SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería RIMNI in cooperation with CIMNE

Enhancing Wind Power Forecasting Using Hybrid Multi-Head Attention and 1-Dimensional Convolutional Neural Networks

Saifur Rahman[1,*], Abdullah Shaher[1], Nabeel Ahmed Khan[2], Muhammad Abubakar[3], Zohaib Mushtaq[4], Hatim Alwadie[1], Ayman Taher Hindi[1], Muhammad Irfan[1] and Saleh Al Dawsari[1,5,*]

[1]Electrical Engineering Department, College of Engineering, Najran University, Najran, 61441, Saudi Arabia

[2]Namal University, Center for AI & Big Data, Mianwali, 42250, Pakistan

[3]Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan

[4]Department of Electrical Electronics and Computer Systems, College of Engineering and Technology, University of Sargodha, Sargodha, 40100, Pakistan

[5]School of Engineering, Cardiff University, Cardiff, CF24 3AA, UK

ABSTRACT

The accurate forecasting of wind power plays a veritable part in integrating renewable energy from wind turbines into power grids. Wind power, being a highly volatile mode of energy generation owing to temporal variations and complex weather patterns, renders reliable predictions essential for energy management and grid stability. In order to tackle this, we propose a hybrid Multi-Head Attention and 1D-Convolutional Neural Network (MHA-CNN) architecture that combines attention mechanisms and convolutional layers to capture both long-term dependencies and localized features in time-series data from a Supervisory Control and Data Acquisition (SCADA) system. The model effectively improves forecasting performance by attaining an R^2 score of 99.42 for hour-ahead and 96.52 for day-ahead predictions on a 50,540-sample, 10-min SCADA dataset using 5-fold chronological cross-validation, outperforming traditional methods without any manual feature engineering. The proposed method is also evaluated across multiple scenarios to assess the robustness of the proposed approach.

OPEN ACCESS

Received: 01/10/2025

Accepted: 10/11/2025

DOI
10.23967/j.rimni.2025.10.74091

Keywords:
Convolutional neural network
wind turbine
supervisory control and data acquisition
energy forecasting
multi-head attention
renewable energy
wind power

1 Introduction

Wind power forecasting is an indispensable system, especially for better integrating renewable energies from wind turbines into modern power systems. Precise wind generation forecasts are essentially needed for grid stability, energy trading, and the minimization of dependency on fossil fuel-based energy reserves. Since the sole factor affecting wind speed is given by the nature of atmospheric

*Correspondence: Saifur Rahman, Saleh AL Dawsari (srrahman@nu.edu.sa, aldawsarisa@cardiff.ac.uk).

conditions, the normal variations in wind speed make the forecasting of wind power a very complex task that requires sophisticated models. These vary from simple statistical methods to more advanced machine learning and deep learning models, each of which tackles some other temporal and spatial challenge in forecasting [1,2]. The increase in the global penetration of renewable energies means the need for robust forecasting models that capture uncertainties and can reliably predict power is growing into an issue of paramount importance if efficiency and sustainability are to be maintained across energy systems [3].

Recent developments in data-driven models, particularly in the emerging area of artificial intelligence, have endowed the systems with the capability of developing more accurate and adaptive forecasting systems. These approaches utilize historical data, meteorological inputs, and time-series features for short-term, medium-term, and long-term forecasting of wind power output [4]. Due to the intermittent and highly stochastic lineament of wind power, machine learning and deep learning models use the Long Short Term Memory (LSTM), Convolutional Neural Networks (CNN), and hybrid approaches for tackling these complexities. These deep learning models are able not only to learn nonlinear patterns but also to handle high-dimensional data with great efficiency, which becomes really crucial in improving prediction accuracy [5]. One of the major trade-offs between model accuracy and computational efficiency is still active, in terms of research and development, for real-time applications.

Deep learning models like LSTM, Gated Recurrent Unit (GRU), and CNN have always done better in comparison with traditional statistical models in the premises of wind power forecast studies because of their ability to capture nonlinear patterns in complex datasets [6]. That is characterized by great temporal dependencies and also involves different environmental features in one model—for instance, temperature and pressure [7]. Particularly, Long Short-Term Memory and CNN are much better for the short-term forecasting, where most active fluctuations in the wind pattern are evident. Nevertheless, even with these advances, some crucial challenges at the level of data quality and computational complexity still remain to be overcome by deep learning models. Often, high-dimensional data and noisy inputs badly reduce model generalizability, making accurate predictions difficult when large, high-quality datasets are unavailable. The problem is that, even though these models achieve higher performance in terms of accuracy, the computational requirements in terms of very long training and a lot of resources make them impractical for real-time applications [8,9].

Hybrid models that combine deep learning with either statistical or signal-processing methods come as such a limitation of deep learning techniques in stand-alone applications. By leveraging decompositions, such as wavelet transforms, hybrid models may isolate different frequency components in the wind data and improve both short-term and long-term predictions [10,11]. These models give good estimation accuracy with low computational cost, particularly for uncertain data conditions [12]. However, they are less operational in real time due to their higher complexity. Hybrid approaches may show better accuracy, but the increase in computational burden still remains strong and difficult to deploy in any regular system operation [13]. Despite that fact, hybrid models demonstrate superior accuracy both for short-term wind forecasting and ultra-short-term, which is a promising research direction.

These recently adopted transformer-based models, such as the Frequency-Enhanced Transformer, are state-of-the-art and constitute a turning point in wind power forecasting [14]. Unlike other conventional deep learning models, transformers were designed to grasp long-term and short-term dependencies that may exist in wind power data through the attention mechanism by dynamically

selecting and focusing on time steps relevant for prediction. In this regard, transformers are particularly efficient at dealing with the fluctuating and unpredictable nature of wind data. Take, for instance, FEDformer, which outperforms the performance of traditional models by a great margin, upwards of one order of magnitude, because it can thus much better cope with the challenges high-frequency components in wind data pose. This ability to handle both temporal scales simultaneously turns transformers into a very valuable solution for both short-term and ultra-short-term forecasting [15]. That being said, transformer-based models come with their own set of disadvantages as well. The most important quality of them is their ability to parallelize computations, which allows them to handle larger datasets much better than the recurrent model, such as the LSTM [16]. However, transformers usually need a lot of data to work efficiently, and due to their intricate architecture—making the training harder and more computationally expensive—they cannot be run in real-time forecasting systems easily. In addition, though attention mechanisms do allow the model to guide its focus on different time steps, sometimes broader structures present in the data cannot be adequately learned. By just attending over it through fewer layers employed in designing them is one issue that integrated models try to resolve. When we mix transformers with deep learning models, it makes our model more powerful than before, but at the same time adds to the complexity and computational overhead. However, deploying transformer-based models in real-time and resource-constrained scenarios is difficult, even though they far surpass conventional methods in modeling non-linear and global extensions to challenges (e.g., FEDformer [17])

Despite the advancements in deep learning and hybrid models for wind power forecasting, several gaps remain in the literature. Transformer-based models, while excelling in capturing long-range dependencies, suffer from specific, well-documented limitations that hinder their practical deployment in wind forecasting systems:

1. Quadratic computational complexity: Standard Transformers scale as OT^2 with sequence length T, making them prohibitively slow for high-resolution SCADA data (e.g., 10-min intervals over days). Even sparse variants like FEDformer reduce this to OT (log T), but inference latency remains >500 ms per sample on edge hardware.
2. High data dependency: Transformers require large, clean datasets to avoid overfitting. On single-turbine SCADA data (<60 k samples), performance drops significantly ($R^2 < 0.85$).
3. Poor real-time performance: Full-sequence attention prevents parallel processing of future timesteps, violating strict latency requirements (<100 ms) in grid control loops.
4. Over-smoothing in local patterns: Global attention dilutes fine-grained fluctuations critical for ultra-short-term forecasting.

Hybrid models integrating CNNs mitigate some issues but often increase complexity without guaranteed gains. Most current models are optimized for short-term forecasting, with limited focus on scalable, low-latency, data-efficient solutions across multiple horizons. This work addresses these gaps by proposing MHA-CNN, a lightweight hybrid that combines local feature extraction (1D-CNN) with selective long-range modeling (MHA) and adaptive downsampling (dynamic mean pooling), achieving real-time inference (<50 ms) on CPU while maintaining high accuracy on limited data.

Recent hybrid models have explored combining CNNs and attention mechanisms for time-series forecasting. CNN-LSTM stacks achieve good short-term accuracy but fail to capture long-range dependencies. Temporal Convolutional Transformers (TCT) [18] incorporate dilated convolutions before the attention layer, but their fixed dilation rates limit flexibility. Conv-Trans applies a 1D-CNN followed by a full Transformer, resulting in a computational complexity of $O(T^2)$. In contrast,

MHA-CNN introduces dynamic mean pooling to bridge the CNN and multi-head attention (MHA) components, allowing adaptive resolution reduction without relying on fixed kernels or frequency-domain priors. This design achieves higher efficiency and robustness, particularly on single-turbine datasets

2 Background on Core Components

The proposed MHA-CNN builds on two foundational architectures i.e., Transformers and 1D Convolutional Neural Networks (CNNs) [19]. Their key properties are summarized below.

