in the data. Third, it designs a temporal fusion layer, which can effectively integrate the long-term and short-term information from the data, and fuse them using concatenation and mapping operations and hierarchical feature extraction and aggregation. The temporal fusion layer can capture both global and local data characteristics, and can extract hierarchical features for WPF. Existing methods either do not use temporal fusion, or use simple concatenation or averaging techniques that may result in information loss or inconsistencies in the data. TCOAT is an end-to-end model that can learn directly from raw wind power data without any preprocessing or post-processing steps.

The main contributions of this paper are:

- We propose TCOAT, a novel method for WPF that uses attention mechanisms to capture the temporal and spatial dependencies in wind power generation data, and to dynamically adjust the weights of each input variable and time step according to their relevance for forecasting.
- We introduce CAUs, a novel component of TCOAT that can learn the directional information and the global information from the data, and model the interactions and correlations among different variables or time steps using self-attention and cross-attention.
- We design a temporal fusion layer, a novel component of TCOAT that can effectively integrate the long-term and short-term information from the data, and fuse them using concatenation and mapping operations and hierarchical feature extraction and aggregation.
- We evaluate TCOAT's performance and generality on two real-world wind power generation datasets from different climate zones. One dataset is from Greece, which has a Mediterranean climate. The second dataset, while the precise location is not disclosed, is derived from a wind farm situated in a flat, inland terrain. We compare TCOAT with twenty-two state-of-the-art methods on various forecasting tasks and metrics. The results demonstrate that TCOAT outperforms existing methods in terms of accuracy, generality and robustness for WPF.

The rest of this paper is organized as follows: Section 2 presents related work. Section 3 describes the study materials. Section 4 presents the proposed TCOAT model. Section 5 describes the experimental settings, reports, and discussions on the experimental results. Section 6 concludes this paper and outlines some future research directions.

2. Related work

2.1. Wind power forecasting models

Wind power forecasting is essential for the integration and operation of wind energy in power systems. It can help optimize the scheduling and dispatch of power generation, reduce the uncertainty and variability of wind power, and enhance the reliability and security of the grid. Wind power forecasting can be classified into four categories according to the forecasting horizon: very short-term (up to 6 h ahead), short-term (6 to 48 h ahead), medium-term (2 to 10 days ahead), and long-term (more than 10 days ahead) [24]. Different forecasting methods have different strengths and weaknesses, depending on the forecasting horizon, the spatial and temporal resolution, the input data, and the evaluation metrics. In this section, we review some of the main approaches for wind power forecasting, namely physical, statistical, and deep neural networks (DNNs) methods. We also discuss their advantages, disadvantages, and challenges.

2.1.1. Physical approaches

Physical models, employing numerical weather prediction (NWP) techniques, are crucial in wind power forecasting. These models are based on atmospheric physics and solve equations of fluid dynamics and thermodynamics. They can account for complex terrain and environmental factors, and provide forecasts for multiple variables, such as wind speed, direction, temperature, and pressure. These variables can affect the power output and the fatigue and damage of wind turbines, as well as the power flow and congestion in transmission lines [25]. One of the most prominent examples of physical models is the Weather Research and Forecasting (WRF) model, which excels in medium to long-term forecasting. However, these models also have drawbacks, such as requiring high computational resources and extensive meteorological data, which can hamper their real-time applications.

Several studies have applied and compared the performance of physical models for wind power forecasting, using different input data, forecasting horizons, and evaluation metrics. For example, Jacondino et al. [26] compared different physics schemes for wind forecasting in Brazil using WRF. They found that the best setup used a local closure PBL, a single-moment microphysics, a two-layer land surface, a profile adjustment cumulus, and cloud-aerosol radiation. Wang et al. [27] proposed a method to correct the wind forecast of the WRF model using random forest (RF) and machine learning (ML) techniques. They used WRF output, GTS observation and ERA5 reanalysis data as inputs, and evaluated the forecasts using RMSE and spatial distribution. They found that the RF-based method improved the average forecast accuracy of 10 m wind, 2 m temperature and sea level pressure by 40%, 36%, and 50%, respectively, compared to the original WRF model. They also found that adding a MLP-based feature selector to the RF model further improved the accuracy by 5%. Zheng et al. [28] used the WRF-RF model for short-term wind power prediction at a wind farm in China. They used data from a NWP model and a wind tower as inputs, and evaluated the forecasts using RMSE and MAPE. They found that the WRF-RF model improved the accuracy of wind power prediction, especially at higher wind speeds. Zhao et al. [29] created a hybrid method that combines a WRF ensemble, a fuzzy system, and a cuckoo search algorithm to forecast wind speed for wind farms. The method reduces NWP errors, selects and weighs the best ensemble members, and performs better than other models in different regions.

One of the advantages of physical models is that they can provide forecasts for any location, even where historical data is not available or sufficient. This is especially useful for remote or offshore wind farms, where data collection can be challenging or costly [30]. However, physical models also face some challenges and limitations. One of them is that they depend on the quality and availability of input data, such as initial and boundary conditions, which can introduce uncertainties and errors in the forecasts. For example, errors in the initial wind speed or direction can propagate and amplify over time, leading to inaccurate forecasts [31]. Another challenge is that they require high-resolution spatial and temporal grids, which can increase the computational complexity and cost of the models. This can limit the applicability of physical models for short-term or very short-term forecasting, where fast and frequent updates are needed [32]. Moreover, physical models may not account for local effects, such as topography, land use, and vegetation, which can influence wind power production at specific sites. These effects can be difficult to model or parameterize, and may require site-specific calibration or validation [33]. Finally, physical models may not be able to capture the stochastic and nonlinear nature of wind power fluctuations, especially in short-term horizons. These fluctuations can be caused by random or chaotic phenomena, such as gusts, ramps, or cut-offs, which can be hard to predict or simulate [34].

2.1.2. Statistical approaches

Statistical models, such as the Auto Regressive Moving Average (ARMA) and its extension, the Auto Regressive Integrated Moving Average (ARIMA), are widely utilized in wind power forecasting. These models are based on linear regression of observed values, and can handle time series data effectively. The ARIMA model, in particular, addresses the non-stationarity of wind data, making it more adaptable to varied forecasting scenarios. However, these models require high-quality, stationary historical data to maintain accuracy, which may limit their applicability in some conditions.

Several studies have applied and compared the performance of ARMA and ARIMA models for wind power forecasting, using different input data, forecasting horizons, and evaluation metrics. For example, Cao et al. [35] combined the ARMA model for forecasting up to one hour ahead, and the pattern-matching method for forecasting up to six hours ahead. They found that the ARMA model had better accuracy in shorter time scales, while the pattern-matching method was more accurate for longer time scales. Milligan et al. [36] tested various alternative ARMA models for up to six hours forecasting horizon, and found that the ARMA (1,24) model had the best performance of all

the models tested. They also observed that the forecasting accuracy decreased significantly for greater forecasting horizons, and that the ARMA models managed to surpass the persistence model in most cases. Ahn et al. [37] proposed a short-term wind power forecasting method using an ensemble model based on wavelet transform and ARIMAX techniques. They claimed that their method outperformed the single ARIMAX model and other benchmark models in terms of accuracy and reliability. Zhang et al. [38] proposed a hybrid model based on DWT, SARIMA and LSTM to forecast short-term offshore wind power. They used DWT to decompose the power signal into linear and nonlinear components, and applied SARIMA and LSTM to capture the seasonal and dynamic patterns, respectively. They achieved lower NMAE and NRMSE than using single models or other hybrid models. Sheoran and Pasari [39] forecasted wind speed from four Indian locations. They updated model parameters dynamically and compared with standard ARIMA and persistence models. They showed that window-sliding ARIMA was more accurate, robust, flexible, and efficient.

Statistical models have some advantages and drawbacks for wind power forecasting. One of the advantages is that they are simple, fast, and easy to implement. They can provide reliable forecasts for short-term horizons, such as minutes or hours ahead, which are useful for operational planning and scheduling. They can also capture the autocorrelation and seasonality of wind power data, which are important features for forecasting. Moreover, statistical models can be combined with other methods, such as physical models, machine learning models, or ensemble methods, to improve their performance and robustness. For example, Singh et al. [40] proposed a hybrid method that used ARIMA and artificial neural networks (ANNs) to forecast wind power for different time scales. Bazionis et al. [41] presented a critical review of various forecast models, including statistical models, and compared their performance indices. However, statistical models also have some drawbacks and challenges. One of them is that they ignore external factors that affect wind power generation, such as weather conditions, terrain features, or turbine characteristics. These factors can introduce uncertainties and errors in the forecasts, especially for longer-term horizons, such as days or weeks ahead. Another drawback is that statistical models are sensitive to outliers and noise in the data, which can distort the model parameters and reduce the forecast accuracy. Furthermore, statistical models may fail to capture complex and nonlinear patterns in the wind power data, such as ramps, gusts, or cut-offs, which can cause significant deviations from the expected values. These patterns can be influenced by random or chaotic phenomena, which are hard to model or predict by linear regression. For instance, Messner et al. [42] conducted a comprehensive review and statistical analysis of errors in wind power forecasts, and found that the error dispersion factor, which measures the variability of errors, depended on the size of the wind farm, the forecasting horizon, and the class of the forecasting method. Olson et al. [43] developed an empirical model that used inputs from a numerical weather prediction (NWP) model to forecast wind power, and compared it with a statistical model.

2.1.3. DNN-based approaches

Deep Neural Networks (DNNs) are composed of multiple interconnected layers of artificial neurons that can learn complex and nonlinear mappings between the input data and the output data [2,44]. DNNs can be classified into different types and categories based on their network structures and functions, such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Attention Mechanisms (AMs), Transformers, and Graph Neural Networks (GNNs) [45]. These types or categories of DNNs have different characteristics, strengths, and weaknesses for wind power forecasting, depending on the input features, the forecasting horizon, and the uncertainty quantification [46].

Several DNN-based methods have been proposed for wind power forecasting, which can be categorized according to their input features, network architectures, and forecasting horizons. Huang et al. [47] used LSTM, CNN, and fully connected layers to capture the long-term dependencies and local features in wind power data, achieving high accuracy and robustness for short-term forecasting. In contrast, Alcantara et al. [48] used CNN and fully connected layers to extract local and global features from spatial data and capture the spatial dependencies in wind power generation, achieving high accuracy and efficiency for medium-term forecasting. Liu et al. [49] used AM, LSTM, and fully connected layers to assign different weights to different inputs or time steps according to their relevance and importance for wind power forecasting, achieving high accuracy and scalability for long-term forecasting. Moreover, Sun et al. [50] used spatio-temporal correlations and transformer neural networks for short-term multi-step wind power forecasting, evaluating the quality of spatial information using distance- and correlation-based metrics, and modeling the wind power using multi-head attention mechanism. They outperformed several baseline and state-of-the-art methods in two case studies, but they ignored the wind direction, seasonal variation, and weather factors. Wu et al. [51] integrated multidimensional data and spatial correlations for wind speed forecasting, using a Wind Transformer to capture temporal features and a GNN [52] to aggregate spatial features. They outperformed existing methods in accuracy and stability. Liu et al. [53] forecasted the ultra-short-term power of wind farm cluster based on power fluctuation pattern recognition and spatio-temporal graph neural network, segmenting the power series into different patterns and training a separate model for each pattern, and capturing the dynamic spatio-temporal correlation between adjacent wind farms under different patterns. However, they had limitations such as high computational cost, fixed pattern partition, and lack of uncertainty quantification.

Compared to other methods, DNN-based wind power forecasting methods have both advantages and drawbacks. On the one hand, they can learn complex and nonlinear patterns and dependencies from historical data, without requiring explicit physical or statistical assumptions. They can also handle noisy, incomplete, or high-dimensional data, and adapt to changing conditions and scenarios. Moreover, they can be combined with other methods, such as physical models, statistical models, or optimization methods, to improve their performance and robustness. For instance, Lu et al. [54] proposed a hybrid method that used IVMD-SE data preprocessing, MC-LSTM predictor and PSO optimization to forecast short-term wind power, which can handle complex data characteristics and improve forecasting accuracy and robustness. Wu et al. [55] presented a comprehensive review on DNN-based approaches in wind forecasting applications, and categorized the existing methods into four types: RNN-based, RBM-based, CNN-based and AE-based models. They also discussed the advantages and disadvantages of each type, as well as the future research directions. On the other hand, deep learning models also have some limitations and challenges in wind forecasting. One of them is that they require high-quality and large-scale data, which may not be easily obtained or sufficient in some scenarios. Another limitation is that they are susceptible to overfitting and underfitting, which may affect their accuracy and robustness. Moreover, deep learning models are computationally expensive and time-consuming, and they need careful tuning of hyperparameters and network architectures. Furthermore, deep learning models are hard to interpret and explain, as they lack transparency and physical meaning. Therefore, it is important to address these issues and improve the performance and reliability of deep learning models in wind forecasting.

2.1.4. Summary

Statistical models, physical models, and DNN-based methods are three main categories of existing methods for wind power forecasting. However, each category has its own advantages and drawbacks, such as data requirements, computational complexity, accuracy, robustness, and interpretability. In this paper, we propose a novel method for wind power forecasting, TCOAT, that integrates temporal collaborative attention and temporal fusion, which can capture both the temporal

and spatial dependencies, as well as the long-term and short-term patterns in wind power generation data. TCOAT also introduces collaborative attention units (CAUs) and a temporal fusion layer, which can enhance the representation and interpretation of the data, and extract hierarchical features for wind power forecasting. TCOAT differs from the existing DNN-based methods in terms of the design and the components, and demonstrates the unique contributions of this work by providing accurate and robust forecasts for wind power forecasting. TCOAT consists of four main components: a temporal encoder, a spatial encoder, a collaborative attention module, and a transformer decoder. TCOAT can handle both sequential and spatial data, capture long-term and short-term dependencies, provide attention maps, and achieve state-of-the-art performance. TCOAT is an end-to-end model that can learn directly from raw wind power data without any preprocessing or post-processing steps.

2.2. Attention mechanisms in time series forecasting

Attention mechanisms have been widely adopted in natural language processing tasks [56,57], as they can effectively capture long-range dependencies in sequential data. This is especially beneficial for energy forecasting, where temporal patterns and relationships span across various time scales. In this subsection, we review the recent developments in applying attention mechanisms for energy forecasting, and critically examine their strengths and weaknesses.

We categorize the existing works that use attention mechanisms for energy forecasting into two groups: single-energy forecasting and multi-energy forecasting. Single-energy forecasting aims to forecast one type of energy source or load, such as wind power [58,59] or electrical load [60]. These works employ various deep learning architectures with attention mechanisms, such as CNNs, RNNs, LSTMs, or dual-attention mechanisms, to achieve high accuracy and robustness for single-energy forecasting tasks. However, they also face some limitations, such as neglecting external factors that may influence energy generation or demand, requiring large amounts of data, or being computationally expensive. Multi-energy forecasting targets to forecast multiple types of energy sources or loads simultaneously, such as electrical and thermal loads or wind and solar power [61–63]. These works propose novel methods for short-term or day-ahead multi-energy load forecasting based on a CNN-BiGRU architecture, a CNN-Seq2Seq model, or an attention mechanism-based transfer learning model. These works demonstrate high efficiency and accuracy for multi-energy forecasting tasks, but they also encounter some challenges, such as handling complex and large-scale datasets, accounting for seasonal variations in energy data, or ensuring the generalizability of the model to different energy systems. In addition to these works that directly focus on energy forecasting tasks, there are also some works that apply attention mechanisms for related tasks that could indirectly benefit energy forecasting applications. For instance, Tekin et al. [64] use attention mechanisms on convolutional LSTMs for spatio-temporal weather forecasting; while Khan et al. [65] present a dual stream network with an attention mechanism for photovoltaic power forecasting. These works could provide valuable insights and techniques for energy forecasting applications, as weather conditions are important factors that affect energy generation and demand. However, these works also have some drawbacks, such as ignoring the nonlinear and nonstationary characteristics of energy data, or failing to capture both long-term and short-term patterns.