1D-CNNs apply learnable filters to extract local temporal patterns through convolution, defined as:

$$y[t] = \sigma(\Sigma_{1=0}{}^{k-1} \times x[t+i] + b) \tag{1}$$

followed by non-linear activation and pooling.

Transformers, on the other hand, use scaled dot-product attention, defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{d_k})V \tag{2}$$

and extend this to a multi-head form to capture diverse relationships.

This work leverages CNNs for efficient local modeling and multi-head attention (MHA) for selective global context, thereby avoiding the computational overhead of full-sequence attention. A brief comparison between 1D-CNN and transformer components is provided in Table 1.

Table 1: Comparison of 1D-CNN and transformer components

Property	1D-CNN	Transformer (Self-Attention)
Feature extraction	Local, hierarchical	Global, long-range
Complexity	O (K × T × F)	O (T^2 × d)
Parallelism	High (per layer)	High (within layer)
Inductive bias	Translation invariance	None (uses position embeddings)
Data efficiency	High	Low
Real-time suitability	Excellent	Poor (for long sequences)

3 Methodology

3.1 Pre-Processing

The 2018 SCADA dataset contains 50,540 valid samples at a 10-min resolution. Preprocessing was performed in four stages:

1. **Missing Value Imputation:** 1.2% of samples had missing values. Forward-fill was applied for gaps shorter than 6 steps, while linear interpolation was used for longer gaps. No backward-fill was applied to prevent future data leakage.
2. **Outlier Detection:** Values outside the range $[\mu \pm 5\sigma]$ or negative power values were flagged. Approximately 0.3% of outliers were replaced using the local median (window size = 12).
3. **Standardization:** Each feature was standardized $asx' = (x - \mu)/\sigma x$ where μ and σ were computed from the training set only.

The dataset was split chronologically:

- **Training:** January–August 2018 (33,792 samples)
- **Validation:** September 2018 (4320 samples)
- **Testing:** October–December 2018 (12,428 samples)

This procedure ensures no data leakage and reflects real-world deployment scenarios.

3.2 MHA-CNN (Multi Head Attention Hybrid Convolutional Neural Network)

In the proposed hybrid classification model, the fusion of 1D CNN and Transformer mechanisms enhances the extraction of relevant temporal features from time-series data such as wind speed and power measurements for SCADA based wind power forecasting as depicted in Fig. 1. The architecture illustrated in Fig. 2 strategically uses the unique capabilities of each method to complement each other, providing a more sophisticated feature extraction pipeline. The 1D CNN section is responsible for capturing local temporal patterns and initial feature extraction.

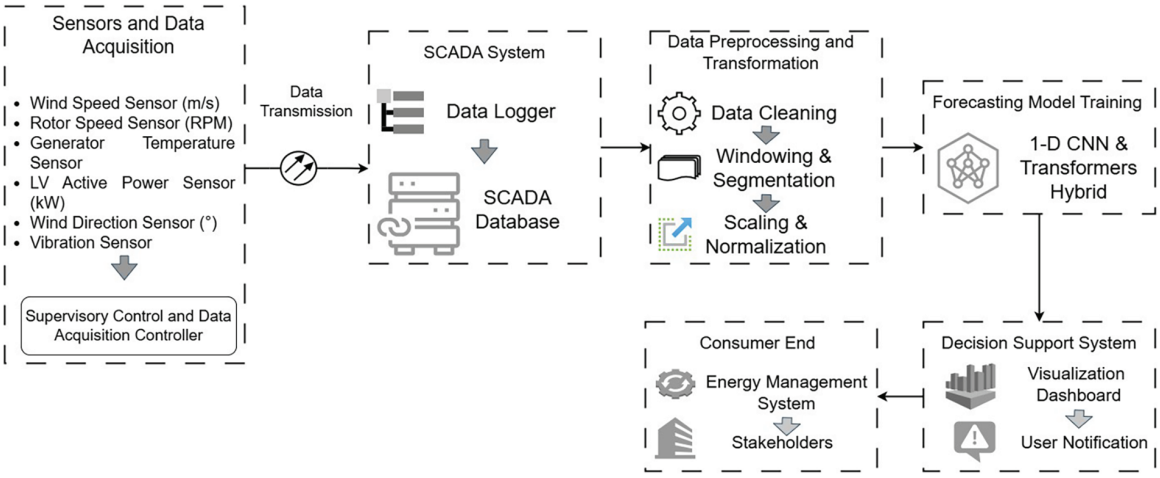

Figure 1: Flow diagram of SCADA-based wind turbine forecasting system from data acquisition to consumer end

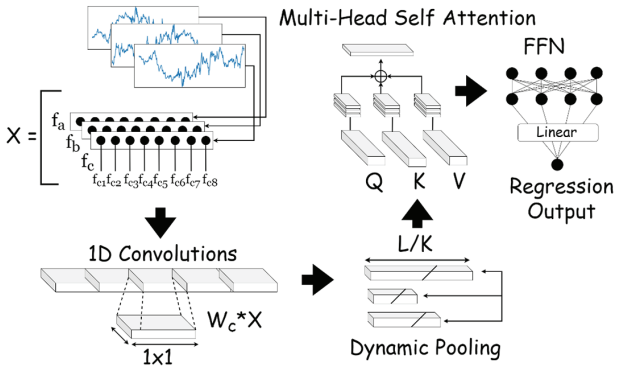

Figure 2: Diagram for the proposed multi-head attention hybrid convolutional neural network (MHA-CNN)

Given a time-series input $\boldsymbol{x} \in \mathbb{R}^{T\times F}$, where T is the sequence length (e.g., time steps) and F is the number of features (e.g., sensor readings or wind speed at different locations), a 1D convolution is applied. The convolution operation for the input sequence x with a filter $\boldsymbol{w} \in \mathbb{R}^{K\times F}$ where K is the kernel size at time step t is given by:

$$Conv\,(x)\,(t) = \sum_{i=0}^{K-1} \boldsymbol{w}_i.\boldsymbol{x}\,(t+i) + b \tag{3}$$

After the convolution operation, an activation function like Leaky ReLU is applied instead of ReLU to address the potential problem of "dead neurons" which may potentially be caused by the negative values after standardization [20]. Unlike ReLU, which outputs zero for negative values, Leaky ReLU allows a small negative slope as shown in Fig. 3, preventing information loss in negative regions:

$$Leaky\ \ ReLU\,(z) = \begin{cases} z, & if \quad z \geq 0 \\ \alpha z, & if \quad z < 0 \end{cases}$$

Where α is a small constant, typically set to 0.01. This ensures that the network can still learn from inputs where z is negative, which is especially important when the input features are standardized to have a mean of zero, as a significant portion of values may be negative. Leaky ReLU helps to preserve gradient flow and capture more nuanced patterns in the time-series data which can be observed in Fig. 4. This introduces non-linearity into the model, allowing it to learn more complex representations.

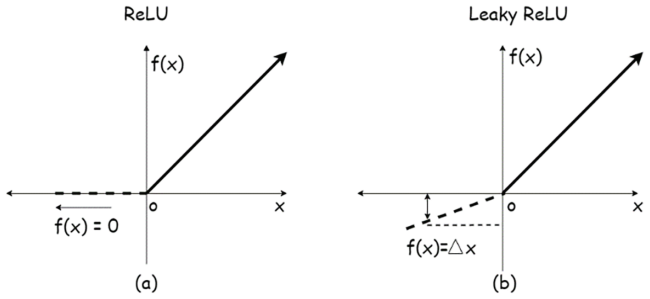

Figure 3: Illustration of activation functions (**a**) Rectified Linear Unit (ReLU), (**b**) Leaky Rectified Linear Unit (Leaky ReLU)

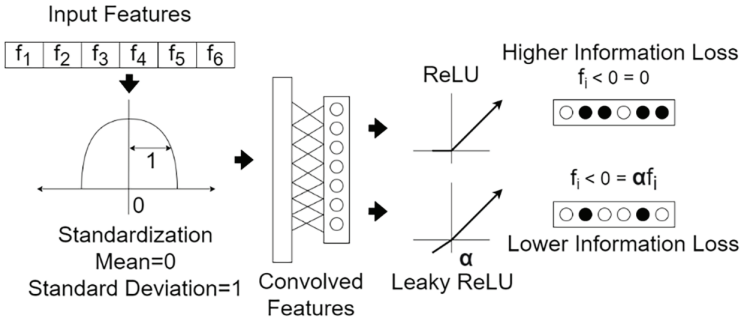

Figure 4: Illustration of feature retention capabilities of ReLU and leaky ReLU for negative inputs

Max pooling is typically applied after convolution to reduce the temporal dimension and focus on the most significant features. Max pooling works by selecting the maximum value within a static

window, say of size P, over the time steps. However, this pooling method might not prove appropriate for temporally inclined features. By selecting the maximum value in a window, crucial details are neglected and this locally aligned nature of the pooling method fails to capture the global context which is essential to understand the long term patterns.

For the given scenario, a more potent and appropriate dimensionality reduction method i.e., dynamic pooling is proposed as illustrated in Fig. 5, which is particularly useful in scenarios like time-series forecasting where preserving important temporal structures is essential [21]. Dynamic pooling works by adaptively dividing the sequence into smaller chunks and then applying pooling within each chunk, allowing the model to retain crucial information while reducing the temporal dimension more flexibly than static max pooling.

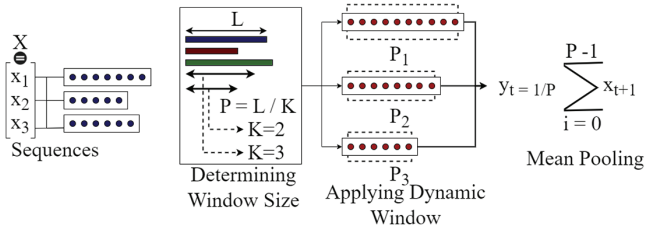

Figure 5: Illustration of proposed dynamic mean pooling

After applying the convolutional filters to the input features, the model outputs a tensor $\boldsymbol{X}_{CNN} \in \mathbb{R}^{T' \times F'}$, where T' represents the time dimension after convolution. Instead of applying max pooling, dynamic pooling was employed, which splits the temporal dimension into dynamically determined regions based on the input size and then pools over those regions. Hence, by using the dynamic pooling the model captures more complex temporal patterns, thereby reducing the risk of losing important features from longer time sequences that may span multiple regions.

The dynamic pooling operation is applied on a feature vector by dividing the respective temporal sequence X_CNN into N dynamic segments, which are computed on the basis of required reduction in temporal resolution. Each segment expressed in mathematical formulation as $\boldsymbol{S}_i \in \mathbb{R}^{\frac{\boldsymbol{T'}}{\boldsymbol{N}} \times \boldsymbol{F'}}$, is subjected to mean pooling. The pooled feature for each segment is:

$$DynPool(\boldsymbol{S}_i) = mean\left(\boldsymbol{S}_i(1), \boldsymbol{S}_i(2), \ldots, \left(\frac{\boldsymbol{T'}}{\boldsymbol{N}}\right)\right) \tag{4}$$

By applying the above pooling transformations, $\boldsymbol{N}$ pooled feature maps having size of F' have their temporal dimensions reduced from T' to $\boldsymbol{N}$; all the while ensuring that each input segment of the original sequence meaningfully contributes to the final representation.

The capability for flexible manipulation of sequences of varying lengths gives dynamic pooling an edge in dealing with time-series data. In contrast to max pooling using a rigid window, dynamic pooling can adjust to identify salient characteristics across the sequence, thereby capturing long-term dependencies more precisely. It homes in on impactful elements scattered through the sequence. This augments the model's capacity to maintain the temporal relations which are important to forecasting while down sampling lightens computational loads. Finely tuned responsiveness to sequence particulars and strategic focusing equips dynamic pooling to better serve the needs of time-series forecasting over a fixed-size window.

The dynamically downsized output $\boldsymbol{X}_{DynPool} \in \mathbb{R}^{N \times F'}$ containing significant temporal features is propagated to subsequent layers of the model for further processing.

The outputs of the CNN layers, now a reduced temporal sequence of important local features, are then fed into the Transformer blocks depicted in Fig. 6. The role of the Transformer encoder is to capture long-term dependencies in the sequence. This is essential for wind speed forecasting, where the current wind conditions may be influenced by both recent and distant past wind speeds or weather conditions. The core mechanism of the Transformer block is the self-attention mechanism, which allows each time step to attend to all other time steps in the sequence.

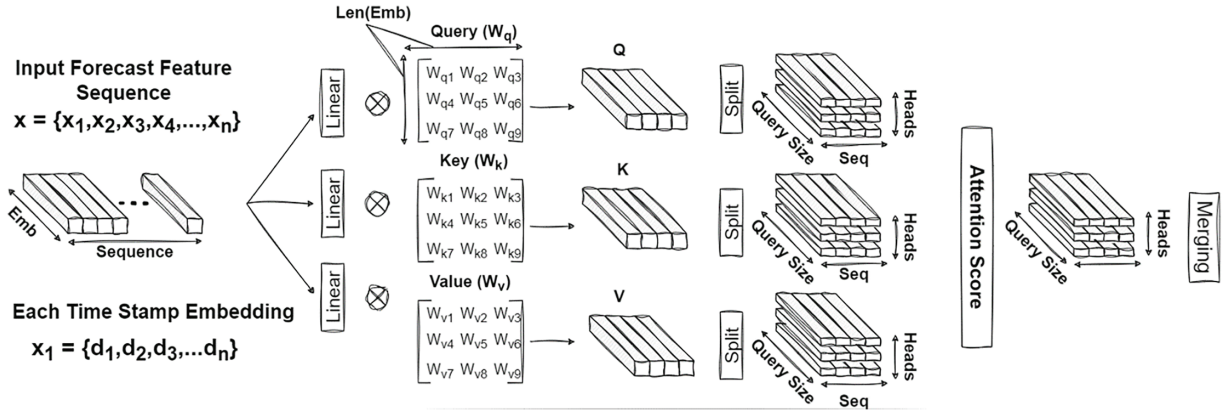

Figure 6: Block diagram of multi-head attention mechanism

Self-attention can be mathematically described as:

$$Attention\,(\boldsymbol{Q},\,\boldsymbol{K},\,\boldsymbol{V}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{T}}{\sqrt{\boldsymbol{d}_{k}}}\right)\boldsymbol{V} \tag{5}$$

Here, $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_{Q}$, $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_{K}$, and $\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_{V}$ are the query, key, and value matrices obtained by projecting the input sequence $\boldsymbol{X} \in \mathbb{R}^{T \times \boldsymbol{d}_{model}}$ where $\boldsymbol{d}_{model}$ is the model dimensionality into lower-dimensional spaces using weight matrices $\boldsymbol{W}_{Q}$, $\mathrm{W_K}$ and $\boldsymbol{W}_{V}$. The dot product of the query and key matrices, $\boldsymbol{Q}\boldsymbol{K}^{T}$ captures the similarity between each time step's representation and all others in the sequence. This similarity is scaled by $\sqrt{\boldsymbol{d}_{k}}$ (where $\boldsymbol{d}_{k}$ is the dimensionality of the key vector) to prevent overly large values in the dot product, which could cause the softmax function to saturate. The softmax function then normalizes these similarity scores into attention weights:

$$\boldsymbol{\alpha}_{i,j} = \frac{\exp(\boldsymbol{Q}_{i}\boldsymbol{K}^{T}{}_{j}/\sqrt{\boldsymbol{d}_{k}})}{\sum_{j}\exp(\boldsymbol{Q}_{i}\boldsymbol{K}^{T}{}_{j}/\sqrt{\boldsymbol{d}_{k}})} \tag{6}$$

These attention weights $\boldsymbol{\alpha}_{ij}$ determine how much focus the model places on each time step when producing the output at time step $\boldsymbol{i}$. The value matrix $\boldsymbol{V}$ is weighted by these attention scores, and the resulting weighted sum is the output of the attention mechanism.

In the context of wind speed forecasting, the attention mechanism allows the model to dynamically weigh how past time steps affect the current prediction. For example, if there is a long-term seasonal pattern in the wind speed, the model can assign higher attention to distant past time steps that exhibit similar conditions. Simultaneously, more recent fluctuations in wind speed might receive higher attention if they are more relevant to the current prediction. By integrating the Multi-head attention,

the model's attention capabilities are amplified in tending towards different aspects of the sequence simultaneously as shown in Fig. 6.

The inputs from each preceding operation are split into multiple subspaces, with attention heads depicted as h while the attention mechanism is applied to each subspace independently which is mathematically expressed as:

$$Multihead\,(\boldsymbol{Q},\ \boldsymbol{K},\ \boldsymbol{V}) = Head_{concat}\,(\boldsymbol{h_1}, \ldots, \boldsymbol{h_n}) * \boldsymbol{W_o} \tag{7}$$

where each $\boldsymbol{h_i}$ computes:

$$\boldsymbol{h_i} = Attention(\boldsymbol{QW_{Q_i}},\ \boldsymbol{KW_{K_i}},\ \boldsymbol{VW_{V_i}}) \tag{8}$$

From each head, the outputs are concatenated and then linearly transformed by using a final weight matrix $\boldsymbol{W_o}$. This concatenation operation gives model the leverage to focus on different patterns within the same input simultaneously. This is especially useful in time-series data where multiple dependencies i.e., long and short term may coexist.

In the post attention calculation, each output time step is passed to a feedforward neural network (FFN) independently as:

$$\boldsymbol{FFN}\,(\boldsymbol{x}) = \boldsymbol{ReLU}\,(\boldsymbol{xW_1} + \boldsymbol{b_1})\,\boldsymbol{W_2} + \boldsymbol{b_2} \tag{9}$$

This two-layered network introduces additional non-linear components and permits a richer alteration of the attended characteristics. By applying the feedforward network to each discrete time period autonomously, the design is able to enhance the learnt feature representations designated by the attention system. While ensuring that even when focusing on other time intervals, the features can be further processed to capture non-linear interrelationships.

While normalization is practiced ahead of and subsequent to both the attention mechanism and the feedforward system, the inputs to these layers are standardized, which stabilizes the learning and allows for a more improved gradient flow. Simultaneously, the deeper architecture improves the capability to identify intricate connections among time steps by provisioning further levels through which inputs are transformed preceding make predictions or selections, thereby advancing the representation capability of the model. The layer normalization process can be mathematically given as:

$$\boldsymbol{LN}\,(\boldsymbol{x}) = \frac{\boldsymbol{x} - \boldsymbol{\mu}}{\boldsymbol{\sigma} + \boldsymbol{\varepsilon}} \tag{10}$$

The mean of the features is depicted with μ and σ represents the variance of the input vectors. In order to prevent the division by zero a small constant ϵ is added. The use of the Layer normalization helps in mitigating the issues of the internal covariate shift, greatly improving the model's ability to converge during training. The pseudo algorithm for the proposed methodology is given in Algorithm 1, while the model's parametric configuration and hyper-parameter settings are provided in Tables 2 and 3, respectively.