Unlike existing approaches, in this work, we employ collaborative attention units that consist of self-attention and cross-attention mechanisms to model intricate interactions among variables and time steps. Moreover, we utilize a temporal fusion layer for the integration of long-term and short-term information, deploying concatenation and mapping operations for hierarchical feature extraction. TCOAT is end-to-end trainable, eliminating the need for preprocessing or post-processing steps, and can be learned directly from raw wind power data. To the best of our knowledge, TCOAT is the first method that combines temporal collaborative attention with temporal fusion for wind power forecasting.

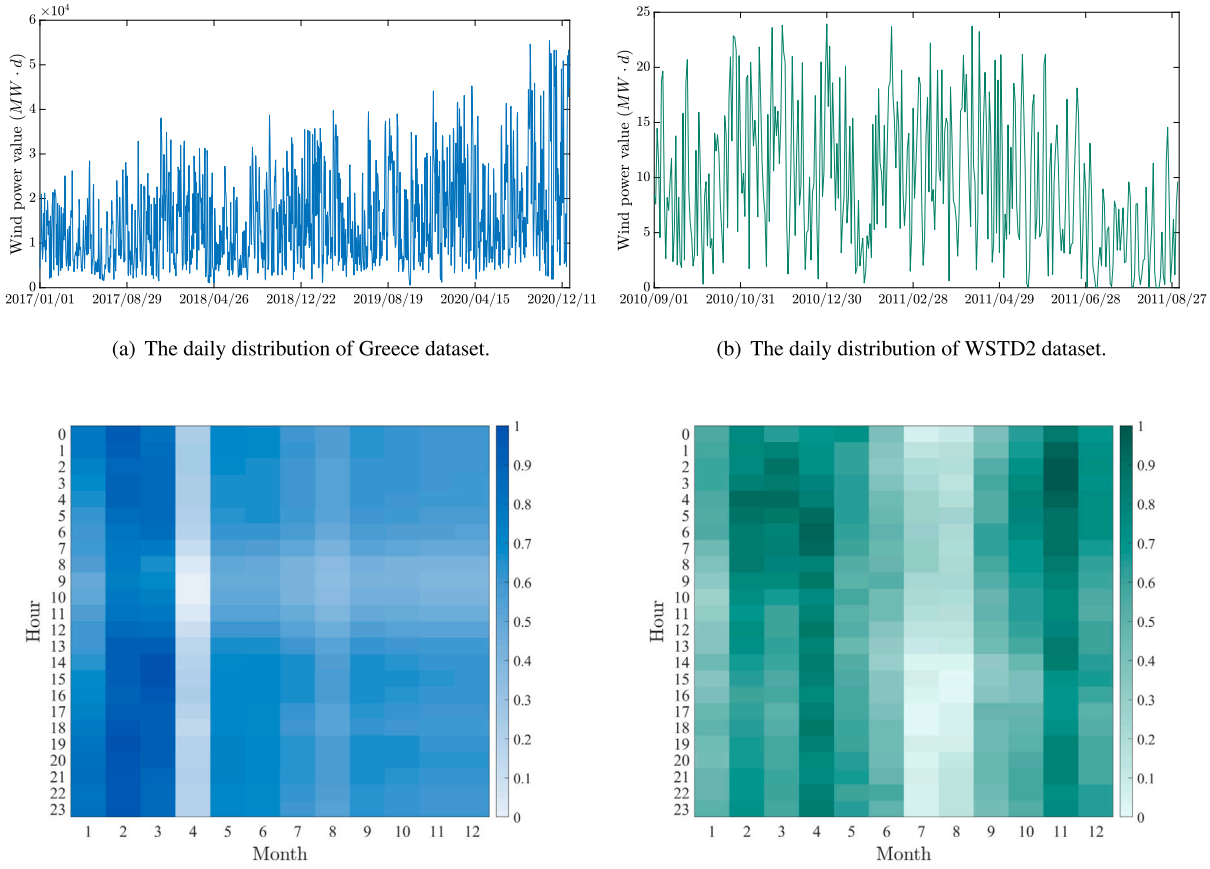
3. Materials

This study uses two datasets of wind power generation data and associated meteorological data with different characteristics and challenges. The first dataset, referred to as the Greece dataset, obtained from the European Network of Transmission System Operators of Electricity (ENTSO-E), contains hourly data from 18 locations in Greece from 2017-01-01 to 2020-12-31. ENTSO-E is an online platform that aggregates energy data from 42 participants in the European centralized energy market. The second dataset, known as the WSTD2 dataset, can be accessed at (https://zenodo.org/records/5516550) [66]. This dataset includes hourly data from 200 randomly chosen turbines situated in a flat terrain inland wind farm, covering the period from 2010-09-01 to 2011-08-31. Wind power generation data exhibit time-varying and nonlinear characteristics, as they are influenced by meteorological factors such as wind speed, wind direction, temperature, humidity, etc., which change over time and affect the efficiency and stability of wind power generation [2]. Moreover, wind power generation data show periodic patterns due to diurnal, seasonal, and climatic variations [49]. These datasets are suitable for evaluating the performance and generalizability of the proposed model.

Fig. 1(a) shows the daily wind power production changes over time for one location in Greece from the Greece dataset. Three main patterns can be observed: (1) The power output varies significantly from month to month, with higher values in May and September, and lower values in April and August. This reflects the long-term changes in wind power generation due to climatic factors such as temperature, precipitation, and pressure; (2) The power output also varies within each day, with lower values around 8 to 12 o'clock, and higher values around 16 to 20 o'clock. This reflects the diurnal changes in wind power generation due to solar radiation and atmospheric stability; (3) The power output does not exhibit obvious seasonal patterns, such as higher values in winter and lower values in summer. This reflects the uncertainty and randomness of wind power generation, as it may be affected by weather, equipment, policy, and other factors that cause anomalies or fluctuations. Fig. 1(c) shows the heatmap of hourly production within a year for the same location. It can be observed that the power output has a clear diurnal pattern, with higher values in the afternoon and lower values in the morning. It can also be observed that the power output has some seasonal patterns, with higher values in spring and autumn, and lower values in summer and winter. However, these patterns are not consistent or regular, as there are some outliers or deviations that indicate the uncertainty and variability of wind power generation.

Fig. 1(b) shows the daily wind power production changes over time from the WSTD2 dataset. Similar patterns can be observed: (1) The power output varies significantly from month to month, with peaks in April and November and troughs in July and August; (2) The power output also fluctuates within each day, with lower values between 14:00 and 18:00 and higher values between 07:00 and 15:00; (3) The power output does not show clear seasonal trends, such as higher values in winter and lower values in summer. These patterns reflect the influence of various factors such as climatic elements, solar radiation, atmospheric stability, and other unpredictable factors like weather conditions and equipment performance. Fig. 1(d) shows the heatmap of hourly production within a year for the same location. It can be observed that the power output has a similar diurnal pattern as the Greece dataset, with higher values in the afternoon and lower values in the morning. It can also be observed that the power output has some seasonal patterns, with higher values in spring and winter and lower values in summer and autumn. However, these patterns are also inconsistent and irregular, as there are some outliers or deviations that reflect the inherent uncertainty and variability of wind power generation.

Two datasets are strategically employed in this paper to assess the proposed TCOAT model, due to the space constraints of the article. The Greece dataset is dedicated to verifying the performance of the model,



(a) The daily distribution of Greece dataset.

(b) The daily distribution of WSTD2 dataset.

(c) The heatmap of hourly production of Greece dataset within a year.

(d) The heatmap of hourly production of WSTD2 dataset within a year.

Fig. 1. The visualized data analyses of Greece dataset and WSTD2 dataset.

validating its accuracy and robustness, and facilitating a comparative analysis with twenty-two state-of-the-art methods across various forecasting tasks and metrics. The WSTD2 dataset, on the other hand, examines the model's generality and its ability to handle diverse data characteristics and scenarios. This approach ensures a comprehensive validation of the model's reliability and wide applicability.

The data preparation phase tackles the prevalent issue of missing values in wind power data with tailored strategies for each dataset. The Greece dataset replaces missing data points with average values calculated from similar dates in the past. This method preserves the integrity, precision, consistency, and temporal patterns of the data, enhancing the robustness of the subsequent forecasting tasks. The WSTD2 dataset, which has only a single year's data and no additional data for imputation, adopts a different approach. It fills missing values with zeros, ensuring the completeness of the dataset and making it suitable for evaluating the adaptability of the proposed model.

4. Methodology

This section delineates the research problem, outlines the data preprocessing steps, and provides a detailed description of the proposed TCOAT model.

4.1. Problem formulation

We consider the problem of predicting the future values of wind power generation using multivariate time series data. Wind power generation is a renewable energy source that depends on both temporal and spatial factors, such as weather conditions, seasonal patterns, and geographical locations. Therefore, forecasting wind power generation is a challenging task that requires capturing the temporal and spatial dependencies of the data, as well as the long-term and short-term patterns. Formally, let $\boldsymbol{X} \in \mathbb{R}^{N\times D}$ be the input data, where N is the number of time steps and D is the number of variables. The input data consists of wind power generation data and associated meteorological data for a given region at a given resolution. Let $\boldsymbol{Y}$ be the output data, where h is the prediction horizon. The output data is the wind power generation for the next period. The goal is to learn a mapping function $f : \boldsymbol{X} \to \boldsymbol{Y}$ that minimizes a loss function $\mathcal{L}(\boldsymbol{Y}, \hat{\boldsymbol{Y}})$, where $\hat{\boldsymbol{Y}} = f(\boldsymbol{X})$ is the predicted data.

The mapping function f can be decomposed into four sub-functions: $f = f_4 \circ f_3 \circ f_2 \circ f_1$, where f_1 is the data preprocessing function, f_2 is the long-term temporal representation function, f_3 is the collaborative attention unit and fusion function, and f_4 is the short-term temporal representation function. Each sub-function can be expressed as follows:

- $f_1 : \boldsymbol{X} \to \boldsymbol{Z}$, where $\boldsymbol{Z} \in \mathbb{R}^{B\times T\times D}$ is the normalized data, and B and T are the batch size and the length of input time steps, respectively.
- $f_2 : \boldsymbol{Z} \to \boldsymbol{L}$, where $\boldsymbol{L} \in \mathbb{R}^{B\times T\times D}$ is the tensor representation of the data.
- $f_3 : \boldsymbol{L} \to \boldsymbol{F}$, where $\boldsymbol{F} \in \mathbb{R}^{B\times 1\times D}$ is the processed data.
- $f_4 : \boldsymbol{S} \to \hat{\boldsymbol{Y}}$, where $\hat{\boldsymbol{Y}} \in \mathbb{R}^{B\times 1\times D}$ is the predicted data.

The loss function $\mathcal{L}(\boldsymbol{Y}, \hat{\boldsymbol{Y}})$ is defined as the mean squared error (MSE) between the true data and the predicted data. The goal is to minimize this loss function by learning the optimal parameters of the mapping function f. The problem can be formulated as an optimization problem as follows:

$$\min_{\theta} \mathcal{L}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}), \tag{1}$$

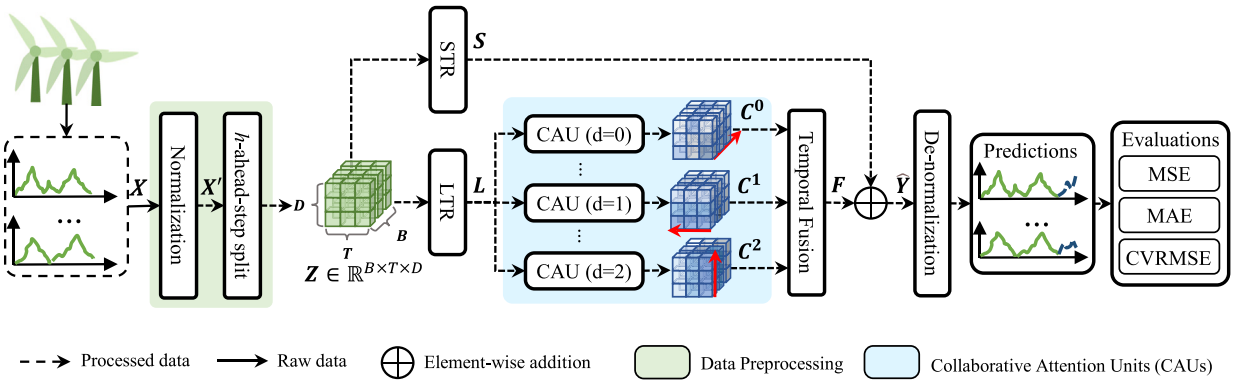


Fig. 2. The schematic illustration of the proposed TCOAT model. The model consists of four components: a long-term temporal representation (LTR) module that uses a GRU network to learn features from historical data, a collaborative attention unit (CAU) module that uses attention mechanisms to capture the directional and global information from the data, a temporal fusion module that uses concatenation and mapping operations to integrate the collaborative information with long-term information, and a short-term temporal representation (STR) module that uses a residual network to learn features from local data.

Algorithm 1 Pseudo-code for training the proposed TCOAT model.

Input: The training set of historical observations $\boldsymbol{X}$, initialize model θ
Output: The trained model θ_{train}
// Feed forward and backward gradient updating
`Trainer(`$\boldsymbol{X}$, θ`)`:
- $\boldsymbol{X}' = \frac{\boldsymbol{X}-\min(\boldsymbol{X})}{\max(\boldsymbol{X})-\min(\boldsymbol{X})}$ ▷ Normalization (Eq. (2))
- $(\mathcal{Z}, \mathcal{Y}) \leftarrow h$-step-forward-split $\boldsymbol{X}'$ ▷ Split (Eq. (4))
- **foreach** batch sample $(\boldsymbol{Z}, \boldsymbol{Y})$ in $(\mathcal{Z}, \mathcal{Y})$ **do**
  - $\boldsymbol{L} \leftarrow$ extract the long-term temporal representation of the input data $\boldsymbol{Z}$ ▷ LTR (Eq. (9))
  - **foreach** direction $d \leftarrow 0$ to 2 **do**
    - $\boldsymbol{C}^d \leftarrow$ calculate the collaborative attention representation using `CAU(`$\boldsymbol{L}, d$`)` ▷ CAU (Algorithm. 2)
  - $\boldsymbol{F} \leftarrow \boldsymbol{G} \leftarrow \boldsymbol{A} \leftarrow [\boldsymbol{Z}; \boldsymbol{C}^0; \boldsymbol{C}^1; \boldsymbol{C}^2]$ ▷ Temporal fusion (Eq. (23); (24); (25))
  - $\boldsymbol{S} \leftarrow$ extract the short-term temporal representation of the input data $\boldsymbol{Z}$ ▷ STR (Eq. (26))
  - $\hat{\boldsymbol{Y}} \leftarrow \boldsymbol{F} \oplus \boldsymbol{S}$ ▷ Prediction (Eq. (27))
  - Loss $\mathcal{L} \leftarrow \boldsymbol{Y}$ and $\hat{\boldsymbol{Y}}$ using MSE ▷ MSE error (Eq. (28))
  - Backward using Adam optimizer [67]
- **return** θ_{train}

where θ denotes the parameters of the mapping function f. This optimization problem can be solved by using Adam, which is a variant of stochastic gradient descent (SGD) with momentum and weight decay. Adam can adaptively adjust the learning rate, making the updated step size suitable for each parameter.

4.2. Overview

Fig. 2 presents a schematic illustration of the proposed Temporal Collaborative Attention (TCOAT). The pseudocode detailing the training process of TCOAT is provided in Algorithm 1. The primary objective of TCOAT is to forecast future values in multivariate time series data, with a specific focus on wind power generation – a domain influenced by both temporal and spatial variables. The entire process comprises four sequential stages: data preprocessing, long-term temporal representation, collaborative attention unit and fusion, and short-term temporal representation. To encapsulate the intricate temporal and spatial dependencies, as well as the long-term and short-term patterns inherent in the data, we introduce TCOAT. The TCOAT model is structured around four core components: a Long-term Temporal Representation (LTR), Collaborative Attention Units (CAUs), a Temporal fusion Layer, and a Short-term Temporal Representation (STR).

The LTR component aims to extract a long-term temporal representation from the input data. The extracted data is then processed by CAUs, which are capable of learning directional information and global information from the data. The CAUs then obtain a collaborative attention representation in multiple directions. Subsequently, the Temporal Fusion Layer integrates these multi-directional collaborative representations, generating fusion data that encapsulates the global characteristics of the original data. Simultaneously, the STR component extracts a short-term temporal representation from the original data. The final prediction is produced by integrating the output of the STR and the Temporal Fusion Layer using a residual network.

The architecture is designed to synergize the strengths of Recurrent Neural Networks (RNNs) and attention mechanisms in the context of multivariate time series forecasting. While RNNs are adept at modeling sequential data dependencies, attention mechanisms excel at discerning the significance and relevance of individual data elements. The integration of these two methodologies enables the generation of predictions that are both accurate and robust. A comprehensive discussion of each stage will be described in the following subsections.

4.3. Data preprocessing

The data preprocessing function f_1 is responsible for normalizing the input data $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ and splitting it into windowed multivariate time series (MTS) with a prediction horizon h. To mitigate the impact of outliers on the learning process of the model and encourage faster convergence, Min-Max normalization is utilized. This technique scales all values of X into a range between 0 and 1. In comparison to Z-Score normalization, Min-Max normalization offers a more straightforward computational process and maintains the original distribution of the data, which can enhance the training process of the model and reduce the influence of outliers. The normalization formula is given by:

$$\boldsymbol{X}' = \frac{\boldsymbol{X} - \min(\boldsymbol{X})}{\max(\boldsymbol{X}) - \min(\boldsymbol{X})}, \tag{2}$$

where $\min(\boldsymbol{X})$ and $\max(\boldsymbol{X})$ are the minimum and maximum values of X, respectively. The de-normalization formula is applied to the outputs of the model in the post-processing stage to recover the original scale of the data. The de-normalization formula is given by:

$$\boldsymbol{X} = \boldsymbol{X}' \cdot (\max(\boldsymbol{X}) - \min(\boldsymbol{X})) + \min(\boldsymbol{X}), \tag{3}$$

The splitting process uses an h-horizon split to transform the data into a supervised learning problem. Given a time series $\boldsymbol{X}' \in \mathbb{R}^{N \times D}$

with N consecutive time intervals and D variables, the h-horizon split is formulated as:

$$\begin{bmatrix} X'_1 & X'_2 & \cdots & X'_T \\ X'_2 & X'_3 & \cdots & X'_{T+1} \\ \vdots & \vdots & \ddots & \vdots \\ X'_{N-T} & X'_{N-T+1} & \cdots & X'_{N-1} \end{bmatrix} \rightarrow \begin{bmatrix} X'_{T+h} \\ X'_{T+1+h} \\ \vdots \\ X'_{N-1+h} \end{bmatrix}, \tag{4}$$

where T is the window size, which determines how many past observations are used as inputs for each prediction. The left part is the normalized inputs of the model, denoted by $\mathcal{Z} \in \mathbb{R}^{(N-T-1)\times T\times D}$, and the right part is the normalized outputs of the model, denoted by $\mathcal{Y} \in \mathbb{R}^{(N-T-1)\times D}$. Several consecutive instances in $(\mathcal{Z}, \mathcal{Y})$ are denoted by $(Z, Y) \in \mathbb{R}^{B\times T\times D} \times \mathbb{R}^{B\times 1\times D}$, where B is the batch size, T is the window size, and D is the number of variables. The splitting process can be generalized to multiple steps ahead by changing the value of h.

4.4. Long-term temporal representation (LTR)

To capture the impact of long-term temporal variables on future wind power generation, we use a gated recurrent unit (GRU) module to extract the hidden representation of the input data. A GRU is a type of recurrent neural network (RNN) that can model the sequential dependencies of the data and handle the vanishing gradient problem [8]. A GRU consists of two gates: a reset gate and an update gate, which control the information flow and the memory state of the network [9]. Given an input tensor $\boldsymbol{Z} \in \mathbb{R}^{B\times T\times D}$, we first apply a linear transformation to each slice of the tensor along the window dimension, denoted by $\boldsymbol{W}_t \in \mathbb{R}^{B\times T\times D}$, where $t = 1, 2, \ldots, T$. Then, we feed each transformed slice to a GRU cell and obtain the hidden state $\boldsymbol{h}_t \in \mathbb{R}^{B\times D}$ at each time step. The GRU cell is defined as follows:

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r[\boldsymbol{h}_{t-1}; \boldsymbol{X}_t; \boldsymbol{W}_t] + b_r), \tag{5}$$

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_z[\boldsymbol{h}_{t-1}; \boldsymbol{X}_t; \boldsymbol{W}_t] + b_z), \tag{6}$$

$$\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W}_s[\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}; X_t; \boldsymbol{W}_t] + b_s), \tag{7}$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t) \odot \tilde{\boldsymbol{h}}_t + \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1}, \tag{8}$$

where $\boldsymbol{r}_t$ and $\boldsymbol{z}_t$ are the reset gate and the update gate, respectively, σ is the sigmoid activation function, $\odot$ is the element-wise product, [;] is concatenation operations, $\boldsymbol{W}_r$, $\boldsymbol{W}_z$, and $\boldsymbol{W}_s$ are the weight matrices, and b_r, b_z, and b_s are the bias terms. To obtain the final long-term temporal representation, we concatenate the hidden states from all time steps and apply another linear transformation, denoted by $\boldsymbol{L} \in \mathbb{R}^{B\times T\times D}$:

$$\boldsymbol{L} = \boldsymbol{W}_l[\boldsymbol{h}_1; \boldsymbol{h}_2; ...; \boldsymbol{h}_T] + b_l, \tag{9}$$

where $\boldsymbol{W}_l$ and b_l are the weight matrix and the bias vector of the linear transformation, respectively. The output of this module is then fed to the next module for collaborative attention and fusion.

We employ the GRU module to extract the impact of long-term temporal variables on future time. Based on the previous $\boldsymbol{h}_{t-1}$, it is possible to derive the hidden representation $\boldsymbol{h}_t$ of GRU:

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}^r \cdot [\boldsymbol{h}_{t-1}; \boldsymbol{A}'; \widetilde{\boldsymbol{Y}}; \boldsymbol{A}^\sharp] + \boldsymbol{b}^r), \tag{10}$$

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}^z \cdot [\boldsymbol{h}_{t-1}; \boldsymbol{A}; \widetilde{\boldsymbol{Y}}; \boldsymbol{A}^\sharp] + \boldsymbol{b}^z), \tag{11}$$

$$\boldsymbol{s}_t = \sigma(\boldsymbol{W}^s \cdot [\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}; \boldsymbol{A}; \widetilde{\boldsymbol{Y}}; \boldsymbol{A}^\sharp] + \boldsymbol{b}^s), \tag{12}$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t) \odot \boldsymbol{s}_t + \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1}, \tag{13}$$

where, respectively, $\boldsymbol{r}_t$, $\boldsymbol{z}_t$, and $\boldsymbol{s}_t$ stand for the reset gate, update gate, and cell state at timestamp t. Meanwhile, $\odot$ represent the Hadamard product, and $\sigma(\cdot)$ represent the sigmoid activation function; the weights are $\boldsymbol{W}^r$, $\boldsymbol{W}^z$ and $\boldsymbol{W}^r$, and the associated biases are $\boldsymbol{b}^r$, $\boldsymbol{b}^z$, and $\boldsymbol{b}^s$. To get the final temporal representation, the GRU layer productions of T groups are concatenated and then given a linear transformation:

$$\boldsymbol{X}_l = \boldsymbol{W}^l \cdot [\boldsymbol{h}_{1,t}; \boldsymbol{h}_{2,t}; \cdots; \boldsymbol{h}_{T,t}] + \boldsymbol{b}^l, \tag{14}$$

where $\boldsymbol{X}_l$ is the final output of GRU module, $\boldsymbol{W}^l$ and $\boldsymbol{b}^l$ are the weighting matrix and biases parameters of a linear transformation, respectively.

Algorithm 2 Pseudo-code of for the learning process of the proposed CAU.

Input: The final long-term temporal representation $\boldsymbol{L}$, direction $\boldsymbol{d}$
Output: The collaborative attention representation $\boldsymbol{C}^d$
`Learning_CAU(`$\boldsymbol{L}, \boldsymbol{d}$`):`

```
// Transform the input L into d-th direction
```
$\boldsymbol{L}_r \leftarrow \mathrm{ReLU}(\boldsymbol{L})$ ▷ Relu (Eq. (15))
$\boldsymbol{V} \leftarrow \boldsymbol{L}_r \cdot \boldsymbol{W}_v$ ▷ Multiply learnable weight (Eq. (16))
$\boldsymbol{H}_{d=0} \leftarrow \frac{\exp(\boldsymbol{V}_{b,t,i})}{\sum_{b=1}^{B}\exp(\boldsymbol{V}_{b,t,i})}, \boldsymbol{H}_{d=1} \leftarrow \frac{\exp(\boldsymbol{V}_{b,t,i})}{\sum_{t=1}^{T}\exp(\boldsymbol{V}_{b,t,i})}, \boldsymbol{H}_{d=2} \leftarrow \frac{\exp(\boldsymbol{V}_{b,t,i})}{\sum_{i=1}^{D}\exp(\boldsymbol{V}_{b,t,i})}$ ▷ Calculate directional score (Eq. (17), (18), (19))
$\boldsymbol{I}^d \leftarrow \boldsymbol{H}^d \cdot \boldsymbol{L}_r$ ▷ Transformed representation (Eq. (20))
```
// Calculate the symmetric attention representation
```
$\boldsymbol{J} \leftarrow \boldsymbol{I}^d \cdot \boldsymbol{W}_j$ ▷ Multiply learnable weight (Eq. (21))
$\boldsymbol{H}_{d=0} \leftarrow \frac{\exp(\boldsymbol{J}_{b,t,i})}{\sum_{b=1}^{B}\exp(\boldsymbol{J}_{b,t,i})}, \boldsymbol{H}_{d=1} \leftarrow \frac{\exp(\boldsymbol{J}_{b,t,i})}{\sum_{t=1}^{T}\exp(\boldsymbol{J}_{b,t,i})}, \boldsymbol{H}_{d=2} \leftarrow \frac{\exp(\boldsymbol{J}_{b,t,i})}{\sum_{i=1}^{D}\exp(\boldsymbol{J}_{b,t,i})}$ ▷ Calculate directional score (Eq. (17), (18), (19))
$\boldsymbol{C}^d \leftarrow \boldsymbol{H}^d \cdot \boldsymbol{W}_c$ ▷ Collaborative attention representation (Eq. (22))
return $\boldsymbol{C}^d$

4.5. Collaborative attention unit (CAU)

The CAU is a key component of the TCOAT model, designed to capture both directional and global information from the input data. It consists of two steps: a Directional Transformation (DT) and a Symmetric Attention (SA). Fig. 3 and Algorithm 2 show the detailed structure and pseudo-code of the CAU, respectively. The TCOAT model integrates multiple CAUs, enabling the representation of Collaborative Attention from various directions. This multi-directional approach enhances the model's ability to capture complex patterns in the data, ensuring a smooth and logical flow of information.

4.5.1. Directional transformation (DT)

In the directional transformation step, we first apply a rectified linear unit (ReLU) function to the input tensor $\boldsymbol{L}$ to ensure the reliability of long-term temporal series feature extraction. Then, we multiply the rectified input tensor with a learnable weight matrix $\boldsymbol{W}_v$ to transform it and learn the temporal patterns during the training process. The process can be expressed as follows:

$$\boldsymbol{L}_r = \mathrm{ReLU}(\boldsymbol{L}), \tag{15}$$

$$\boldsymbol{V} = \boldsymbol{L}_r \cdot \boldsymbol{W}_v, \tag{16}$$

where $\boldsymbol{V} \in \mathbb{R}^{B\times T\times D}$ is the result of multiplication, and $\boldsymbol{W}_v \in \mathbb{R}^{T\times D}$ is the weight matrix. Next, we feed $\boldsymbol{V}$ into a softmax layer to enlarge the difference in several aspects. These aspects are instance dimension, temporal dimension, and variate dimension.

The instance attention is employed to observe and highlight the connections between consecutive instances. It enhances the key look-back windows by highlighting the outbreak values. The attention mechanism on instance dimension can be formulated as follows:

$$\boldsymbol{H}^d = \frac{\exp(\boldsymbol{V}_{b,t,i})}{\sum_{b=1}^{B}\exp(\boldsymbol{V}_{b,t,i})}, \quad d = 0, \tag{17}$$

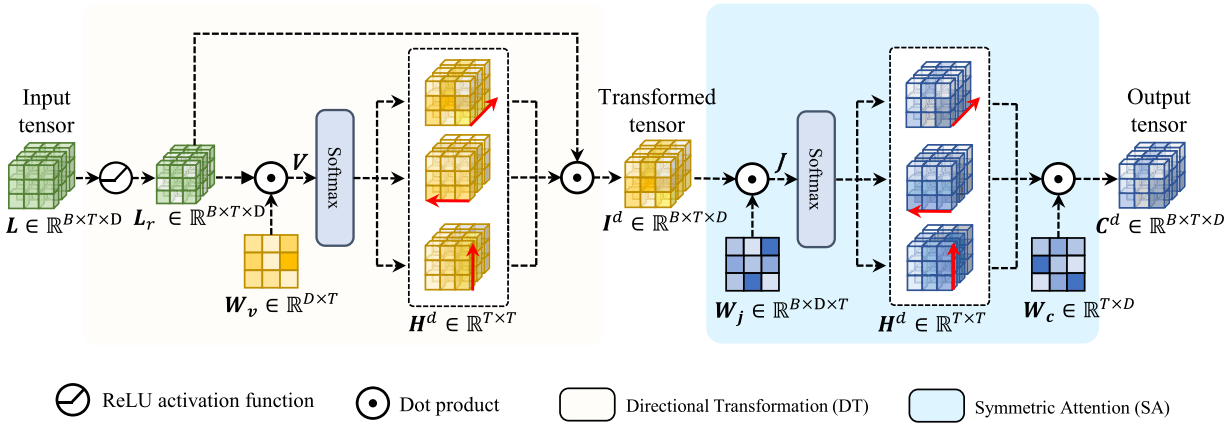


Fig. 3. The schematic illustration of the CAU. The CAU transforms the input data into a tensorial representation and computes attention scores and memory weights for each tensor direction in the DT step. The SA step uses symmetric self-attention and attention mechanisms in three directions to collaboratively enhance the interactions and correlations among different variables or time steps.

The temporal attention is employed to observe and highlight the connections between input and output time steps. It enhances the key time steps by using the feedback of the output time step. The attention mechanism on the temporal dimension can be formulated as follows:

$$\boldsymbol{H}^d = \frac{\exp(\boldsymbol{V}_{b,t,i})}{\sum_{t=1}^{T}\exp(\boldsymbol{V}_{b,t,i})}, \quad d = 1, \tag{18}$$

The variate attention is employed to observe and highlight the connections between the input time series and the output target. It enhances the key factors by using the feedback of the output. The attention mechanism on the variate dimension can be formulated as follows:

$$\boldsymbol{H}^d = \frac{\exp(\boldsymbol{V}_{b,t,i})}{\sum_{i=1}^{D}\exp(\boldsymbol{V}_{b,t,i})}, \quad d = 2. \tag{19}$$

The highlighted score tensor $\boldsymbol{H}^d$ is then multiplied with the processed input tensor $\boldsymbol{L}_r$ to generate the transformed tensor $\boldsymbol{I}^d$ dynamically. The multiplication process can be described as follows:

$$\boldsymbol{I}^d = \boldsymbol{H}^d \cdot \boldsymbol{L}_r, \tag{20}$$

where $\boldsymbol{I}^d \in \mathbb{R}^{B\times T\times D}$ is the transformed tensor, $\boldsymbol{H}^d$ is the highlighted score tensor and $\boldsymbol{L}_r$ is the input tensor processed by a ReLU function.