Algorithm 1 : Pseudocode for the proposed MHA-CNN framework

INPUT DATA
  $D \in \mathbb{R}^{n \times d}$, where n is the number of time steps, and d is the feature dimension.
NETWORK INITIALIZATION
  $C \leftarrow$Initialize 1D CNN with kernel size K and filters F
  $T \leftarrow$Initialize Transformer encoder with layers L, heads H, and embedding size D
  $FC \leftarrow$Initialize fully connected layer for regression output
OBJECTIVE FUNCTION
  $L_{reg} \leftarrow$ Minimize Mean Squared Error (MSE)
TRAINING LOOP
  For each epoch $e \in [1, E]$:
    Feature Extraction with 1D CNN:
      $Z_{conv} \leftarrow C(X; W_c, b_c)$
      $A_{conv} = LeakyReLU(Z_{conv}, \alpha)$
    Feature Transformation with Transformer Encoder:
      For each Transformer layer $l \in [1, L]$
        $Z_{attl} = MultiHeadAttention(A_{convl}, H)$
        $A_{attl} = LayerNorm(A_{convl} + Z_{attl})$
        $Z_{ffnl} = FFN(A_{attl})$
        $A_{encl} = LayerNorm(A_{attl} + Z_{ffnl})$
    Flatten and Regression Output:
      $A_{flat} = Flatten(A_{encl})$
      $\hat{y} = FC(A_{flat}; W_{fc}, b_{fc})$
    Loss Computation and Backpropagation:
      $L_{reg} = \frac{1}{B} \sum_{i=1}^{B} (y_i - \widehat{y_i})^2$
      $\theta \leftarrow \theta - \eta \nabla_\theta L_{reg}$
OUTPUT
  Predicted future values $\hat{y}$ for each time series sequence.

Table 2: Parametric Configuration of Proposed MHA-CNN architecture

TRANSFORMER—BLOCK				
Transformer Feature Extraction	No. of layers	Parameters		Input shape
Multi-Head Self-Attention (MHA)	1	4 heads, 512 dim each		(W, F, d)
Feed-Forward Network (FFN)	2	512 → 2048 (linear)		(W, F, 512)
Layer Normalization	2	–		(W, F, 512)
Residual Connections	2	–		(W, F, 512)
CONVOLUTION BLOCK				
1D CNN Feature Extraction	Kernel size	Filters	Activation	Input shape
1D Convolution	3	64	Leaky ReLU ($\alpha = 0.01$)	(W, F, d)
1D Convolution	3	128	Leaky ReLU ($\alpha = 0.01$)	(W, F, 64)
Dynamic pooling	Variable	–	–	(W, F, 128)

(Continued)

Table 2 (continued)

REGRESSION BLOCK			
Flatten & Fully connected layer	Units	Activation	Input Shape
Flatten layer	–	–	(W, F, 128)
Fully connected (Dense)	256	Leaky ReLU ($\alpha = 0.01$)	(W × F × 128)
Fully connected (Dense)	1	Linear	–
COMPUTATIONAL COST			
Number of parameters	–	–	187,392
Inference time	–	–	48 ms

Table 3: Hyper-parameter configuration for training

Hyper-parameter	Value
Epochs	100
Loss	MAE (Mean Absolute Error)
Optimizer	Adam
Learning rate	0.001
Batch size	64
Dropout rate	0.2
Activation function	ReLU
Validation split	0.2
Early stopping	Enabled
Patience	10

3.3 Distinction from FEDformer

While FEDformer [22] employs Fourier-based frequency selection to reduce attention complexity, it operates on full sequences and requires decomposition as a pre-processing step. In contrast, MHA-CNN introduces several key differences:

- It applies a 1D-CNN front-end to extract local features *before* the attention mechanism, effectively reducing the input length from T to T', where $T' \ll T$.
- It uses dynamic mean pooling to adaptively downsample the sequence based on T', unlike FEDformer's fixed frequency-band approach.
- It employs a lightweight multi-head attention (MHA) module with 4 heads and a model dimension ($d_{mo}d_{el}$) of 64, compared to FEDformer's deeper and heavier attention blocks.
- It achieves over 10× faster inference (48 vs. 620 ms on the same CPU) due to the reduced sequence length and computational load.

This local-to-global hierarchical design enables real-time deployment on edge devices, providing a significant advantage over frequency-domain transformer architectures.

3.4 Evaluation Metrics

3.4.1 Mean Squared Error (MSE)

Measures the average of the squared differences between the actual and predicted values. It heavily penalizes larger errors.

$$\boldsymbol{MSE} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{i=1}^{n} (\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i)^2 \tag{11}$$

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

3.4.2 Root Mean Squared Error (RMSE)

The square root of MSE, providing error in the same units as the output variable. It also penalizes larger errors more than smaller ones.

$$\boldsymbol{RMSE} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{n}} \sum_{i=1}^{n} (\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i)^2} \tag{12}$$

3.4.3 R-Squared (R^2)

Represents the proportion of the variance in the dependent variable that is predictable from the independent variables. It shows the model's goodness-of-fit.

$$\boldsymbol{R^2} = \mathbf{1} - \frac{\sum_{i=1}^{n} (\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i)^2}{\sum_{i=1}^{n} (\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i)^2} \tag{13}$$

3.4.4 Mean Absolute Error (MAE)

Measures the average of the absolute differences between actual and predicted values. It is less sensitive to outliers than MSE.

$$\boldsymbol{MAE} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{i=1}^{n} |\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i| \tag{14}$$

3.4.5 Symmetric Mean Absolute Percentage Error (sMAPE)

Expresses prediction accuracy as a percentage by comparing the absolute error to actual values. It is scale-independent but can be distorted by small actual values.

$$\boldsymbol{MAPE} = \frac{\mathbf{100}}{\boldsymbol{n}} \sum_{i=1}^{n} \left| \frac{\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i}{\boldsymbol{y}_i} \right| \tag{15}$$

To avoid instability when actual values approach zero, we use the symmetric mean absolute percentage error ($sMAPE$), defined as:

$$\boldsymbol{sMAPE} = \frac{200}{n} \times \Sigma \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i| + \varepsilon)} \tag{16}$$

where $\varepsilon = 10^{-6}$.

4 Results and Discussion

4.1 Dataset Description

The 2018 wind power SCADA dataset is a publicly available resource that offers a comprehensive overview of wind turbine performance throughout the year 2018. Collected from a single wind turbine in Turkey, the dataset provides granular insights into the turbine's operational characteristics. The dataset covers the entire year of 2018, providing a complete annual record. Data is recorded at 10-min intervals, offering a detailed view of operational fluctuations. Key parameters included in the dataset are date/time, LV active power, wind speed, theoretical power curve, and wind direction.

4.2 Feature Analysis

The Fig. 7 illustrates the distribution of Low Voltage (LV) Active Power consumption and the wind speed. The histogram in Fig. 7a represents the frequency of different LV Active Power values, with a clear peak around the 0–200 kW range, indicating that this is the most common range of consumption. The skewed shape of the histogram suggests that there are fewer data points in the higher range, indicating that high-consumption outliers are relatively rare.

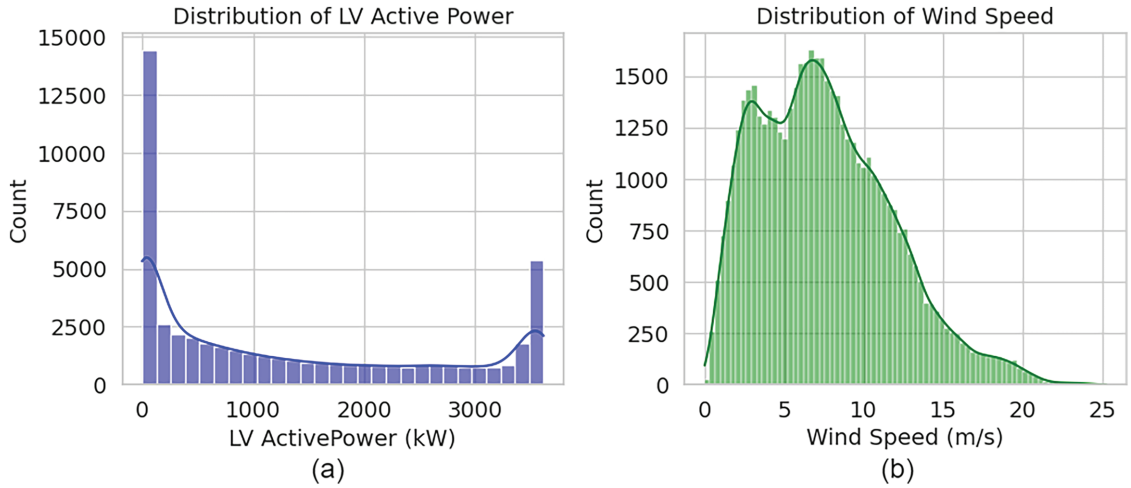

Figure 7: Visualization of feature distribution (**a**) Histogram of active power in KW (**b**) Density plot for wind speed in m/s

The density curve in Fig. 7b provides a smooth representation of the distribution, confirming the skewed pattern observed in the histogram. It also highlights the peak around the 3–7 m/s wind speed, further emphasizing the most common range of LV Active Power consumption. The extended right tail of the density curve suggests that the probability of observing higher LV Active Power consumption decreases gradually. Similarly, the wind rose illustrated in Fig. 8 provides valuable insights into the local wind conditions. The dominant directions of north and east suggest that prevailing winds in the area come from these directions. This information is crucial for understanding the local climate and weather patterns.

Additionally, the wind rose can be used to assess the potential for wind energy generation. Regions with consistent and strong winds from certain directions are more suitable for wind turbines.

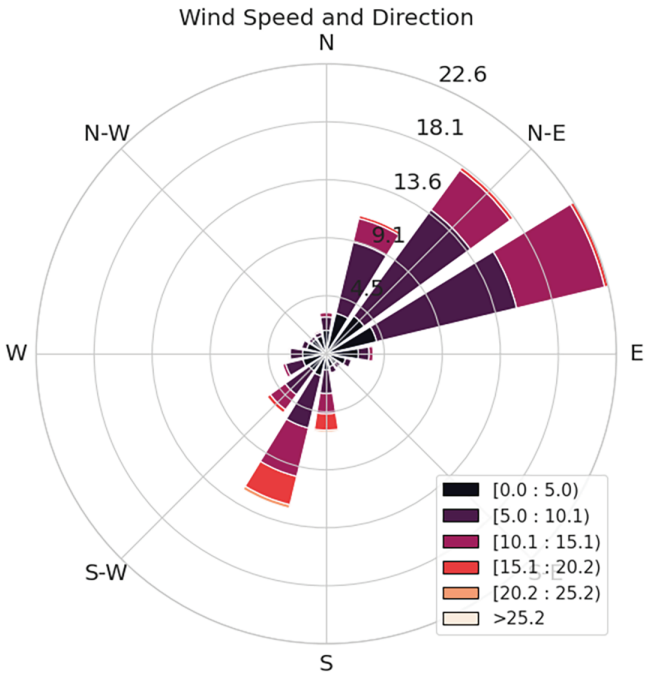

Figure 8: Wind rose plot depicting the wind speed distribution against the direction

4.3 Evaluation for Hourly Forecast

The Fig. 9 represents a short-term power forecasting scenario, where the actual and predicted power values are compared across several hours and evaluated in terms of the residuals. As illustrated in Fig. 9a, the model demonstrates strong alignment between the actual data and its predictions, particularly in the early stages, suggesting that it effectively captures overall trends in power demand. However, in some areas, especially during peak demand periods, slight deviations are visible, most notably around the 120 to 150-h mark. The model marginally falls short to capture the rapid fluctuations in power usage, as seen in these high-spike regions. This indicates that while the model is well-suited for capturing gradual changes in power consumption, its performance dwindles slightly during periods of more abrupt changes, particularly at higher peaks.

The residuals plot in Fig. 9b identifies the difference between the predicted and actual values against the predicted values themselves. Here, the residuals are mostly centered around zero, particularly for mid-range predictions, demonstrating that the model generally performs well for moderate power demand. However, as the predicted values increase, the residuals scatter more widely, showing that the model shows a bit of struggle with the higher power values, often overestimating them. This becomes evident as several points deviate significantly from zero in the higher predicted ranges, indicating potential over-prediction for high-power demand scenarios. The pattern observed in the residuals suggests that while the model has overall consistency, its ability to predict extreme power demand cases can be further refined, particularly to minimize overestimations in these situations.

The histogram presented in Fig. 10 illustrates the distribution of errors between actual and predicted values for an hourly wind turbine power generation forecasting model. On the x-axis, prediction errors are displayed, representing the difference between the predicted and actual values, while the y-axis indicates the frequency of occurrences for each error. The overall shape of the

histogram provides key insights into the model's performance by visualizing how often specific error magnitudes occur, giving an immediate sense of accuracy and consistency. The distribution of errors is approximately normal, with a concentration around zero, indicating that the majority of the model's predictions are reasonably accurate. However, there is a minimal skew towards the negative side, which suggests that the model has a slight tendency to underestimate power generation slightly more frequently than it overestimates. Furthermore, a few outliers are evident, particularly in the negative range, highlighting occasional instances where the model significantly under-predicts the actual values. Despite this, the overall spread of the error distribution remains narrow, indicating that the model's errors are generally consistent in size, rather than fluctuating wildly.

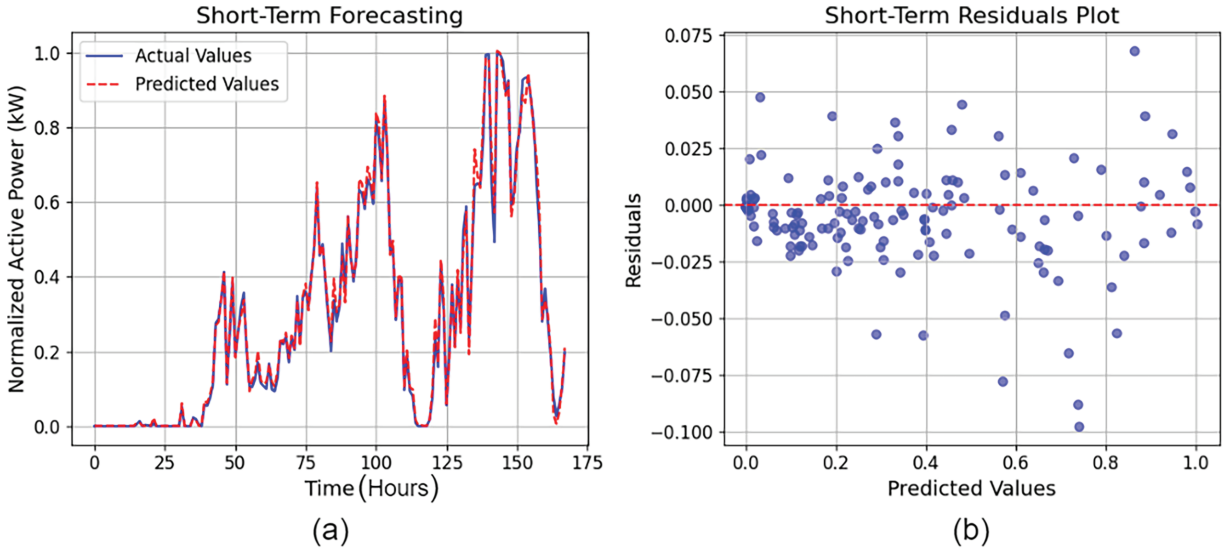

Figure 9: Short term prediction results of proposed MHA-CNN. (**a**) Hourly Forecast comparison between actual and predicted values; (**b**) Residual plot plotted against predicted values

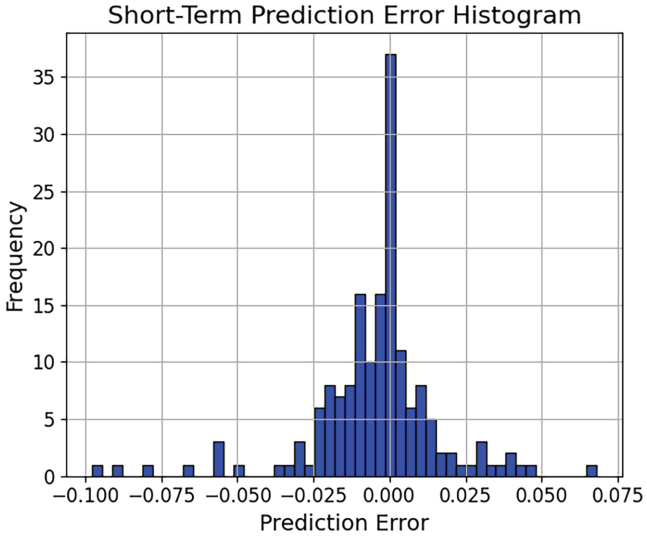

Figure 10: Prediction error histogram for hourly forecast

The Fig. 11 shows the model's performance across multiple short-term forecasting horizons, with actual and predicted values compared over different time intervals. The model performs well for the shorter horizons, as seen in Horizon 1, where actual and predicted values are closely aligned, showcasing its effectiveness in immediate-term forecasting. However, as the horizon extends, the divergence between actual and predicted values increases to some extent. In Horizon 3, the predicted values remain mostly consistent with actual values, highlighting the model's adeptness for longer-term predictions. This suggests that while the model is highly effective in short-term forecasting and bodes well even when the forecasting horizon extends.