4.5.2. Symmetric attention (SA)

In the symmetric attention step, we enhance the temporal features of the transformed tensor from different directions. We multiply the transformed tensor $\boldsymbol{I}^d$ with a learnable weight matrix $\boldsymbol{W}_j$ to produce a score tensor $\boldsymbol{J}$. The multiplication process can be described as follows:

$$\boldsymbol{J} = \boldsymbol{I}^d \times \boldsymbol{W}_j, \tag{21}$$

where $\boldsymbol{J} \in \mathbb{R}^{B\times T\times T}$ is the score tensor, $\boldsymbol{I}^d$ is the transformed tensor and $\boldsymbol{W}_j$ is the learnable weight matrix.

Then we apply a softmax function to $\boldsymbol{J}$ to obtain a highlighted score tensor $\boldsymbol{H}^d$, which assigns weights to each element in $\boldsymbol{J}$ according to its importance for future prediction. The softmax function is applied along different dimensions, depending on the direction d. The specific operation processes are the same as formula (17), (18) and (19).

Finally, we multiply the highlighted score tensor $\boldsymbol{H}^d$ with a learnable weight matrix $\boldsymbol{W}_c$ to calculate the final attention tensor $\boldsymbol{C}^d$, which represents the collaborative attention representation from direction d. The multiplication process can be described as follows:

$$\boldsymbol{C}^d = \boldsymbol{H}^d \times \boldsymbol{W}_c, \tag{22}$$

where $\boldsymbol{C}^d \in \mathbb{R}^{B\times T\times D}$ is the attention tensor in dth direction, $\boldsymbol{H}^d$ is the highlighted score tensor and $\boldsymbol{W}_c$ is the learnable weight matrix.

4.6. Temporal fusion

To effectively benefit from those attention representations, a temporal fusion layer was proposed to aggregate the outputs from CAU. The graphical process of the temporal fusion layer is plotted in Fig. 4.

The temporal fusion layer is constituted by a global autoregression (GAR) layer and a linear layer. The outputs from CAUs are first concatenated together. The input tensor is also concatenated to capture short-term temporal dynamics. The process can be formulated as follows:

$$\boldsymbol{A} = [\boldsymbol{Z}; \boldsymbol{C}^0; \boldsymbol{C}^1; \boldsymbol{C}^2], \tag{23}$$

where $\boldsymbol{A} \in \mathbb{R}^{B\times T\times 4*D}$ is the concatenated tensor, $\boldsymbol{Z}$ is normalized time series from data processing, $\boldsymbol{C}^0$, $\boldsymbol{C}^1$ and $\boldsymbol{C}^2$ are attention tensors viewed from dimensions 0, 1, and 2, respectively, [;] is concatenation operations.

The concatenated tensor is passed through a GAR layer to learn the various temporal patterns. The process can be described as follows:

$$\boldsymbol{G} = \sum_{i=t}^{T} \boldsymbol{W}_g \times \boldsymbol{A}_{:,i,:} + b_g, \tag{24}$$

where $\boldsymbol{G} \in \mathbb{R}^{B\times 1\times 4*D}$ is the learned temporal patterns, $\boldsymbol{W}_g \in \mathbb{R}^{T\times 4*D}$ is the weight matrix, b_g is a bias.

The learned pattern tensor is passed through a linear fusion layer to generate consecutive model outputs. The process can be described as follows:

$$\boldsymbol{F} = \sum_{i=t}^{T} \boldsymbol{W}_l \times \boldsymbol{G}_{:,:,i} + b_l, \tag{25}$$

where $\boldsymbol{F} \in \mathbb{R}^{B\times 1\times D}$ is the model outputs, $\boldsymbol{W}_l \in \mathbb{R}^{4*D\times D}$ is the weight matrix, b_l is a bias.

4.7. Short-term temporal representation (STR)

In general, future wind power generation will be more influenced by short-term temporal data than by long-term temporal data. Simple models, such as linear models or RNN-based methods, can be used to capture short-term time series features. Fig. 4's lower part presents the processing. Different short-term time series feature capture models can be applied to various horizon split data sets to get various outcomes:

$$\boldsymbol{S} = \mathcal{R}(\boldsymbol{Z}_{R:}), \tag{26}$$

where $\boldsymbol{Z}_{R:} \in \mathbb{R}^{B\times R\times D}$ is the input data of the short-term temporal part, and it represents the last R time steps of input temporal data, $\boldsymbol{S} \in \mathbb{R}^{B\times 1\times D}$ is the short-term temporal process result, and $\mathcal{R}$ is the short-term temporal process function, linear and GRU and other models are

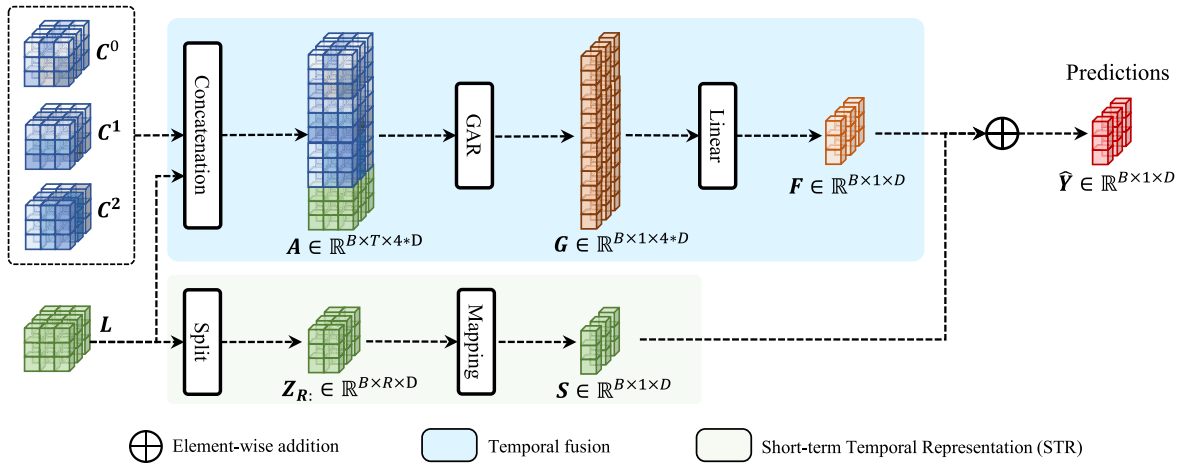


Fig. 4. The schematic illustration of the conditional fusion procedures. The layer concatenates the input data and the outputs from the CAUs, and then applies a global autoregression (GAR) layer and a linear layer to fuse them. The layer also splits the input data into different parts and feeds them into a short-term temporal representation (STR) module that uses a residual network to capture the short-term variations of the data. The final predictions are obtained by combining the outputs from the temporal fusion layer and the STR module.

optional. Then, in order to obtain the final forecast data, we combine the short-term data features with the long-term data features:

$$\hat{\boldsymbol{Y}} = \boldsymbol{F} \oplus \boldsymbol{S}, \tag{27}$$

where $\hat{\boldsymbol{Y}} \in \mathbb{R}^{B\times 1\times D}$ is the final output of TCOAT, and $\boldsymbol{F}$ represents the final output of temporal fusion, $\boldsymbol{S}$ represents the final output of short-term temporal process, and $\oplus$ is the symbol of the residual operation.

5. Experiments

In this section, we describe the data, experimental settings, model implementations, and results. We also compare the proposed model with other methods.

5.1. Data and experimental settings

In the experiments, the Greece dataset served as the primary source of data. To further validate the generalizability of the TCOAT model, the WSTD2 dataset was also incorporated. This auxiliary dataset offers a unique set of data characteristics for analysis. Detailed introductions to both datasets can be found in Section 3.

The preprocessing of the primary dataset was meticulously executed as follows: (1) Aggregation: Hourly data points were aggregated into daily averages to reduce data noise and complexity; (2) Standardization: The daily values were standardized using their annual mean values to account for changes in wind turbine installations over time; (3) Min-Max Scaling: We applied min–max scaling to the standardized values, normalizing them into a range between 0 and 1, which is optimal for neural network training; The primary dataset was divided into training and testing sets in a 4:1 ratio to ensure ample data for training while preserving the temporal sequence. For the additional dataset, we followed a similar preprocessing methodology and divided it into training and testing sets using an 80:20 ratio, considering its unique characteristics and the need to validate the model across varied conditions.

The TCOAT model was implemented using PyTorch v2.0.0 [68]. The computational resources included a server with an Intel® Xeon® Gold 5218R CPU (2.10 GHz), 256 GB memory, and four Tesla V100-PCIE-16 GB GPUs. The prediction horizon was set to one day for the primary dataset, a standard in wind power forecasting [69], with input intervals of 1, 3, 5, and 7 days. These intervals were chosen to evaluate the model's performance over different time scales, reflecting the trade-off between accuracy and complexity.

5.1.1. Comparison baselines

In this section, we present a comprehensive comparison of the proposed TCOAT model with twenty-two state-of-the-art methods for multivariate time series forecasting. We summarize the main features and characteristics of these methods in Table 1, such as the model type, the components, the advantages, and the disadvantages. The model type indicates whether the method is based on statistical, machine learning, or deep learning techniques. The components describe the main modules or layers of the method that are used to capture the temporal and collaborative patterns in the data. The advantages highlight the strengths or benefits of the method for multivariate time series forecasting. The disadvantages point out the limitations or drawbacks of the method that may affect its performance or applicability.

5.1.2. Model configurations

We conducted five repeated experiments on the wind power time series data to evaluate the performance of each method. We used this approach instead of cross-validation to preserve the temporal order of the data, which is essential for forecasting tasks. We trained the models using the Adam optimizer [67] with the mean squared error (MSE) as the loss function, following previous studies that showed their effectiveness for wind power forecasting [89]. We applied the grid search method to optimize the hyper-parameters for each method over a predefined range of values. The optimal hyper-parameters for each baseline method obtained by the grid search method are shown in Table 6.

5.1.3. Evaluation metrics

To evaluate the performance of the proposed TCOAT model and compare it with other methods, we use three evaluation metrics that are commonly used in the field of energy prediction. These metrics are Mean Square Error (MSE), Mean Absolute Error (MAE), and Coefficient of Variation of Root Mean Square Error (CVRMSE). These metrics can measure the accuracy and reliability of the prediction models, as well as reflect the characteristics and challenges of wind power data.

MSE is a scale-dependent metric that measures the average squared difference between the predicted and actual values. MSE is sensitive to outliers and large errors, which means that it penalizes large deviations more than small ones. MSE is defined as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{28}$$

where n is the number of samples, y_i is the actual value, and $\hat{y}_i$ is the predicted value.

Table 1
Benchmark methods for wind power forecasting.

Method	Type	Components	Advantages	Disadvantages
GAR [70]	Linear	Autoregressive	Simple and fast	Cannot capture nonlinear dependencies
AR [71]	Linear	Autoregressive	Simple and fast	Cannot capture cross-series dependencies
VAR	Linear	Autoregressive	Can capture cross-series dependencies	Cannot capture nonlinear dependencies
DLinear [72]	Linear	Linear layers with decomposition	Can model trend and seasonality separately	Cannot capture nonlinear dependencies or interactions between components
NLinear [72]	Linear	Linear layer with normalization	Can reduce the scale of input data	Cannot capture nonlinear dependencies or temporal patterns
FiLM [73]	Linear + neural network	Linear and Legendre Polynomials projections with frequency enhancement	Can model trend, seasonality, and frequency components of time series data	Cannot capture nonlinear dependencies or interactions between components
LSTM [74]	Neural network	LSTM cells with gate units	Can learn long-term temporal dependencies	Prone to overfitting and vanishing gradients
GRU [75]	Neural network	GRU cells with simpler gate structure	Similar to LSTM but faster and easier to train	May lose some information due to fewer gates
ED [76]	Neural network	LSTM encoder–decoder	Can process variable-length input and output sequences	May suffer from information bottleneck due to fixed-length hidden state
CNN1D	Neural network	Convolutional layers 1D	Can learn local feature representations of time series	Cannot capture long-term temporal dependencies or cross-series dependencies
CRNN [77]	Neural network	Convolutional layers + RNN layers	Can combine local features and historical information	May have high computational cost due to multiple components
CRNNRes [78]	Neural network	Convolutional layers + RNN layers + residual connections	Similar to CRNN but can recover specific information that convolution reduces	May have high computational cost due to multiple components and residual connections
LSTNet [79]	Neural network	Redesigned convolutional and recurrent structures	Can model long-term dependencies and periodic patterns in time series data	May have high computational cost due to multiple components and attention mechanism
Transformer [80]	Neural network	Encoder–decoder with attention mechanisms	Can process variable-length input and output sequences without recurrent neural networks	May suffer from information loss due to positional encoding and fixed-length hidden state
Informer [81]	Neural network	Improved Transformer with ProbSparse attention, distilling, and generative decoder	Can handle long sequence time-series forecasting with high efficiency and low memory consumption	May not be able to capture complex nonlinear dependencies well due to linear projection layers
Autoformer [82]	Neural network	Improved Transformer with Auto-Correlation attention and series decomposition	Can model long-term periodic patterns and dependencies in time series data	May have high computational cost due to multiple components and attention mechanisms
FEDformer [83]	Neural network	Improved Transformer with frequency-based low-rank attention and mixture of experts decomposition	Can decompose time series into different frequency components and learn their interactions	May have high computational cost due to multiple components and attention mechanisms
DSANet [84]	Neural network	Convolutional layers + self-attention module	Can capture global and local temporal patterns and dependencies in multivariate time series	May not be able to handle long-term dependencies well due to fixed-length input and output
TPA-LSTM [85]	Neural network	LSTM cells with temporal pattern attention	Can capture nonlinear interdependencies among time steps and series	May have high computational cost due to attention mechanism
StemGNN [86]	Graph neural network	Graph and Fourier transforms	Can capture inter-series and temporal patterns in multivariate time series data	May not be able to handle long-term dependencies well due to fixed-length input and output
GAIN [87]	Graph neural network	Graph neural networks and collaborative attention	Can predict time series based on multivariate time series data with spatial correlations	May have high computational cost due to graph convolution and attention mechanism
MSL [88]	Shapelet learning	Multiple shapelets learned from historical observations	Can identify crucial subsequences from time series data	Cannot capture consecutive temporal dependencies or cross-series dependencies

MAE is another scale-dependent metric that measures the average absolute difference between the predicted and actual values. MAE is less sensitive to outliers and large errors than MSE, which means that it treats all errors equally. MAE is defined as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{29}$$

CVRMSE is a scale-independent metric that measures the normalized root mean square error relative to the mean of the actual values. CVRMSE can compare the performance of different models or datasets with different scales or units. CVRMSE is defined as follows:

$$\mathrm{CVRMSE} = \frac{\sqrt{\mathrm{MSE}}}{\bar{y}} \times 100\%, \tag{30}$$

where $\bar{y}$ is the mean of the actual values.

The lower the values of these metrics, the better the performance of the prediction model. However, these metrics also have some limitations. For example, MSE and MAE do not consider the temporal correlation or order of the time series data, which may affect the prediction accuracy. CVRMSE may not reflect the absolute error or deviation of the prediction model, which may affect the reliability of the

Table 2
The prediction results of attention combinations on the wind datasets in terms of MSE, MAE and CVRMSE, whereCAUs{i_1}, CAUs{i_1, i_2}, or CAUs{i_1, i_2, i_3} means directional attentions on i_1, i_2, i_3-th aspect. The best results are shown in bold, the second-best results are underlined, and the worst results are in wavy lines; Unit of h: day.