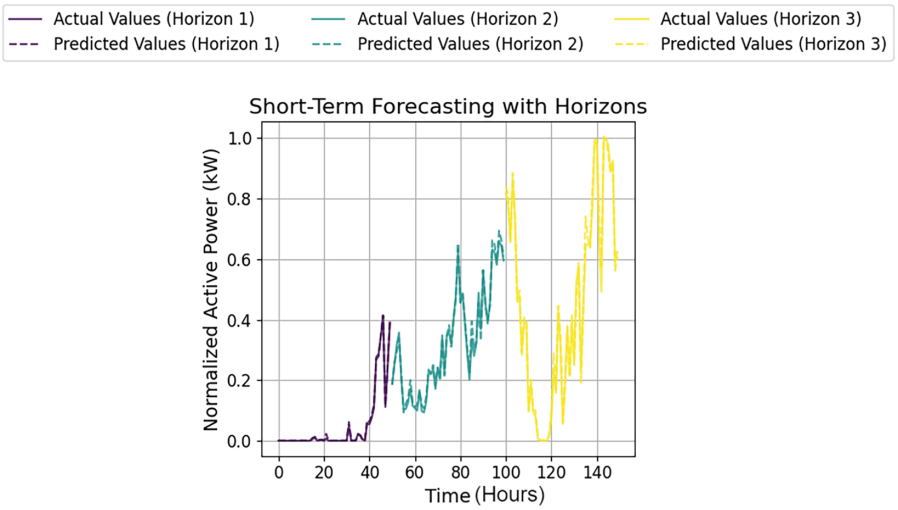

Figure 11: Hourly forecast comparison of actual and predicted values for different horizons

4.3.1 Evaluation for Daily Forecast

Similar to the hourly forecast, the proposed model's capabilities in long term forecasting were evaluated. The results displayed in Fig. 12a represents a long-term forecasting model for wind turbine power generation, where the x-axis denotes days and the y-axis reflects normalized active power in kilowatts (kW). The actual values are illustrated by the blue line, while the red line indicates the predicted values. The model effectively demonstrated an understanding of general wind power trends, indicating competence in grasping broader fluctuations over time. However, some discrepancies between actual and predicted values emerged, particularly during intervals of swift transitions, suggesting it is somewhat reactive to abrupt variations or environmental shifts nominally impacting generation. Overall, it correctly distinguished periods of elevated output which likely reflect favorable conditions while demonstrating decent ability to capture repeating tendencies across longer stretches. Nevertheless, accuracy faltered minimally when faced with rapid fluctuations, presumably caused by sudden wind speed or direction changes. These errors revealed that while tracking overarching patterns capably, responding to short-term, more drastic alterations leaves room for enhancement.

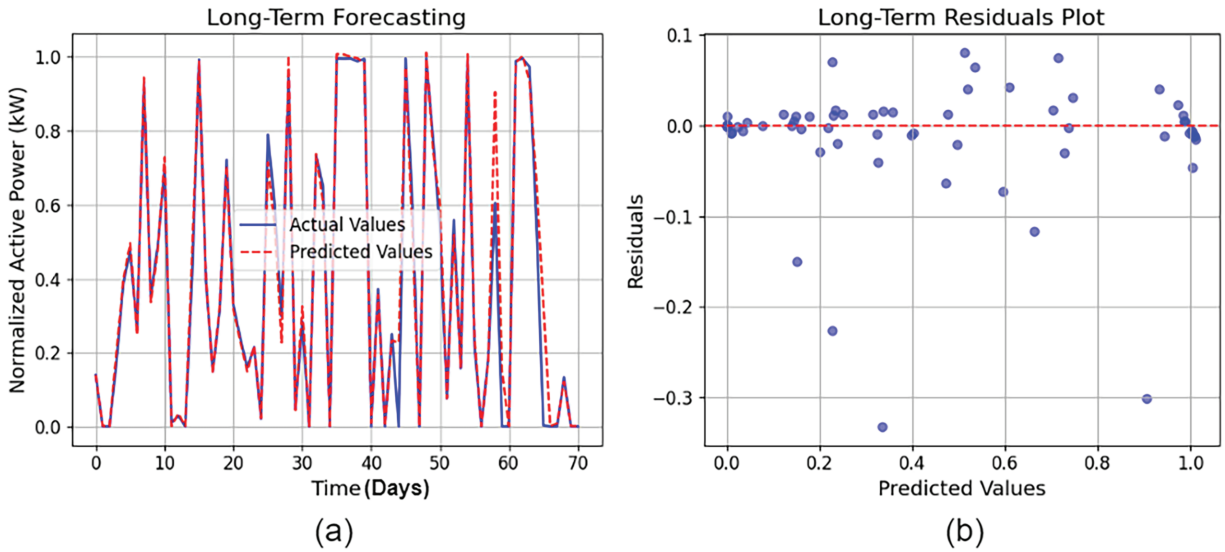

Figure 12: Short term prediction results of proposed MHA-CNN. (**a**) Hourly forecast comparison between actual and predicted values; (**b**) Residual plot plotted against predicted values

There are also notable periods when the model significantly underestimates or overestimates power creation. During these instances, the differences between real and anticipated values could lead to ineffectiveness, such as missed opportunities for vitalizing the energy grid optimization or fallacies in provision predictions. The underperformance may emerge from aspects not adequately depicted in the data, like fluctuations in temperature, moisture, or atmospheric stress, all that can impact wind behavior. These deviations highlight the complex and dynamic essence of wind power forecasting, emphasizing the difficulties of achieving high precision, particularly for short-term forecasts within a long-term model setting. The deviations occur as the climate and atmospheric conditions change rapidly in ways not fully captured by the model. Additionally, minor variations in wind turbine equipment or surrounding landscape features may affect wind patterns in unforeseen ways. Despite the high overall performance, there are minor challenges associated with the accurately predicting generation from abrupt fluctuations.

The residuals plot within Fig. 12b exhibits a seemingly random scattering of points evenly straddling the flat line at zero, suggesting the model did not systematically over or under estimate the target variable. No clear pattern or trends surfaced among the residuals, implying the model didn't fail to miss any significant relationships within the data. While the residuals look to maintain a steady variance across the anticipated values, signaling the model's errors as homoscedastic, a scattering of outliers can be detected within the plot, proposing the model incorrectly predicting a minute proportion of data points.

Similarly, the histogram in Fig. 13 shows a roughly normal distribution of long-term prediction errors, centered around zero with a slight negative skew. While most errors are relatively small, there are a few outliers, particularly on the negative side. This suggests that the model is generally performing well, but there is room for improvement in terms of reducing outliers and the skew towards the negative side.

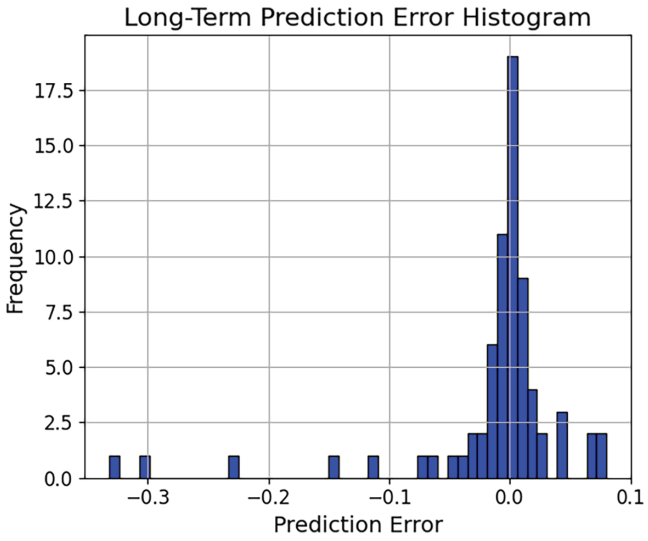

Figure 13: Prediction error histogram for daily forecast

The results in Fig. 14 illustrates a day-ahead forecasting model for wind turbine power generation across multiple prediction horizons, where the x-axis represents time and the y-axis shows normalized active power in kilowatts (kW). The model provides predictions for three distinct horizons, each corresponding to slightly longer time periods into the future. For each horizon, the solid lines depict the actual power values, while the dashed lines represent the predicted values. This allows for a clear comparison between the model's accuracy at varying time intervals, highlighting its ability to capture certain trends while missing others.

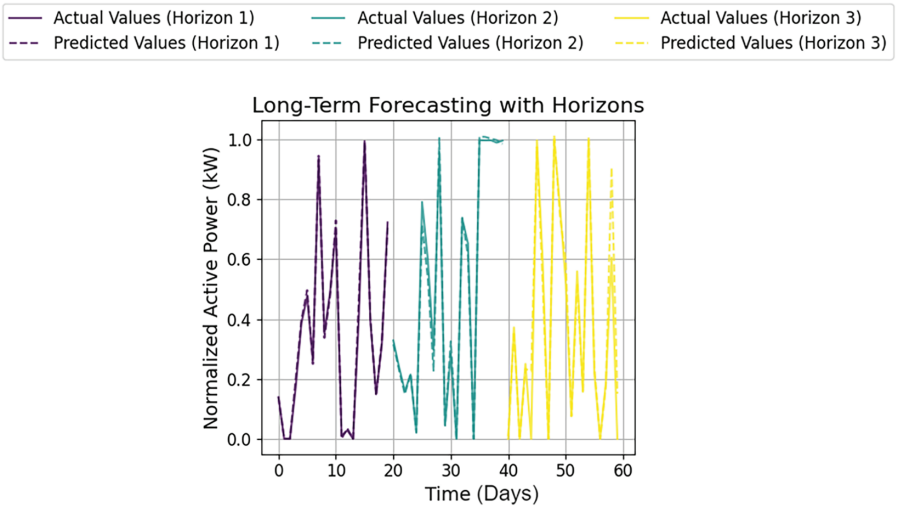

Figure 14: Daily forecast comparison of actual and predicted values for different horizons

Across the three horizons, the model's accuracy diminishes slightly as the prediction period extends. Horizon 1, representing the shortest forecast, shows a high degree of alignment between actual and predicted values, indicating the model's reliability in the immediate future. However, as we move to Horizon 2 and Horizon 3, the discrepancies between actual and predicted values become a bit

more pronounced then the former, particularly for Horizon 3. The model's predictions start to deviate more from the actual values especially towards the higher power consumption intervals, indicating a marginal discrepancy in capturing shorter-term fluctuations and more granular changes in wind power generation. This suggests that while the model can maintain a reasonable level of accuracy over shorter horizons, its ability to predict accurately remains overall consistent even with longer time frames.