Structure	$h = 1$			$h = 3$			$h = 5$			$h = 7$		
	MSE	MAE	CVRMSE	MSE	MAE	CVRMSE	MSE	MAE	CVRMSE	MSE	MAE	CVRMSE
CAUs{0}	89.141020	7.455473	0.466461	190.789688	10.710353	0.682436	178.393216	10.662587	0.659879	185.291901	10.853207	0.672530
CAUs{1}	99.727310	7.995392	0.493391	192.054016	10.774353	0.684694	185.216174	10.888910	0.672392	186.275101	10.987542	0.674314
CAUs{2}	99.623505	7.989298	0.493135	192.668442	10.766847	0.685788	184.245422	10.866638	0.670629	186.175034	10.945656	0.674132
CAUs{0, 0}	88.513838	**7.366273**	0.464810	175.301437	10.562387	0.654150	173.882065	10.600887	0.651489	180.833363	**10.815968**	0.664372
CAUs{0, 1}	**88.209052**	7.390437	**0.464018**	**165.119263**	**10.396696**	**0.634868**	**173.061798**	**10.564649**	**0.649958**	**179.194824**	10.889776	**0.661714**
CAUs{0, 2}	89.789558	7.469786	0.468158	175.727554	10.590677	0.654945	174.546042	10.567832	0.652721	184.640320	10.854650	0.671346
CAUs{1, 1}	99.849643	7.988314	0.493694	192.185333	10.786354	0.684928	185.056775	10.878568	0.672104	187.006236	10.943714	0.675636
CAUs{1, 2}	99.733067	7.981781	0.493406	192.554153	10.781023	0.685585	183.836032	10.861772	0.669882	185.007304	10.912992	0.671978
CAUs{2, 2}	99.566441	7.972888	0.492993	192.839142	10.765305	0.686092	183.453433	10.861510	0.669185	185.967438	10.950788	0.673757
CAUs{0, 0, 0}	89.989612	7.463878	0.468668	182.044006	10.739547	0.666612	177.198434	10.657067	0.657642	184.284312	10.859870	0.670701
CAUs{0, 0, 1}	88.768106	7.375317	0.465470	173.187408	10.465259	0.650194	174.519796	10.610096	0.652668	183.695099	10.862698	0.669627
CAUs{0, 0, 2}	88.930673	7.409399	0.465904	170.772614	10.469582	0.645645	177.232968	10.670097	0.657737	184.326543	10.864437	0.670778
CAUs{0, 1, 1}	92.489603	7.592930	0.475118	176.551422	10.598987	0.656478	176.950177	10.611993	0.657190	184.288396	10.828673	0.670708
CAUs{0, 1, 2}	91.179321	7.529140	0.471715	176.916565	10.533190	0.657157	176.860547	10.619254	0.657016	183.853648	10.846724	0.669912
CAUs{0, 2, 2}	92.155937	7.590806	0.474237	175.264099	10.551161	0.654081	176.534933	10.616940	0.656432	184.838598	10.854416	0.671709
CAUs{1, 1, 1}	99.470265	7.963058	0.492755	192.478394	10.781264	0.685450	184.808762	10.869243	0.671653	186.331080	10.939775	0.674415
CAUs{1, 1, 2}	99.296432	7.966491	0.492324	192.737885	10.766327	0.685912	185.194586	10.867844	0.672354	186.137990	10.967845	0.674066
CAUs{1, 2, 2}	99.411821	7.968369	0.492610	192.964767	10.784638	0.686315	184.773367	10.873874	0.671588	186.342183	10.945947	0.674435
CAUs{2, 2, 2}	99.480754	7.962707	0.492781	193.152878	10.772007	0.686650	185.290240	10.873751	0.672528	186.541326	10.963457	0.674792

model. Therefore, using multiple evaluation metrics can provide a more comprehensive and objective assessment of the prediction models.

5.2. Parameter sensitivity analyses

TCOAT consists of the following four key components: a LTR, CAUs, a Temporal Fusion Layer, and a STR. The LTR module uses a GRU as its RNN unit, with a hidden size of 64 and a layer count of 1, and the STR module has a residual window size of 3. In this study, these components have fixed hyperparameters, while the CAUs have variable hyperparameters that affect the model performance. CAUs are attention mechanisms that can extract directional information from the input data and generate directional attention representations. The direction can be batch (B), window size (T), or multivariate time series (D). To investigate how the choice of different CAUs affects wind power forecasting performance, we conducted several experiments with different numbers and combinations of CAUs, using a combination quantity of less than 4. We concatenated the outputs of the CAUs to form the final attention representation. We denoted the attention combinations as CAUs{i_1}, CAUs{i_1, i_2}, or CAUs{i_1, i_2, i_3}, where i_1, i_2, i_3 indicate the dimensions that the CAUs focus on. We used three metrics: mean square error (MSE), mean absolute error (MAE), and coefficient of variation of root mean square error (CVRMSE). We considered four prediction horizons: 1 day ahead ($h = 1$), 3 days ahead ($h = 3$), 5 days ahead ($h = 5$), and 7 days ahead ($h = 7$). The results, shown in Table 2, indicate that using multiple CAUs can improve the prediction performance compared to using a single CAU, as multiple CAUs can capture more information and diversity from the input data. Moreover, the results show that selecting an appropriate attention direction can also improve the prediction performance. For example, CAUs{0,1} has the best results for $h = 3$, $h = 5$, and $h = 7$, which means that using attention on the batch and window size directions is more effective than using attention on other directions. The results also show that the prediction performance is relatively stable across different prediction horizons, which indicates that the proposed model can handle long-term forecasting tasks well. The CAUs combinations can be applied to various time series forecasting tasks that involve multidimensional input data.

To determine the optimal combination of CAUs for the proposed model, we performed experiments on the wind power dataset with different combinations of CAUs and different prediction horizons (h = 1, 3, 5, and 7). Other parameters were kept constant for all experiments. We combined CAUs from different dimensions, ranging from one to three CAUs per combination. The final attention representation was obtained by concatenating the outputs of all CAUs in a combination. Table 2 shows the results of the experiments. The combinations are denoted as CAUs{i_1}, CAUs{i_1, i_2}, or CAUs{i_1, i_2, i_3}, where i_1, i_2, i_3 indicate the dimensions that the CAUs focus on. From the table, we can see that CAUs{0} perform better than CAUs{1} and CAUs{2} when used alone. This means that 0-direction attention can capture more effective information than 1-direction or 2-direction attention, and the attention representations are different for each direction. However, when combined with other CAUs, CAUs{0} can enhance the performance of other directions. This indicates that combining different directions can improve the quality of the attention representation. For short-term prediction (h=1), multiple CAUs that include 0-direction (such as CAUs{0,1}, CAUs{0,2}, CAUs{0,1,1}, CAUs{0,1,2}, etc.) have similar performance to CAUs{0}. For mid-term prediction (h=3,5 and 7), multiple CAUs that include 0-direction have better performance than CAUs{0}. This suggests that incorporating more directions can help the model capture the underlying energy generation patterns better. As CAUs{0,1} can achieve the best performance for all prediction horizons, we choose 0 and 1 as the optimal directional combination of CAUs for the proposed model.

5.3. Comparison with baselines on multi-horizon prediction

In this subsection, we compare TCOAT with other methods on different time horizons for wind power forecasting. We use three metrics: MSE, MAE, and CVRMSE. We consider four time horizons: 1 day ahead ($h = 1$), 3 days ahead ($h = 3$), 5 days ahead ($h = 5$), and 7 days ahead ($h = 7$). The results in Table 3 shows that:

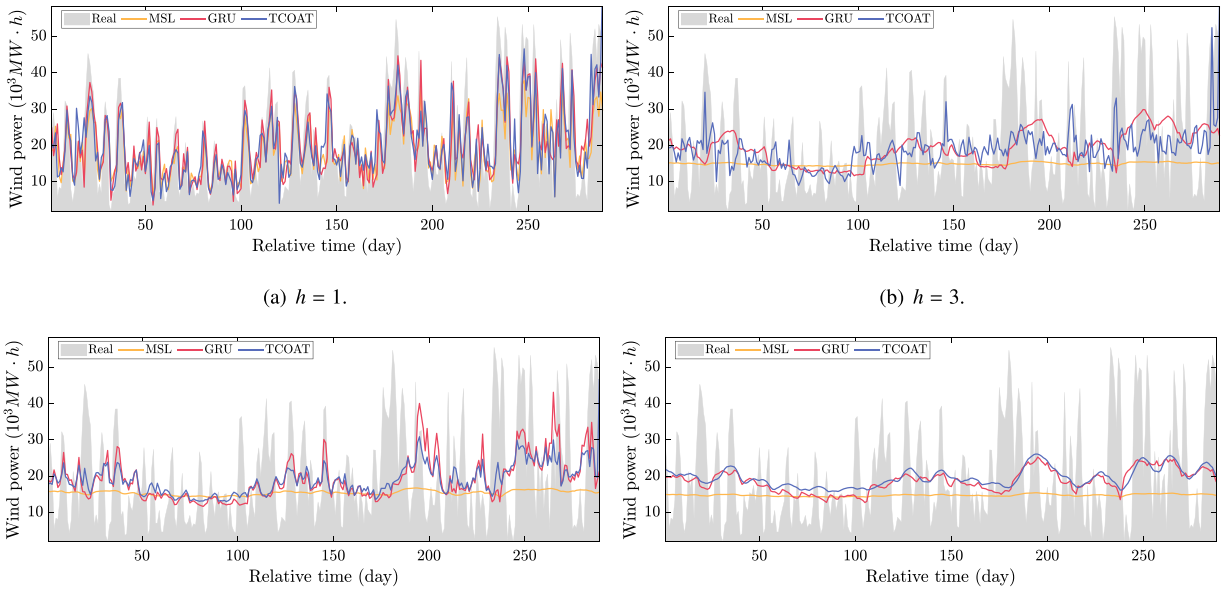
- Linear models (GAR, AR, and VAR) have similar performance for h=1, indicating that they can capture short-term trends in wind power. However, their performance deteriorates as h increases, showing that they have difficulty in forecasting the future over multiple horizons. VAR has the worst performance among all models at h=7. GAR is the most stable among them, but it still performs poorly for h=3, h=5, and h=7, which means that the linear model cannot capture the complex patterns of wind power generation.
- Linear model variations (DLinear, NLinear, Film) that use data decomposition, normalization, or frequency augmentation do not outperform linear models significantly. They have low performance across all horizons, especially Film for h=1. These results

Table 3
Performance comparison in wind power prediction. The best results are shown in bold, the second-best results are underlined, and the worst results are in wavy lines; Unit of h: day.

Model	h = 1			h = 3			h = 5			h = 7		
	MSE	MAE	CVRMSE	MSE	MAE	CVRMSE	MSE	MAE	CVRMSE	MSE	MAE	CVRMSE
GAR	107.059937	8.215399	0.511209	195.129089	10.990736	0.690154	194.998367	11.071337	0.689922	196.631302	10.923676	0.693161
AR	107.059929	8.215398	0.511209	204.643738	11.020720	0.706780	194.907990	11.067622	0.689762	224.185181	11.160784	0.740135
VAR	105.499558	8.212302	0.507470	202.201935	11.232389	0.702550	212.181076	11.247817	0.719678	201.425186	11.308241	0.701560
DLinear	107.187714	8.220581	0.511514	197.490189	10.928032	0.694317	194.596939	10.983214	0.689212	210.039108	10.954163	0.716404
NLinear	116.757805	8.325392	0.533860	217.222214	11.563727	0.728177	201.441467	11.394403	0.701228	198.787476	11.360843	0.696951
FiLM	187.586472	10.981367	0.676683	203.167282	11.365602	0.704225	202.799408	11.402464	0.703587	196.467117	11.200686	0.692872
LSTM	96.815292	7.866727	0.483804	188.990631	10.925255	0.679211	182.677216	10.412541	0.667770	193.943436	10.845874	0.688407
GRU	93.298294	7.603762	0.477223	181.482742	10.877367	0.665583	183.816483	10.884043	0.669849	183.036743	10.920844	0.668770
ED	102.191811	8.224937	0.499451	196.289063	11.245865	0.692202	180.293564	**10.302210**	0.663399	188.735062	11.100488	0.679101
CNN1D	107.348450	8.322651	0.511897	197.096039	11.057461	0.693623	195.252823	11.025487	0.690372	195.186615	10.887697	0.690610
CRNN	102.104042	8.214901	0.499236	188.390350	11.224201	0.678132	201.627731	11.528132	0.701552	193.406128	10.950719	0.687453
CRNNRes	97.259697	7.990826	0.487249	184.929138	11.125570	0.671873	207.420944	11.604235	0.711559	193.970245	10.977121	0.688455
LSTNet	98.855507	8.002940	0.491230	194.352371	11.185124	0.688779	198.950027	11.189010	0.696878	193.321854	10.918596	0.687303
Transformer	115.212799	9.050337	0.530316	195.356232	11.351683	0.690555	198.036163	11.829300	0.695276	195.490723	11.473658	0.691148
Informer	107.352913	8.652519	0.511908	188.467255	11.225601	0.678270	193.255280	11.726625	0.686832	190.404556	11.395064	0.682097
Autoformer	107.609352	8.323877	0.512519	204.596619	11.392544	0.706698	209.537201	11.800620	0.715180	199.185165	11.436871	0.697648
FEDformer	112.737099	8.535225	0.524588	204.154694	11.474975	0.705935	209.762802	11.534991	0.715565	197.095367	11.273081	0.693978
DSANet	112.407669	8.503561	0.523821	187.524765	10.814522	0.676572	213.651184	10.928378	0.722167	187.359970	**10.795478**	0.676622
TPA-LSTM	111.674713	8.578475	0.522110	180.052689	10.761929	0.662956	197.486984	10.864628	0.694311	179.853745	10.951759	0.662930
StemGNN	213.951080	11.006669	0.722673	219.605392	11.081501	0.732160	217.613831	11.054159	0.728833	217.612076	11.054134	0.728830
GAIN	100.697830	8.158662	0.495787	183.303040	10.738844	0.668913	198.245804	11.172346	0.695644	190.160355	10.933922	0.681660
MSL	111.519417	8.463414	0.521747	216.158707	11.015688	0.726392	214.340485	10.992064	0.723331	217.488617	11.032725	0.728998
TCOAT	88.209052	**7.390437**	**0.464018**	**165.119263**	**10.396696**	**0.634868**	**173.061798**	10.564649	**0.649958**	**179.194824**	10.889776	**0.661714**

indicate that these linear models are not suitable for wind power forecasting because they cannot handle the short-term variations and long-term dependencies of wind power.

- Recurrent neural networks (RNNs) such as LSTM, GRU, and ED generally achieve second-best or third-best results across various horizons. Notably, the GRU model consistently showed the best performance among these RNNs, effectively capturing the temporal dependencies in wind power data. This is indicative of the strength of its gating mechanism and its ability to leverage historical information for current data predictions. However, when compared to our proposed TCOAT model, both LSTM and GRU models, despite their merits, exhibit limitations. The LSTM, known for its ability to handle long-term dependencies, falls short in terms of predictive accuracy and flexibility when dealing with the complex, non-linear patterns characteristic of wind power forecasting. This is evident from the performance metrics in Table 3, where TCOAT consistently outperforms LSTM, especially in terms of MSE, MAE, and CVRMSE across all forecast horizons. The LSTM model's limitations in inference power and reliance on substantial training data quality and quantity become apparent when juxtaposed with the advanced capabilities of TCOAT. Our model incorporates the novel integration of dynamic attention mechanisms, collaborative attention units for assimilating multi-dimensional data, and a temporal fusion layer for effective long-term and short-term pattern analysis. These contribute to its performance and address the gaps observed in traditional LSTM models. This comparative analysis underscores the novelty and effectiveness of the TCOAT model in wind power forecasting, offering a more accurate, reliable, and nuanced approach than existing LSTM-based methods.
- CNN- and RNN-based models (CNN1D, CRNN, CRNNRes, and LSTNet) perform better than linear and linear variants models, but worse than RNN-based models. CNN1D has a similar performance to GAR, which means it does not extract useful features from input sequences effectively. CRNN models use both local features and historical information to enhance prediction accuracy, unlike pure convolutional neural networks that only rely on local features. For short-term horizon prediction tasks (h=1), CRNNRes models are more stable than CRNN models, indicating that the residual connection helps to capture the low horizon features. However, CRNNRes performs worse than CRNN for h=5 and h=7 prediction tasks. This may be because the residual window is not large enough to support long-term forecasting with sufficient residual information. LSTNet adds a skip window to CRNNRes, which splits the input sequence into small segments and models them using GRU. The performance of LSTNet is similar to CRNNRes, which means that skipping windows does not help to learn useful representations. Overall, the hybrid RNN models (i.e., CRNN and CRNNRes) are not as effective as the RNN model alone, which suggests that convolutional models are not sufficiently accurate to represent temporal dependencies by capturing regional features.
- Self-attention-based models (Transformer, Informer, Autoformer, and Fedformer) have slightly better results than linear models but worse than RNN models. Among them, Informer has the best performance but only slightly better than Transformer. Autoformer and Fedformer are only better than Transformer when h=1 but slightly worse than Transformer when h=3, h=5 or h=7. This suggests that their attention mechanism and series decomposition are not effective in finding periodic patterns and dependencies in wind power data.
- CNN-RNN and attention models as the two core components of the hybrid attention model (DSANet and TPA-LSTM) perform better than CNN-RNN models. TPA-LSTM achieves the second-best performance among all methods at h=3 and h=7, demonstrating that its temporal pattern attention can capture long-term dependencies by two core components.
- Graph attention-based models (StemGNN and GAIN) use a graph attention mechanism to model the spatial correlation of wind power data. StemGNN has the worst performance among all methods at all horizons, indicating that its graph attention mechanism cannot learn meaningful representations. GAIN improves over StemGNN by using a collaborative attention mechanism, which can enhance the spatial–temporal features. GAIN performs better than CNN- or RNN-based methods, but still worse than RNN-based methods.
- MSL learns shapelets from historical data to represent wind power patterns. Its performance is low at all horizons, suggesting that shapelets are not effective features for wind power forecasting.



(a) $h = 1$. (b) $h = 3$.

(c) $h = 5$. (d) $h = 7$.

Fig. 5. The visualized comparisons on the real values with the other three methods. Unit of h: day.

In the wind power prediction experiment conducted in the Greece dataset, four time horizons were considered: 1 day ahead, 3 days ahead, 5 days ahead, and 7 days ahead. From a single time horizon perspective, the GRU model emerged as the best-performing baseline method. However, TCOAT outperformed all methods to achieve the best results across all metrics and time horizons, particularly in lower time horizons. To provide a comprehensive performance measure, the average of the four time horizons was calculated for each method's MSE, MAE, and CVRMSE. In this context, TCOAT demonstrated significant improvements. Specifically, when compared with the GRU model - the top-performing baseline method among the twenty-two state-of-the-art methods evaluated - TCOAT achieved a maximum reduction of 5.62%, 2.59%, and 2.85% in MSE, MAE, and CVRMSE, respectively. This highlights the effectiveness of TCOAT in enhancing the accuracy of wind power prediction. This demonstrates that its novel components (LTR, CAUs, STR, and temporal fusion) can effectively capture the temporal dependencies and collaborative patterns in wind power data across different time scales. Moreover, TCOAT is more robust to the increase of horizon than other methods.

Fig. 5 shows the comparison of the actual and predicted wind power by TCOAT, GRU, and MSL. TCOAT performs the best in tracking the wind power variations, especially when the prediction horizon h is medium (3), as it can identify part of the peak-trough trend in Fig. 5(b), while the two other benchmarks fail to respond. However, as h increases, all the methods tend to underestimate the peak values and lose accuracy. GRU can capture the time-dependent relationship at h=1 by using the RNN component to learn features from historical information, but it fails to predict the peaks and valleys of the time series accurately at h=3, h=5, and h=7. TCOAT leverages a long-term pattern framework to capture long-term representations, a directional collaborative attention mechanism to focus on relevant features, and a short-term pattern framework to capture short-term representations, which enables it to produce more accurate and realistic predictions.

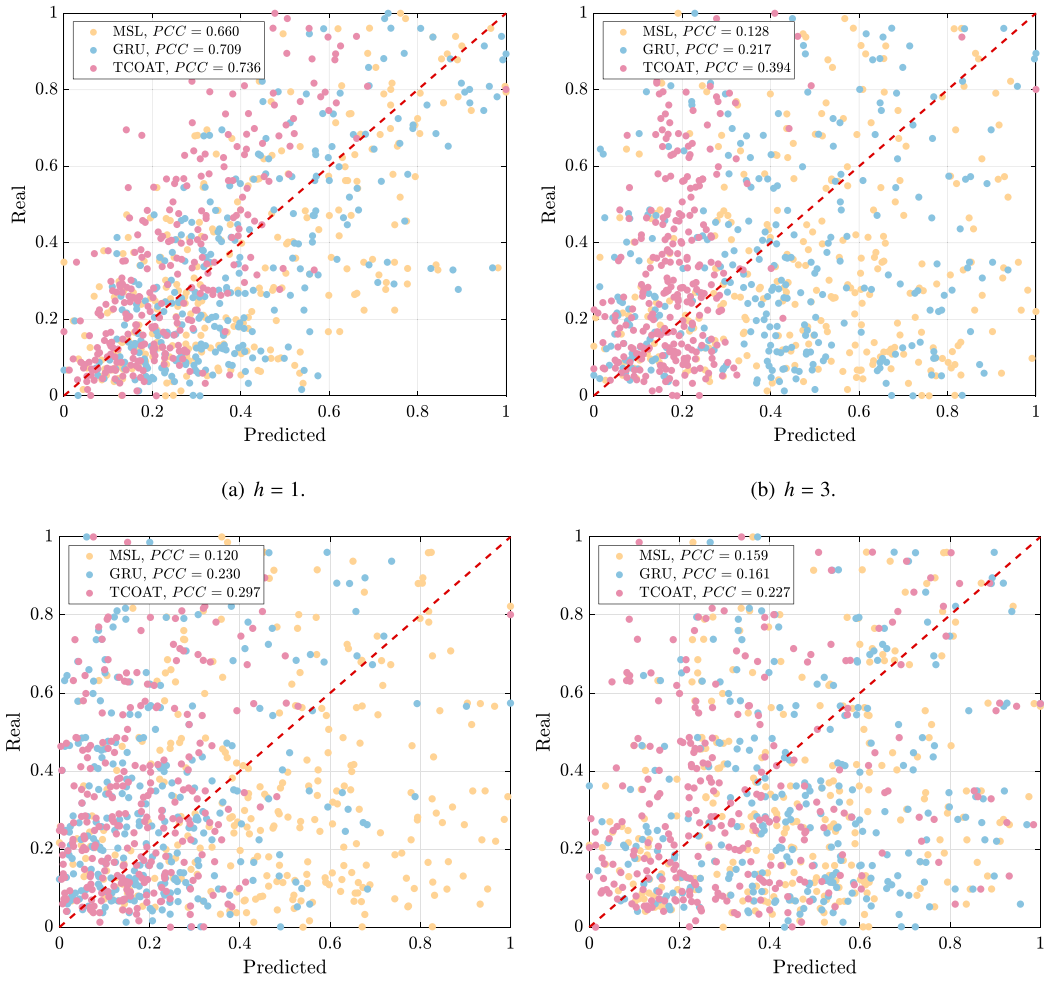
Fig. 6 displays the normalized results from actual and predicted values. The Pearson correlation (PCC) between the actual value and the predicted value of different models is also annotated in the figure. The prediction error increases as the observation size increases, which is consistent with Fig. 5. This indicates the difficulty of predicting wind power accurately during high fluctuation periods, especially for long-term forecasting. Most of the data points are above the diagonal line, which means that the prediction model tends to overestimate the actual values when they reach the peak. This could be due to the instability of the wind power data during high fluctuation periods, which makes it hard for the prediction model to find stable patterns. The wind power data has a high standard deviation and a low autocorrelation during high fluctuation periods, which indicates a high degree of randomness and unpredictability. It could also be due to the prediction model's limitation in capturing the sudden changes in the data, which leads to a premature reaction in forecasting the peak values. Nevertheless, TCOAT still outperforms other methods in terms of Pearson correlation coefficient (PCC), which reflects the linear relationship strength between the actual and predicted values. PCC is an important indicator of the prediction model's performance, as it measures how well the model can capture the trend and pattern of the data. TCOAT has a higher PCC than other methods for all four forecasting horizons.

5.4. Model ablation study

We designed the TCOAT model to capture the complex temporal and collaborative patterns in wind power data. To evaluate how each component contributes to the accuracy of the model, we conducted an ablation study by comparing TCOAT with six variants that remove one or more components. We tested the models on four different prediction horizons ($h = 1, 3, 5, 7$) and reported the results in Table 4.

The results show that TCOAT outperforms all the variants on all metrics and horizons, demonstrating the effectiveness of its novel components. Each component plays an important role in improving the performance of the model, and removing any component leads to a significant drop in performance. We discuss the impact of each component in detail below.

- **LTR**: This component captures the long-term changes in wind power generation by using a recurrent neural network to learn features from historical information. Removing LTR (w/o LTR) results in poor performance, especially for longer horizons. This indicates that LTR can capture the long-term dependencies in wind power data and help the model make more accurate predictions.
- **CAUs(DT)**: This component generates a new time series based on the characteristics of the input time series and the importance of each moment for future prediction. Removing CAUs(DT) (w/o



(a) $h = 1$. (b) $h = 3$.

(c) $h = 5$. (d) $h = 7$.

Fig. 6. The correlation visualizations of TCOAT predictions and three other benchmark predictions; Unit of h: day.

Table 4
Model ablation study. The best results are shown in bold, the second-best results are underlined, and the worst results are in wavy lines; Unit of h: day.

Model	$h = 1$			$h = 3$			$h = 5$			$h = 7$		
	MSE	MAE	CVRMSE	MSE	MAE	CVRMSE	MSE	MAE	CVRMSE			
TCOAT	**88.209052**	**7.390437**	**0.464018**	**165.119263**	**10.396696**	**0.634868**	**173.061798**	**10.564649**	**0.6499585**	**179.194824**	10.889776	**0.661714**
w/o LTR	99.306294	7.966326	0.492349	191.195865	10.802640	0.683161	179.499512	10.703563	0.661937	188.020944	10.940787	0.677814
w/o CAUs(DT)	93.373634	7.655972	0.477412	175.767138	10.559080	0.655006	173.306039	10.590805	0.650401	186.674304	10.944325	0.675381
w/o CAUs(SA)	99.292628	7.961520	0.492314	188.348081	10.984047	0.678013	183.273358	10.936804	0.668858	187.303388	10.957723	0.676519
w/o CAUs	99.980958	8.004636	0.494018	193.564978	10.947162	0.687379	183.344696	10.955248	0.668988	187.314881	10.949467	0.676537
w/o STR	98.671052	7.856977	0.490581	181.633734	10.621680	0.665434	184.227639	10.637395	0.670584	189.200436	**10.840509**	0.679934

CAUs(DT)) also leads to poor performance, except for a slight improvement at $h = 3$. This indicates that CAUs(DT) can enhance the temporal fusion structure by transforming the input time series in different directions.

- **CAUs(SA)**: This component assigns weights to directional transformation data in a certain direction dimension and ensures a balanced alignment. Removing CAUs(SA) (w/o CAUs(SA)) performs worse than w/o CAUs(DT), suggesting that CAUs(SA) can learn meaningful attention representation from different directions and improve the performance of the model.
- **CAUs**: This component is composed of CAUs(DT) and CAUs(SA). Removing the entire CAUs module (w/o CAUs) results in a significant drop in performance, indicating that CAUs can capture the collaborative patterns in wind power data and help the model make more accurate predictions.
- **STR**: This component captures the short-term variations of wind power generation by using a convolutional neural network to learn features from local information. Removing STR (w/o STR) also results in a significant drop in performance, showing that STR can capture the short-term dependencies in wind power data and help the model make more accurate predictions.

Therefore, the ablation study confirms that each component of TCOAT is effective and necessary for predicting wind power generation. The TCOAT model structure design considers not only the wind power

Table 5
Performance comparisons on wind power prediction. The best results are shown in bold, the second-best results are underlined, and the worst results are in wavy lines; Unit of *h*: day.

Model	MSE	MAE	CVRMSE
GAR	10.971498	2.755303	0.924539
AR	10.926890	2.717245	0.922657
VAR	12.407728	2.897383	0.983192
DLinear	10.545882	2.720891	0.906428
NLinear	10.913733	2.703481	0.922102
FiLM	12.570126	2.945149	0.989605
LSTM	9.837881	2.570797	0.875473
GRU	9.806232	2.630778	0.874064
ED	10.171904	2.734773	0.890211
CNN1D	11.293797	2.768536	0.93802
CRNN	10.473486	2.738216	0.903312
CRNNRes	11.132644	2.718664	0.931304
LSTNet	11.104560	2.703753	0.930128
Transformer	10.523831	2.774823	0.90548
Informer	10.515368	2.742601	0.905116
Autoformer	10.175673	2.563494	0.890376
FEDformer	12.323840	2.871534	0.979862
DSANet	10.310382	2.739258	0.896251
TPA-LSTM	12.195922	2.942518	0.974764
StemGNN	10.460149	2.762231	0.902736
GAIN	10.537903	2.751385	0.906085
MSL	10.367334	2.782756	0.898723
TCOAT	9.559267	**2.553933**	**0.862987**

influences but also the short-term and long-term time dependence and collaborative attention in the time series.

5.5. Generalization study

This subsection evaluates the generalization ability of our proposed TCOAT model by utilizing an auxiliary dataset, known as WSTD2. We compare TCOAT with twenty-two state-of-the-art methods for wind power forecasting, using three metrics: MSE, MAE, and CVRMSE. We consider a single time horizon: 1 day ahead (h = 1).

Before comparing TCOAT with other methods, we conducted some experiments to find the optimal settings for our model. We fixed the prediction horizon (h) and the output window size (H) at 1, the batch size (B) at 32, and the CAUs settings at 0, 1, and varied the window size (T) from 1 to 25. We used the same evaluation metrics and experimental settings as before, running each experiment five times and reporting the average results. The results indicated that the best performance was achieved when $T = 22$, so we set T to 22 for subsequent experiments. Next, we fixed the prediction horizon (h), the output window size (H), and the window size (T) to 1, 1, and 22, respectively, and varied the batch size (B) from 2^0 to 2^7. The results showed that the best performance was achieved when $B = 2^7$, so we set B to 2^7 for the final comparison with other methods.