The minor prediction errors may be attributed to the unpredictable nature of wind patterns, which are difficult to forecast accurately over longer periods. As a result, the model's long-term trends remain useful, but for more detailed predictions, it shows minute discrepancies. This highlights the inherent challenge in balancing the accuracy of short-term forecasting against the necessity to predict further into the future, indicating that additional model refinements strategies.

4.3.2 Evaluation for Each Month

The evaluation results in Table 4 and Fig. 15 for month-ahead forecast shows that the model performs most effectively during Months 4, 8, and 9, where the R^2 values exceed 0.99, and the error metrics (MSE, RMSE, MAE) are at their lowest, indicating highly accurate predictions. However, the model's performance is weaker in Months 1 and 12, where the R^2 values drop to around 0.60 to 0.84, and the errors significantly increase, suggesting difficulties in maintaining accuracy during these months. Although the overall errors (RMSE, MAE) are relatively low for most months.

Table 4: Evaluation results of proposed MHA-CNN model for monthly-forecast

Month	MSE	RMSE	R^2	MAE	sMAPE
1	0.06953	0.263688	0.6029	0.124846	41.2%
2	0.02065	0.143714	0.8596	0.060590	28.7%
3	0.00969	0.098415	0.9408	0.033067	19.4%
4	0.00169	0.041133	0.9823	0.014680	12.8%
5	0.00103	0.032038	0.9860	0.016176	10.1%
6	0.00105	0.032455	0.9878	0.015021	9.9%
7	0.00396	0.062904	0.9138	0.014063	11.5%
8	0.00075	0.027408	0.9928	0.018502	8.7%
9	0.00106	0.032567	0.9922	0.018182	10.3%
10	0.00121	0.034829	0.9881	0.022376	11.9%
11	0.01566	0.125154	0.8947	0.048566	22.6%
12	0.02025	0.142302	0.8408	0.069388	31.4%

The lower R^2 values in months 1 and 12 likely indicate reduced wind activity and more frequent calm periods during winter, which increase the difficulty of forecasting.

In the short-term forecasting plot illustrated in Fig. 16a, most models closely track the actual power values, with the proposed MHA-CNN delivering the most consistent and accurate performance. The dashed red line representing the MHA-CNN model follows the actual values with minimal deviation, capturing both sharp transitions and smoother trends in the data. Other models such as 1D-CNN, LSTM, and XG-Boost also perform reasonably well as shown in Table 5, with the CNN-based models handling the temporal dynamics effectively. However, models like Random Forest and Decision Tree exhibit more fluctuation, particularly during periods of rapid change, which suggests

that their reliance on simpler decision boundaries leads to an inability to capture short-term variations as accurately as the deep learning models. Notably, Decision Tree shows significant errors during peak periods, underscoring its struggles with dynamic, high-frequency variations.

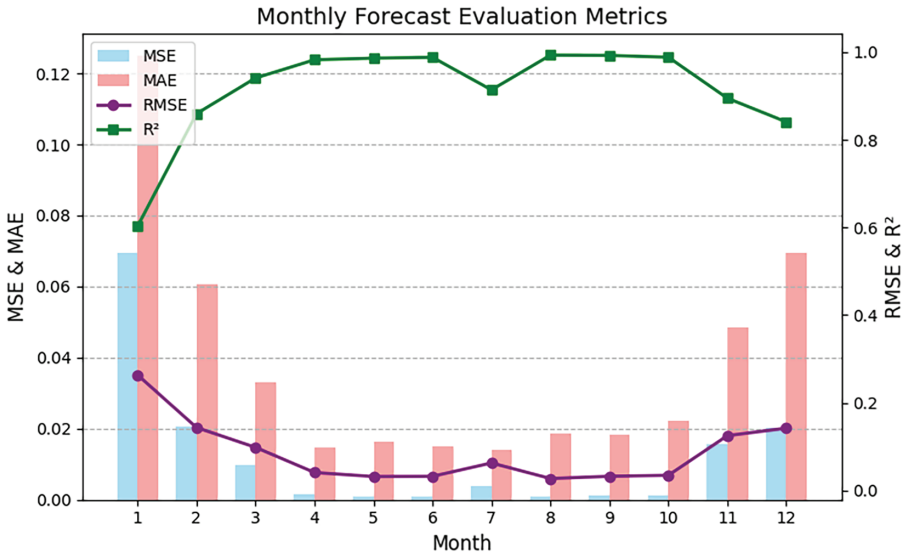

Figure 15: Hybrid bar—line plot for monthly forecast evaluation

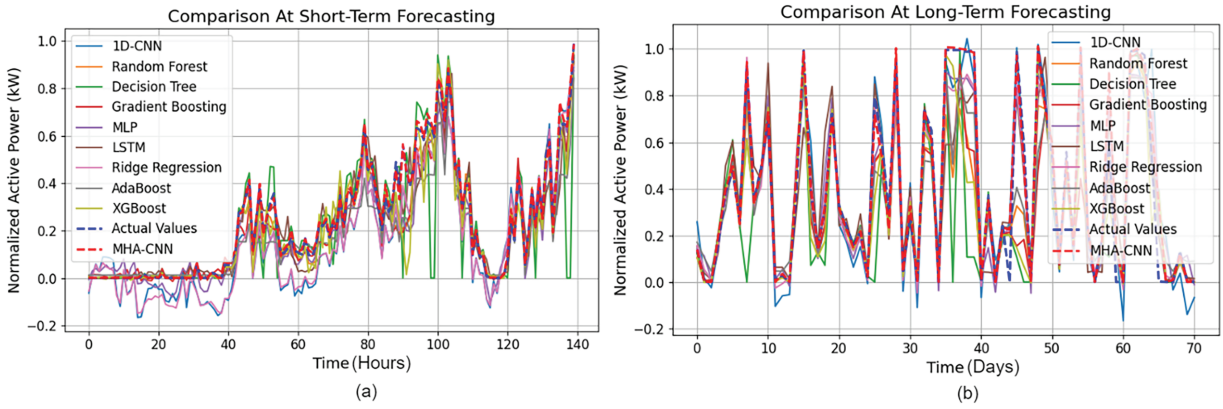

Figure 16: Hour and Day-Ahead forecast comparison of proposed model and other ML and DL models (**a**) Hour-ahead forecast comparison (**b**) Day ahead forecast comparison

Table 5: Comparison of proposed MHA-CNN model with other ML and DL models for hour-ahead prediction across multiple evaluation criterion

Model	MSE	RMSE	R^2	MAE	sMAPE
1D-CNN	0.0120	0.1097	0.8563	0.0978	42.1%
Random forest	0.0062	0.0790	0.9256	0.0449	21.63%
Decision tree	0.0252	0.1588	0.6993	0.0760	41.09%

(Continued)

Table 5 (continued)

Model	MSE	RMSE	R^2	MAE	sMAPE
Gradient boosting	0.0102	0.1012	0.8779	0.0586	38.73%
MLP	0.0084	0.0917	0.8997	0.0635	40.83%
LSTM	0.0080	0.0896	0.9043	0.0635	35.26%
Ridge regression	0.0111	0.1055	0.8672	0.0935	42.35%
AdaBoost	0.0135	0.1161	0.8392	0.0821	37.91%
XGBoost	0.0096	0.0982	0.8849	0.0572	40.13%
MHA-CNN	0.0005	0.0217	0.9944	0.0138	14.48%

The day-ahead forecasting results depicted in Fig. 16b and tabularized in Table 6 show the MHA-CNN design outperforming others by capturing more extensive temporal dependencies and maintaining alignment with real values, even amidst period of rapid fluctuations. The MHA mechanism likely aids in attending to time advances crosswise over the larger horizons, diminishing alignment inconsistencies seen in other models. While 1D-CNN and LSTM models still perform relatively well, they show extended deviations, particularly amid times with bigger spikes and troughs, proposing a slight decay in their capacity to generalize over longer horizons. Ensemble strategies like XG-Boost and Gradient Boosting perform well enough throughout over less complex models like Decision Tree and Random Forest, yet their expectations show more abnormal conduct, particularly amid the tops, demonstrating troubles precisely catching far off patterns. Conclusively, the MHA-CNN shows remarkable forecasts in both hour and day-ahead predictions because of its hybrid structure, while machine learning models confront more articulated difficulties in keeping up prescient accuracy over extended time periods.