We group the methods into nine categories based on their main techniques: linear models, linear model variations, recurrent neural networks (RNNs), CNN- and RNN-based models, self-attention-based models, CNN-RNN and attention models, graph attention-based models, shapelet learning model, and our proposed model. The results in Table 5 show that:

- Linear models (GAR, AR, and VAR) perform poorly, indicating that the wind power data has complex nonlinear and temporal patterns that cannot be captured by simple linear models. Among them, AR slightly outperforms GAR and VAR, suggesting that the wind power data has some autocorrelation structure.
- Linear model variations (DLinear, NLinear, FiLM) perform slightly better than linear models, indicating that the wind power data has some nonlinear patterns that can be captured by adding nonlinear activation functions or feature-wise linear modulation. However, FiLM performs the worst among all methods, suggesting that this technique is not suitable for wind power data.
- RNNs (LSTM, GRU, and ED) perform well, achieving the second and third-best results among all methods. This indicates that the wind power data has strong temporal dependencies that can be captured by RNNs.
- CNN- and RNN-based models (CNN1D, CRNN, CRNNRes, and LSTNet) perform worse than RNNs, indicating that the wind power data does not contain much spatial information that can be captured by CNNs. CRNNRes performs worse than CRNN, suggesting that the residual module does not find effective features. CNN1D, CRNNRes, and LSTNet have similar performance, suggesting they have similar limitations in modeling wind power data.
- Self-attention-based models (Transformer, Informer, Autoformer, and Fedformer) outperform the linear models but not the RNNs, indicating that the wind power data has some long-range dependencies that can be captured by self-attention, but also some short-term dependencies that are better captured by RNNs. Among them, Autoformer achieves the best MSE and the second-best MAE among all methods, indicating that it can learn effective features from the wind power data. Informer performs slightly worse than Autoformer, and Transformer slightly worse than Informer, suggesting that information attention and probabilistic time series modeling techniques are beneficial for wind power forecasting. Fedformer performs poorly, suggesting that the feature-enhanced dual transformer architecture is not suitable for wind power data.
- CNN-RNN and attention models, the two core components of the hybrid attention model (DSANet and TPA-LSTM), perform differently. DSANet outperforms the CNN- and RNN-based models but not the RNNs, indicating that the dual self-attention network can capture both local and global dependencies in wind power data. TPA-LSTM performs poorly, similar to FiLM and Fedformer, indicating that the temporal pattern attention technique is not effective for wind power data.
- Graph attention-based models (StemGNN and GAIN) perform similarly, outperforming the linear models, CNN- and RNN-based models, and self-attention-based models, but not the RNNs. This indicates that the graph attention technique can capture the spatial–temporal dependencies in wind power data. StemGNN performs slightly better than GAIN, suggesting that the spatial–temporal embedding technique is beneficial for wind power forecasting.
- MSL, which learns shapelets from the wind power data, outperforms the self-attention-based models and graph attention-based models, but still lags behind the RNNs. This indicates that the shapelet learning technique can capture some local patterns in wind power data, but not the global patterns.
- TCOAT, our proposed model, achieves the best results on all metrics, demonstrating the generalization ability of TCOAT. Compared to the best baseline GRU, TCOAT improves the MSE, MAE, and CVRMSE by 2.5%, 0.4%, and 1.26%, respectively.

In summary, we have shown that TCOAT can generalize well to different wind power datasets, and outperform the existing methods for wind power forecasting. This indicates that TCOAT can effectively capture the complex nonlinear and temporal patterns in wind power data, and provide accurate and reliable forecasts for wind power generation.

6. Conclusion and future work

Wind power forecasting stands as a pivotal task for the effective integration and management of wind energy systems. Accurate forecasting not only optimizes the operation and maintenance of wind turbines but also mitigates the uncertainty and risk associated with

power supply, thereby amplifying both the economic and environmental advantages of wind power generation. In this research, we introduced Temporal Collaborative Attention (TCOAT), a data-driven approach designed to capture the intricate temporal and spatial dependencies inherent in wind power generation data. TCOAT employs attention mechanisms to dynamically adjust the weights of each input variable and time step based on their contextual relevance for forecasting. Furthermore, the model incorporates collaborative attention units to assimilate both directional and global information from the input data. It also employs self-attention and cross-attention mechanisms to explicitly model the interactions and correlations among different variables or time steps. Additionally, TCOAT features a temporal fusion layer that effectively integrates long-term and short-term information through concatenation and mapping operations, as well as hierarchical feature extraction and aggregation.

To evaluate the performance of TCOAT, we conducted extensive experiments on two real-world wind power datasets from different regions with distinct climate conditions. Our empirical results, compared with twenty-two state-of-the-art methods, show that TCOAT surpasses them in terms of both accuracy and robustness, especially for short-term and very short-term forecasting horizons. A model ablation study further confirms the effectiveness of each component of TCOAT, while a parameter sensitivity analysis reveals the influence of various hyperparameters on the model's performance. The experiment using the second dataset as an additional dataset verifies the generality of the proposed model.

However, TCOAT also has some limitations and challenges that need to be addressed in future work. First, TCOAT does not provide any uncertainty quantification or probabilistic forecasts, which may affect the decision-making and risk management of wind power integration. Second, the current implementation of TCOAT does not consider the real-time scenario, which adapts to the changing dynamics or patterns of wind power generation over time, and thus may require periodic retraining or updating of the model. Third, we have only tested our method on the predictions using onshore datasets, while more complex remote or offshore wind farms need to be considered for testing, such as those that consider the sea state conditions. Furthermore, data privacy and security issues may also arise when sharing or transferring data across different parties or regions.

For future work, we aim to address these limitations and generalize our method to other forms of renewable energy, such as solar and hydropower, and to explore multi-source or multi-region forecasting. We also intend to enrich our model by incorporating external factors like grid load or market price, with the goal of enhancing forecasting accuracy and reliability. Moreover, we plan to delve into more advanced attention mechanisms, such as transformer or graph attention, to further improve the model's representation learning and feature extraction capabilities.

CRediT authorship contribution statement

Yue Hu: Writing – review & editing, Writing – original draft, Validation, Methodology, Conceptualization. **Hanjing Liu:** Writing – original draft, Software, Methodology, Conceptualization. **Senzhen Wu:** Writing – review & editing, Writing – original draft, Software, Methodology. **Yuan Zhao:** Writing – review & editing, Methodology, Conceptualization. **Zhijin Wang:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Investigation, Conceptualization. **Xiufeng Liu:** Writing – review & editing, Writing – original draft, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This research was supported in part by the Fujian Province Natural Science Foundation of Fujian Province (CN) (nos. 2021J01857, 2021J01859, and 2022J01335), and the RE-INTEGRATE project (no. 101118217) funded by the European Union Horizon 2020 research and innovation programme.

Appendix

Hyper-parameter setting

The parameters setting of the proposed method and benchmarks are listed in Table 6.

Table 6
Hyper-parameter settings.

Model	Parameter	Option range
LSTM GRU ED	Hidden size	$\{2^4, 2^5, 2^6\}$
DLinear	Decomposition kernel size	3–9 (2 per step)
FiLM	The dimension of the model	$\{2^4, 2^5, 2^6\}$
CNN1D	CNN kernel size	3–9 (2 per step)
	CNN out channels	$\{2^2, 2^3, 2^4, 2^5, 2^6\}$
CRNN	GRU hidden size	$\{2^4, 2^5, 2^6\}$
	GRU layers	1–3 (1 per step)
CRNNRes	Residual window size	1–7 (1 per step)
	Residual ratio	0.1–0.5 (0.1 per step)
LSTNet	Skip window size	1–7 (1 per step)
	Skip GRU hidden size	$\{2^4, 2^5, 2^6\}$
	Skip GRU layers	1–3 (1 per step)
Transformer	Encoder layers	1–3 (1 per step)
Informer	Decoder layers	1–3 (1 per step)
Autoformer	The label length	1–10 (1 per step)
FEDformer	The numbers of heads	$\{2^2, 2^3, 2^4\}$
	The dimension of the model	$\{2^4, 2^5, 2^6\}$
DSANet	CNN kernel size	3–9 (2 per step)
	CNN out channels	$\{2^2, 2^3, 2^4, 2^5, 2^6\}$
	Attention layers	1–3 (1 per step)
	The numbers of heads	$\{2^2, 2^3, 2^4\}$
	The dimension of the model	$\{2^4, 2^5, 2^6\}$
TPA-LSTM	GRU hidden size	$\{2^4, 2^5, 2^6\}$
	GRU layers	1–3 (1 per step)
StemGNN	Residual window size	1–10 (1 per step)
	Block size	1–10 (1 per step)
	Leaky rate	0.1–0.3 (0.1 per step)
GAIN	GAT hidden size	$\{2^4, 2^5, 2^6\}$
	The number of heads of GAT	$\{2^0, 2^1, 2^2, 2^3, 2^8\}$
MSL	Shapelet size	$\{2^2, 2^3, 2^4, 2^5, 2^6\}$

Abbreviation

The meanings of abbreviations are listed in Table 7.

Table 7
Abbreviations and meanings.

Abbreviation	Meaning
AM	Attention Mechanism
ANN	Artificial Neural Network
AR	Autoregressive
ARMA	Auto Regressive Moving Average
ARIMA	Auto Regressive Integrated Moving Average
CAU	Collaborative Attention Unit
CNN	Convolutional Neural Network
CNN1D	One-Dimensional CNN
CRNN	Convolutional Recurrent Neural Network

(*continued on next page*)

Table 7 (*continued*).

Abbreviation	Meaning
CRNNRes	Residual Convolutional Recurrent Neural Network
CVRMSE	Coefficient of Variation of Root Mean Square Error
DNN	Deep Neural Network
DSANet	Dual Self-Attention Network
DT	Directional Transformation
ED	Encoder–Decoder
ENTSO-E	European Network of Transmission System Operators of Electricity
FEDformer	Feature-Enhanced Dual Transformer
FiLM	Feature-Wise Linear Modulation
GAIN	Gated Multi-scale Aggregation Network
GRU	Gated Recurrent Unit
Informer	Information Attention-based Network
LSTM	Long Short-Term Memory
LSTNet	Long- and Short-Term Network
LTR	Long-term Temporal Representation
MAE	Mean Absolute Error
ML	Machine Learning
MSE	Mean Square Error
MSL	Multivariate Shapelet Learning
MTS	Multi-Temporal Scale
NWP	Numerical Weather Prediction
PM	Persistence Model
RNN	Recurrent Neural Network
RF	Random Forest
SA	Symmetric Attention
SGD	Stochastic Gradient Descent
StemGNN	Spatial-Temporal Embedding with Multi-Graph CNN
STR	Short-term Temporal Representation
TCOAT	Temporal Collaborative Attention
TPA	Temporal Pattern Attention with LSTM
WPF	Wind Power Forecasting
WSTD2	Wind Spatio-Temporal Dataset2

References

[1] World Wind Energy Association. WWEA annual report 2022. 2023.

[2] Wang Y, Zou R, Liu F, Zhang L, Liu Q. A review of wind speed and wind power forecasting with deep neural networks. Appl Energy 2021;304:117766. http://dx.doi.org/10.1016/j.apenergy.2021.117766.

[3] Liu H, Chen C. Data processing strategies in wind energy forecasting models and applications: A comprehensive review. Appl Energy 2019;249:392–408. http://dx.doi.org/10.1016/j.apenergy.2019.04.188.

[4] Suman A. Role of renewable energy technologies in climate change adaptation and mitigation: A brief review from nepal. Renew Sustain Energy Rev 2021;151:111524. http://dx.doi.org/10.1016/j.rser.2021.111524.

[5] González-Sopeña J, Pakrashi V, Ghosh B. An overview of performance evaluation metrics for short-term statistical wind power forecasting. Renew Sustain Energy Rev 2021;138:110515. http://dx.doi.org/10.1016/j.rser.2020.110515.

[6] Shabbir N, AhmadiAhangar R, Kütt L, Iqbal MN, Rosin A. Forecasting short term wind energy generation using machine learning. In: Proceedings of the 60th international scientific conference on power and electrical engineering of Riga Technical University. Riga, Latvia: IEEE; 2019, p. 1–4. http://dx.doi.org/10.1109/RTUCON48111.2019.8982365.

[7] Duan J, Wang P, Ma W, Fang S, Hou Z. A novel hybrid model based on nonlinear weighted combination for short-term wind power forecasting. Int J Electr Power Energy Syst 2022;134:107452. http://dx.doi.org/10.1016/j.ijepes.2021.107452.

[8] Hu W, Yang Q, Chen H-P, Yuan Z, Li C, Shao S, et al. New hybrid approach for short-term wind speed predictions based on preprocessing algorithm and optimization theory. Renew Energy 2021;179:2174–86. http://dx.doi.org/10.1016/j.ijepes.2021.107452.

[9] Karamichailidou D, Kaloutsa V, Alexandridis A. Wind turbine power curve modeling using radial basis function neural networks and Tabu search. Renew Energy 2021;163:2137–52. http://dx.doi.org/10.1016/j.renene.2020.10.020.

[10] Banik A, Behera C, Sarathkumar TV, Goswami AK. Uncertain wind power forecasting using LSTM-based prediction interval. IET Renew Power Gener 2020;14(14):2657–67. http://dx.doi.org/10.1049/iet-rpg.2019.1238.

[11] Quan H, Khosravi A, Yang D, Srinivasan D. A survey of computational intelligence techniques for wind power uncertainty quantification in smart grids. IEEE Trans Neural Netw Learn Syst 2019;31(11):4582–99. http://dx.doi.org/10.1109/TNNLS.2019.2956195.

[12] Constantinescu EM, Zavala VM, Rocklin M, Lee S, Anitescu M. A computational framework for uncertainty quantification and stochastic optimization in unit commitment with wind power generation. IEEE Trans Power Syst 2010;26(1):431–41. http://dx.doi.org/10.1109/TPWRS.2010.2048133.

[13] He Q, Wang J, Lu H. A hybrid system for short-term wind speed forecasting. Appl Energy 2018;226:756–71. http://dx.doi.org/10.1016/j.apenergy.2018.06.053.

[14] Ferreira M, Santos A, Lucio P. Short-term forecast of wind speed through mathematical models. Energy Rep 2019;5:1172–84. http://dx.doi.org/10.1016/j.egyr.2019.05.007.

[15] Wang J, Li Y. An innovative hybrid approach for multi-step ahead wind speed prediction. Appl Soft Comput 2019;78:296–309. http://dx.doi.org/10.1016/j.asoc.2019.02.034.

[16] Zhang H, Peng Z, Tang J, Dong M, Wang K, Li W. A multi-layer extreme learning machine refined by sparrow search algorithm and weighted mean filter for short-term multi-step wind speed forecasting. Sustain Energy Technol Assess 2022;50:101698. http://dx.doi.org/10.1016/j.seta.2021.101698.

[17] Wang H, Han S, Liu Y, Yan J, Li L. Sequence transfer correction algorithm for numerical weather prediction wind speed and its application in a wind power forecasting system. Appl Energy 2019;237:1–10. http://dx.doi.org/10.1016/j.apenergy.2018.12.076.

[18] Liu C, Zhang X, Mei S, Zhen Z, Jia M, Li Z, et al. Numerical weather prediction enhanced wind power forecasting: Rank ensemble and probabilistic fluctuation awareness. Appl Energy 2022;313:118769. http://dx.doi.org/10.1016/j.apenergy.2022.118769.

[19] Wood N. Wind flow over complex terrain: A historical perspective and the prospect for large-eddy modelling. Bound-Lay Meteorol 2000;96:11–32. http://dx.doi.org/10.1023/A:1002017732694.

[20] Chandra DR, Kumari MS, Sydulu M. A detailed literature review on wind forecasting. In: Proceedings of international conference on power, energy and control. Dindigul, India: IEEE; 2013, p. 630–4. http://dx.doi.org/10.1109/ICPEC.2013.6527734.

[21] Kavasseri RG, Seetharaman K. Day-ahead wind speed forecasting using f-ARIMA models. Renew Energy 2009;34(5):1388–93. http://dx.doi.org/10.1016/j.renene.2008.09.006.

[22] Meng A, Ge J, Yin H, Chen S. Wind speed forecasting based on wavelet packet decomposition and artificial neural networks trained by crisscross optimization algorithm. Energy Convers Manage 2016;114:75–88. http://dx.doi.org/10.1016/j.enconman.2016.02.013.

[23] Lu P, Ye L, Zhao Y, Dai B, Pei M, Tang Y. Review of meta-heuristic algorithms for wind power prediction: Methodologies, applications and challenges. Appl Energy 2021;301:117446. http://dx.doi.org/10.1016/j.apenergy.2021.117446.

[24] Feng C, Zhang J. Wind power and ramp forecasting for grid integration. Adv Wind Turb Technol 2018;299–315. http://dx.doi.org/10.1007/978-3-319-78166-2_11.

[25] Fischereit J, Brown R, Larsén XG, Badger J, Hawkes G. Review of mesoscale wind-farm parametrizations and their applications. Bound-Lay Meteorol 2022;182(2):175–224. http://dx.doi.org/10.1007/s10546-021-00652-y.

[26] Jacondino WD, da Silva Nascimento AL, Calvetti L, Fisch G, Beneti CAA, da Paz SR. Hourly day-ahead wind power forecasting at two wind farms in northeast Brazil using WRF model. Energy 2021;230:120841. http://dx.doi.org/10.1016/j.energy.2021.120841.