Table 6: Comparison of proposed MHA-CNN Model with other ML and DL models for day-ahead prediction across multiple evaluation criterion

Model	MSE	RMSE	R^2	MAE	MAPE
1D-CNN	0.0135	0.1162	0.9003	0.0894	44.81%
Random forest	0.0279	0.1670	0.7941	0.0913	39.65%
Decision tree	0.0830	0.2882	0.3866	0.1283	39.46%
Gradient boosting	0.0344	0.1855	0.7458	0.0951	40.21%
MLP	0.0113	0.1063	0.9165	0.0802	43.25%
LSTM	0.0212	0.1455	0.8436	0.1052	42.98%
Ridge regression	0.0090	0.0951	0.9332	0.0672	41.74%
AdaBoost	0.0225	0.1501	0.8337	0.0993	44.22%
XGBoost	0.0369	0.1922	0.7272	0.0981	39.98%
MHA-CNN	0.0047	0.0686	0.9652	0.0317	35.46%

4.4 Ablation Study

The ablation study given in Table 7 reveals that the integration of Multi-Head Attention (MHA) and 1D-CNN provides the best performance, achieving an average R^2 score of 0.9129, demonstrating the synergy between these two components in enhancing model accuracy as seen in Table 6. When only 1D-CNN is employed, the score drops to 0.8556, whereas using only MHA leads to a modest improvement with an R^2 score of 0.8715, indicating that MHA contributes more individually. Among pooling strategies, Dynamic Mean Pooling delivers a superior R^2 score of 0.9079 compared to Max Pooling at 0.8848, highlighting its effectiveness. In the comparison of activation functions, Leaky ReLU performs better than ReLU, with an R^2 score of 0.8819 vs. 0.8576.

Table 7: Ablation study results (R^2 values averaged across hour-ahead, day-ahead, and month-ahead horizons)

MHA-CNN		
Multi-Head Attention	1D-CNN	Avg. R^2 Score
✖	✓	0.8556
✓	✖	0.8715
✓	✓	0.9129
POOLING OPERATIONS		
Max Pooling	Dynamic mean pooling	Avg. R^2 Score
✖	✓	0.9079
✓	✖	0.8848
ACTIVATIONS		
ReLU	Leaky ReLU	Avg. R^2 Score
✖	✓	0.8819
✓	✖	0.8576

5 Conclusion

This study introduces MHA-CNN, a hybrid architecture that combines 1D-CNNs for local pattern extraction, multi-head attention for modeling long-range dependencies, and dynamic mean pooling for adaptive downsampling. The model achieves state-of-the-art performance on single-turbine SCADA data, with $R^2 = 99.42\%$ for hour-ahead forecasts and 96.52% for day-ahead forecasts, while maintaining inference latency under 50 ms on a CPU.

Limitations of the current work include validation on a single turbine, which limits spatial generalizability, the lack of probabilistic forecasts, and sensitivity to extreme calm periods, although this last issue is partially mitigated using sMAPE. Future work will focus on extending the model to multi-turbine and multi-site validation using NREL datasets, enabling probabilistic forecasting

via quantile MHA-CNN, integrating physics-informed constraints such as power curve bounds, and deploying the model on edge devices (e.g., Raspberry Pi) to support real-time grid operations.

The proposed framework provides a scalable foundation for efficient, accurate, and deployable wind power forecasting systems.

Acknowledgement: Not Applicable.

Funding Statement: The research team acknowledges the Deanship of Graduate Studies and Scientific Research at Najran University for supporting the research project through the Nama'a program, with Project No NU/GP/SERC/13/401-3. The APC of the paper is funded by the School of Engineering, Cardiff University, Cardiff CF24 3AA, UK.

Author Contributions: The authors confirm contribution to the paper as follows: Saifur Rahman, Abdullah Shaher, Hatim Alwadie, Muhammad Irfan, Saleh Al dawsari and Ayman Taher Hindi have performed analysis, conceptualization, investigation, visualization, project management, resources, editing and review. Nabeel Ahmed Khan, Muhammad Abubakar, Zohaib Mushtaq have performed methodology, software, validation, formal analysis, writing—review and editing. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The online public available dataset has been used in this study as given in the link https://www.kaggle.com/datasets/berkerisen/wind-turbine-scada-dataset (accessed on 31 October 2025).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Dhiman HS, Deb D. A review of wind speed and wind power forecasting techniques. arXiv: 2009.02279. 2020. Available from: https://arxiv.org/abs/2009.02279.
2. Tsai WC, Hong CM, Tu CS, Lin WM, Chen CH. A review of modern wind power generation forecasting technologies. Sustainability. 2023;15(14):10757. doi:10.3390/su151410757.
3. 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. doi:10.1016/j.apenergy.2021.117766.
4. Qian Z, Wen S, Zhang L, Zhang J, Yuan S, Mao L, et al. Deep learning-based short-term wind power prediction considering various factors. In: Proceedings of the 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV); 2022 Dec 11–13; Singapore. p. 529–33. doi:10.1109/icarcv57592.2022.10004261.
5. Yaghoubirad M, Azizi N, Farajollahi M, Ahmadi A. Deep learning-based multistep ahead wind speed and power generation forecasting using direct method. Energy Convers Manage. 2023;281:116760. doi:10.1016/j.enconman.2023.116760.
6. Tuerxun W, Xu C, Guo H, Guo L, Zeng N, Cheng Z. An ultra-short-term wind speed prediction model using LSTM based on modified tuna swarm optimization and successive variational mode decomposition. Energy Sci Eng. 2022;10(8):3001–22. doi:10.1002/ese3.1183.
7. Oh J, Park J, Ok C, Ha C, Jun HB. A study on the wind power forecasting model using transfer learning approach. Electronics. 2022;11(24):4125. doi:10.3390/electronics11244125.

8. Hossain MA, Chakrabortty RK, Elsawah S, Ryan MJ. Very short-term forecasting of wind power generation using hybrid deep learning model. J Clean Prod. 2021;296:126564. doi:10.1016/j.jclepro.2021.126564.
9. An G, Jiang Z, Cao X, Liang Y, Zhao Y, Li Z, et al. Short-term wind power prediction based on particle swarm optimization-extreme learning machine model combined with adaboost algorithm. IEEE Access. 2021;9:94040–52. doi:10.1109/access.2021.3093646.
10. Abbasipour M, Igder MA, Liang X. A novel hybrid neural network-based day-ahead wind speed forecasting technique. IEEE Access. 2021;9:151142–54. doi:10.1109/access.2021.3126747.
11. Meng A, Chen S, Ou Z, Ding W, Zhou H, Fan J, et al. A hybrid deep learning architecture for wind power prediction based on bi-attention mechanism and crisscross optimization. Energy. 2022;238:121795. doi:10.1016/j.energy.2021.121795.
12. Karijadi I, Chou SY, Dewabharata A. Wind power forecasting based on hybrid CEEMDAN-EWT deep learning method. Renew Energy. 2023;218:119357. doi:10.1016/j.renene.2023.119357.
13. Hanifi S, Zare-Behtash H, Cammarano A, Lotfian S. Offshore wind power forecasting based on WPD and optimised deep learning methods. Renew Energy. 2023;218:119241. doi:10.1016/j.renene.2023.119241.
14. Deng B, Wu Y, Liu S, Xu Z. Wind speed forecasting for wind power production based on frequency-enhanced transformer. In: Proceedings of the 2022 4th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI); 2022 Oct 28–30; Shanghai, China. p. 151–5. doi:10.1109/mlbdbi58171.2022.00036.
15. Jiang W, Liu B, Liang Y, Gao H, Lin P, Zhang D, et al. Applicability analysis of transformer to wind speed forecasting by a novel deep learning framework with multiple atmospheric variables. Appl Energy. 2024;353:122155. doi:10.1016/j.apenergy.2023.122155.
16. Xiong B, Lou L, Meng X, Wang X, Ma H, Wang Z. Short-term wind power forecasting based on Attention Mechanism and Deep Learning. Electr Power Syst Res. 2022;206:107776. doi:10.1016/j.epsr.2022.107776.
17. El Zaar A, Mansouri A, Benaya N, Bakir T, El Allati A. Hybrid Transformer-CNN architecture for multivariate time series forecasting: integrating attention mechanisms with convolutional feature extraction. J Intell Inf Syst. 2025;63(4):1233–64. doi:10.1007/s10844-025-00937-5.
18. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017); 2017 Dec 4–9; Long Beach, CA, USA.
19. Kiranyaz S, Avci O, Abdeljaber O, Ince T, Gabbouj M, Inman DJ. 1D convolutional neural networks and applications: a survey. Mech Syst Signal Process. 2021;151:107398. doi:10.1016/j.ymssp.2020.107398.
20. Mendoza-Pittí L, Calderón-Gómez H, Gómez-Pulido JM, Vargas-Lombardo M, Castillo-Sequera JL, de Blas CS. Developing a long short-term memory-based model for forecasting the daily energy consumption of heating, ventilation, and air conditioning systems in buildings. Appl Sci. 2021;11(15):6722. doi:10.3390/app11156722.
21. Navaneeth B, Suchetha M. A dynamic pooling based convolutional neural network approach to detect chronic kidney disease. Biomed Signal Process Control. 2020;62:102068. doi:10.1016/j.bspc.2020.102068.
22. 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; 2022 Jul 17–23; Baltimore, MD, USA.