[27] Wang A, Xu L, Li Y, Xing J, Chen X, Liu K, et al. Random-forest based adjusting method for wind forecast of WRF model. Comput Geosci 2021;155:104842. http://dx.doi.org/10.1016/j.cageo.2021.104842.

[28] Zheng L, Zhou S, Yu Y, Shang Y, Gao Z. Short-term wind power prediction model based on WRF-RF model. In: Proceedings of the 8th international conference on cloud computing and big data analytics. Chengdu, China: IEEE; 2023, p. 599–604. http://dx.doi.org/10.1109/ICCCBDA56900.2023.10154834.

[29] Zhao J, Guo Z-H, Su Z-Y, Zhao Z-Y, Xiao X, Liu F. An improved multi-step forecasting model based on WRF ensembles and creative fuzzy systems for wind speed. Appl Energy 2016;162:808–26. http://dx.doi.org/10.1016/j.apenergy.2015.10.145.

[30] Giebel G, Kariniotakis G. Wind power forecasting—A review of the state of the art. Renew Energy Forecast 2017;59–109. http://dx.doi.org/10.1016/B978-0-08-100504-0.00003-2.

[31] Pinson P. Wind energy: Forecasting challenges for its operational management. Statist Sci 2013;28(4):564–85. http://dx.doi.org/10.1214/13-STS445.

[32] Li Y-L, Zhu Z-A, Chang Y-K, Chiang C-K. Short-term wind power forecasting by advanced machine learning models. In: Proceedings of international symposium on computer, consumer and control. Taiwan: IEEE; 2020, p. 412–5. http://dx.doi.org/10.1109/IS3C50286.2020.00112.

[33] Fang J, Peringer A, Stupariu M-S, Pătru-Stupariu I, Buttler A, Golay F, et al. Shifts in wind energy potential following land-use driven vegetation dynamics in complex terrain. Sci Total Environ 2018;639:374–84. http://dx.doi.org/10.1016/j.scitotenv.2018.05.083.

[34] Alam J. Interaction of vortex stretching with wind power fluctuations. Phys Fluids 2022;34(7). http://dx.doi.org/10.1063/5.0099347.

[35] Cao Y, Liu Y, Zhang D, Wang W, Chen Z. Wind power ultra-short-term forecasting method combined with pattern-matching and ARMA-model. In: Proceedings of Grenoble Conference. Grenoble, France: IEEE; 2013, p. 1–4. http://dx.doi.org/10.1109/PTC.2013.6652257.

[36] Milligan M, Schwartz MN, Wan Y. Statistical wind power forecasting for U.S. wind farms. In: Proceedings of the 17th Conference on Probability and Statistics in the Atmospheric Sciences. Seattle, Washington: American Meteorological Society; 2003, p. 1–10. http://dx.doi.org/10.1109/PTC.2013.6652257.
[37] Ahn E, Hur J. A short-term forecasting of wind power outputs using the enhanced wavelet transform and arimax techniques. Renew Energy 2023;212:394–402. http://dx.doi.org/10.1016/j.renene.2023.05.048.
[38] Zhang W, Lin Z, Liu X. Short-term offshore wind power forecasting-a hybrid model based on Discrete Wavelet Transform (DWT), Seasonal Autoregressive Integrated Moving Average (SARIMA), and deep-learning-based Long Short-Term Memory (LSTM). Renew Energy 2022;185:611–28. http://dx.doi.org/10.1016/j.renene.2021.12.100.
[39] Sheoran S, Pasari S. Efficacy and application of the window-sliding ARIMA for daily and weekly wind speed forecasting. J Renew Sustain Energy 2022;14(5):053305. http://dx.doi.org/10.1063/5.0108847.
[40] Singh PK, Singh N, Negi R. Wind power forecasting using hybrid ARIMA-ANN technique. In: Proceedings of ambient communications and computer systems. Singapore: Springer; 2019, p. 209–20. http://dx.doi.org/10.1007/978-981-13-5934-7_19.
[41] Bazionis IK, Georgilakis PS. Review of deterministic and probabilistic wind power forecasting: Models, methods, and future research. Electricity 2021;2(1):13–47. http://dx.doi.org/10.3390/electricity2010002.
[42] Messner JW, Pinson P, Browell J, Bjerregård MB, Schicker I. Evaluation of wind power forecasts—An up-to-date view. Wind Energy 2020;23(6):1461–81. http://dx.doi.org/10.1002/we.2497.
[43] Olson JB, Kenyon JS, Djalalova I, Bianco L, Turner DD, Pichugina Y, et al. Improving wind energy forecasting through numerical weather prediction model development. Bull Am Meteorol Soc 2019;100(11):2201–20. http://dx.doi.org/10.1175/BAMS-D-18-0040.1.
[44] Lin Z, Liu X. Wind power forecasting of an offshore wind turbine based on high-frequency SCADA data and deep learning neural network. Energy 2020;201:117693. http://dx.doi.org/10.1016/j.energy.2020.117693.
[45] Díaz-Vico D, Torres-Barrán A, Omari A, Dorronsoro JR. Deep neural networks for wind and solar energy prediction. Neural Process Lett 2017;46:829–44. http://dx.doi.org/10.1007/s11063-017-9613-7.
[46] Abedinia O, Bagheri M, Naderi MS, Ghadimi N. A new combinatory approach for wind power forecasting. IEEE Syst J 2020;14(3):4614–25. http://dx.doi.org/10.1109/JSYST.2019.2961172.
[47] Huang B, Liang Y, Qiu X. Wind power forecasting using attention-based recurrent neural networks: a comparative study. IEEE Access 2021;9:40432–44. http://dx.doi.org/10.1109/ACCESS.2021.3065502.
[48] Alcántara A, Galván IM, Aler R. Direct estimation of prediction intervals for solar and wind regional energy forecasting with deep neural networks. Eng Appl Artif Intell 2022;114:105128. http://dx.doi.org/10.1016/j.engappai.2022.105128.
[49] Liu X, Cao Z, Zhang Z. Short-term predictions of multiple wind turbine power outputs based on deep neural networks with transfer learning. Energy 2021;217:119356. http://dx.doi.org/10.1016/j.energy.2020.119356.
[50] Sun S, Liu Y, Li Q, Wang T, Chu F. Short-term multi-step wind power forecasting based on spatio-temporal correlations and transformer neural networks. Energy Convers Manage 2023;283:116916. http://dx.doi.org/10.1016/j.enconman.2023.116916.
[51] Wu Q, Zheng H, Guo X, Liu G. Promoting wind energy for sustainable development by precise wind speed prediction based on graph neural networks. Renew Energy 2022;199:977–92. http://dx.doi.org/10.1016/j.renene.2022.09.036.
[52] Huang Y, Zhao Y, Wang Z, Liu X, Liu H, Fu Y. Explainable district heat load forecasting with active deep learning. Appl Energy 2023;350:121753. http://dx.doi.org/10.1016/j.apenergy.2023.121753.
[53] Liu X, Zhang Y, Zhen Z, Xu F, Wang F, Mi Z. Spatio-temporal graph neural network and pattern prediction based ultra-short-term power forecasting of wind farm cluster. IEEE Trans Ind Appl 2023;PP(99):1–10. http://dx.doi.org/10.1109/TIA.2023.10269018.
[54] Lu W, Duan J, Wang P, Ma W, Fang S. Short-term wind power forecasting using the hybrid model of improved variational mode decomposition and maximum mixture correntropy long short-term memory neural network. Int J Electr Power Energy Syst 2023;144:108552. http://dx.doi.org/10.1016/j.ijepes.2022.108552.
[55] Wu Z, Luo G, Yang Z, Guo Y, Li K, Xue Y. A comprehensive review on deep learning approaches in wind forecasting applications. CAAI Trans Intell Technol 2022;7(2):129–43. http://dx.doi.org/10.1049/cit2.12076.
[56] Jia Y. Attention mechanism in machine translation. In: Proceedings of the 3rd international conference on electrical, mechanical and computer engineering, vol. 1314, no. 1. Guizhou, China: IOP Publishing; 2019, 012186. http://dx.doi.org/10.1088/1742-6596/1314/1/012186.
[57] Qiu D, Yang B. Text summarization based on multi-head self-attention mechanism and pointer network. Complex Intell Syst 2022;1–13. http://dx.doi.org/10.1007/s40747-021-00527-2.
[58] Tian C, Niu T, Wei W. Developing a wind power forecasting system based on deep learning with attention mechanism. Energy 2022;257:124750. http://dx.doi.org/10.1016/j.energy.2022.124750.
[59] Aslam M, Kim J-S, Jung J. Multi-step ahead wind power forecasting based on dual-attention mechanism. Energy Rep 2023;9:239–51. http://dx.doi.org/10.1016/j.egyr.2022.11.167.
[60] Lin J, Ma J, Zhu J, Cui Y. Short-term load forecasting based on LSTM networks considering attention mechanism. Int J Electr Power Energy Syst 2022;137:107818. http://dx.doi.org/10.1109/iSPEC53008.2021.9735593.
[61] Niu D, Yu M, Sun L, Gao T, Wang K. Short-term multi-energy load forecasting for integrated energy systems based on CNN-BiGRU optimized by attention mechanism. Appl Energy 2022;313:118801. http://dx.doi.org/10.1016/j.apenergy.2022.118801.
[62] Zhang G, Bai X, Wang Y. Short-time multi-energy load forecasting method based on CNN-Seq2Seq model with attention mechanism. Mach Learn Appl 2021;5:100064. http://dx.doi.org/10.1016/j.mlwa.2021.100064.
[63] Yuan Y, Chen Z, Wang Z, Sun Y, Chen Y. Attention mechanism-based transfer learning model for day-ahead energy demand forecasting of shopping mall buildings. Energy 2023;270:126878. http://dx.doi.org/10.1016/j.energy.2023.126878.
[64] Tekin SF, Karaahmetoglu O, Ilhan F, Balaban I, Kozat SS. Spatio-temporal weather forecasting and attention mechanism on convolutional LSTMs. 2021, http://dx.doi.org/10.48550/arXiv.2102.00696, arXiv preprint arXiv:2102.00696.
[65] Khan ZA, Hussain T, Baik SW. Dual stream network with attention mechanism for photovoltaic power forecasting. Appl Energy 2023;338:120916. http://dx.doi.org/10.1016/j.apenergy.2023.120916.
[66] Ding Y. Data science for wind energy. Boca Raton: Chapman and Hall/CRC; 2019, http://dx.doi.org/10.1201/9780429027629.
[67] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: Proceedings of the 3rd international conference on learning representations. San Diego, CA, USA: OpenReview.net; 2015, p. 1–15. http://dx.doi.org/10.1145/1830483.1830503.
[68] Kisvari A, Lin Z, Liu X. Wind power forecasting–A data-driven method along with gated recurrent neural network. Renew Energy 2021;163:1895–909. http://dx.doi.org/10.1016/j.renene.2020.10.119.
[69] Zhou K, Wang WY, Hu T, Wu CH. Comparison of time series forecasting based on statistical ARIMA model and LSTM with attention mechanism. In: Proceedings of the 2nd international conference on artificial intelligence and computer science, vol. 1631, no. 1. Hangzhou, Zhejiang, China: IOP Publishing; 2020, 012141. http://dx.doi.org/10.1088/1742-6596/1631/1/012141.
[70] Torres JF, Hadjout D, Sebaa A, Martínez-Álvarez F, Troncoso A. Deep learning for time series forecasting: A survey. Big Data 2021;9(1):3–21. http://dx.doi.org/10.1089/big.2020.0159.
[71] Box GE, Jenkins GM, Reinsel GC, Ljung GM. Time series analysis: forecasting and control. John Wiley & Sons; 2015.
[72] Zeng A, Chen M, Zhang L, Xu Q. Are transformers effective for time series forecasting? In: Proceedings of the 37th AAAI conference on artificial intelligence, vol. 37, no. 9. 2023, p. 11121–8. http://dx.doi.org/10.1609/aaai.v37i9.26317.
[73] Zhou T, Ma Z, Wang X, Wen Q, Sun L, Yao T, et al. FiLM: Frequency improved Legendre memory model for long-term time series forecasting. In: Proceedings of the 36th international conference on neural information processing systems, vol. 35. New Orleans, LA, USA; 2022, p. 12677–90. http://dx.doi.org/10.48550/arXiv.2109.03254.
[74] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80. http://dx.doi.org/10.1162/neco.1997.9.8.1735.
[75] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014, http://dx.doi.org/10.48550/arXiv.1412.3555, CoRR abs/1412.3555.
[76] Cho K, van Merrienboer B, Bahdanau D, Bengio Y. On the properties of neural machine translation: Encoder-decoder approaches. In: Proceedings of the 8th workshop on syntax, semantics and structure in statistical translation. Doha, Qatar: Association for Computational Linguistics; 2014, p. 103–11. http://dx.doi.org/10.3115/v1/W14-4012.
[77] Shi X, Chen Z, Wang H, Yeung D-Y, Wong W-K, Woo W-c. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: Proceedings of the 28th international conference on advances in neural information processing systems. Montreal, Quebec, Canada; 2015, p. 802–10. http://dx.doi.org/10.48550/arXiv.1506.04214.
[78] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE Computer Society; 2016, p. 770–8. http://dx.doi.org/10.1109/CVPR.2016.90.
[79] Lai G, Chang W, Yang Y, Liu H. Modeling long- and short-term temporal patterns with deep neural networks. In: Proceedings of the 41st international conference on research and development in information retrieval. Ann Arbor, MI, USA: ACM; 2018, p. 95–104. http://dx.doi.org/10.1145/3209978.3210006.
[80] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Proceedings of the 31th international conference on neural information processing systems, vol. 30. Long Beach, CA, USA: Curran Associates, Inc.; 2017, p. 5998–6008. http://dx.doi.org/10.5555/3295222.3295349.
[81] Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the 35th AAAI conference on artificial intelligence. Virtual Event: AAAI Press; 2021, p. 11106–15. http://dx.doi.org/10.1609/aaai.v35i12.17325.

[82] Wu H, Xu J, Wang J, Long M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: Proceedings of the 34th international conference on neural information processing systems. Virtual; 2021, p. 22419–30. http://dx.doi.org/10.48550/arXiv.2106.13008.

[83] Zhou T, Ma Z, Wen Q, Wang X, Sun L, Jin R. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: Proceedings of the 39th international conference on machine learning, vol. 162. Baltimore, Maryland, USA: PMLR; 2022, p. 27268–86. http://dx.doi.org/10.48550/arXiv.2201.12740.

[84] Huang S, Wang D, Wu X, Tang A. DSANet: Dual self-attention network for multivariate time series forecasting. In: Proceedings of the 28th ACM international conference on information and knowledge management. Beijing, China: ACM; 2019, p. 2129–32. http://dx.doi.org/10.1145/3357384.3358132.

[85] Shih S, Sun F, Lee H. Temporal pattern attention for multivariate time series forecasting. Mach Learn 2019;108(8–9):1421–41. http://dx.doi.org/10.1007/s10994-019-05815-0.

[86] Cao D, Wang Y, Duan J, Zhang C, Zhu X, Huang C, et al. Spectral temporal graph neural network for multivariate time-series forecasting. In: Proceedings of the 34th international conference on advances in neural information processing systems. Virtual; 2020, p. 17766–78. http://dx.doi.org/10.48550/arXiv.2103.077191.

[87] Wang Z, Liu X, Huang Y, Zhang P, Fu Y. A multivariate time series graph neural network for district heat load forecasting. Energy 2023;278:127911. http://dx.doi.org/10.1016/j.energy.2023.127911.

[88] Wang Z, Cai B. COVID-19 cases prediction in multiple areas via shapelet learning. Appl Intell 2022;52(1):595–606. http://dx.doi.org/10.1007/s10489-021-02391-6.

[89] Chen Z, Li Z, Zhang X. Wind power forecasting based on LSTM neural network. Int J Recent Technol Eng 2019;8(2S11):3810–4. http://dx.doi.org/10.35940/ijrte.B1298.0982S1119.