IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 September 2023, accepted 20 October 2023, date of publication 23 October 2023, date of current version 31 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327135

RESEARCH ARTICLE

A Novel Two-Dimensional Convolutional Neural Network-Based an Hour-Ahead Wind Speed Prediction Method

MOHAMMADHOSSEIN NAZEMI, SHAIKAT CHOWDHURY, AND XIAODONG LIANG, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, SK S7N 5A9, Canada

Corresponding author: Xiaodong Liang (xil659@mail.usask.ca)

This work was supported in part by the University of Saskatchewan, Canada.

ABSTRACT With increasing penetration of wind power, accurate prediction of wind speed is essential for planning and operation of power grids. In this paper, a novel two-dimensional (2D) convolutional neural network (CNN)-based wind speed forecasting technique is proposed for an hour-ahead wind speed prediction. The wind speed at a specific time can be predicted in less than a few milliseconds using the proposed approach and meteorological data from a few hours earlier. The input feature selection, data preprocessing, and model evaluation of the proposed approach are presented; the efficiency of 2D CNN is compared to that of one-dimensional (1D) CNN, Long Short-Term Memory (LSTM), and Multi-Layer Perceptron (MLP). A three-year historical wind speed dataset from 2020 to 2022 collected at Saskatoon International Airport in Saskatoon, Saskatchewan, Canada, is used in this study. It is found that 2D CNN shows superior performance in addressing regression and prediction challenges. Experimental results verify that the proposed 2D CNN-based forecasting techniques can provide accurate wind speed prediction. Using deep learning for wind speed prediction can reduce costs while boost energy output and contribute to sustainable and green energy development in Saskatchewan and beyond.

INDEX TERMS Feature selection, meteorological data, short-term prediction, two-dimensional convolutional neural networks, wind speed forecasting.

I. INTRODUCTION

Wind power is increasingly integrated into power grids, and leads to significant challenges to planning and operation of power grids. Energy production of wind farms can be improved and maintenance costs can be reduced by analyzing wind patterns and accurately predicting wind speeds [1]. Wind power generation has been growing annually, according to the International Renewable Energy Agency (IRENA) [2], the total installed wind power generation capacity in 2021 was 733 GW worldwide, a 274% increase compared to 196 GW in 2011. Onshore wind farms are the most developed and widely used globally, while offshore wind farms are still in their early stages due to technical difficulties and cost constraints [3].

Wind speed prediction methods can be categorized according to their prediction horizons as follows: very short-term, short-term, medium-term, long-term, and very long-term prediction. Very short-term prediction is defined as the prediction within a few seconds to 30 minutes, the short-term prediction is from 30 minutes to 6 hours, the medium-term prediction is between 6 and 24 hours, the long-term prediction is between 24 to 72 hours, and the very long-time prediction is 72 hours or longer [4], [5]. In this paper, we focus on an hour-ahead short-term wind speed forecasting.

Wind speed forecasting can also be categorized as physical and statistical approaches [6]. The physical approach relies on numerical weather prediction (NWP) techniques through weather forecast data, such as temperature, pressure,

The associate editor coordinating the review of this manuscript and approving it for publication was Ehab Elsayed Elattar.

surface roughness, and obstacles [7]; it requires accurate input data and is computationally intensive, so is less suitable for real-time wind speed forecasting [6]. Statistical time series models are more suitable for short-term wind speed forecasting using historical data and statistical equations for data analysis [8].

Wind speed forecasting has been conducted using linear models, including the Kalman filter [9], moving average (MA) [10], exponential smoothing (ES) [11], locally weighted regression (LWR) [12], autoregressive moving average (ARMA) [13], and autoregressive integrated moving average (ARIMA) and the hybrid techniques [14], [15]. However, these models are usually static without considering environmental factors.

Due to recent advancements in machine learning techniques, wind speed prediction using machine learning and deep learning has attracted significant research interest. Machine learning techniques define relationships between wind speed and environmental factors including temperature, humidity and atmospheric pressure in wind speed prediction. Through feature extraction and pattern recognition, wind speed prediction can be adjusted using new data with significantly improved accuracy and reliability.

Commonly used machine learning algorithms for wind speed prediction include Artificial Neural Networks (ANNs), Support Vector Regression (SVR), and Random Forests etc. ANNs and the Multi-Layer Perceptron (MLP) models can learn complex relationships between input variables and the output [16], [17]. SVR aims to find a hyperplane that can maximize the margin between actual and forecasted values [18], [19]. Deep learning techniques, such as Convolutional Neural Networks (CNNs) [20], Long Short-Term Memory (LSTM) networks [21], and Generative Adversarial Networks (GANs), are very useful for dealing with complex and large datasets [22]. Non-linear models have been proposed for wind speed prediction, such as a non-linear autoregressive model developed in [23] to forecast day-ahead mean hourly wind speed using a general regression neural network to characterize non-linear patterns in datasets and improve prediction accuracy.

2D CNN is a deep learning technique. TABLE 1 highlights the distinctive features and advantages of 2D CNN in comparison to other machine learning and deep learning techniques. Although 2D CNN offers unique features, such as spatial information extraction, temporal feature learning, automatic feature extraction, hierarchical learning, transfer learning, and pre-training, currently, there is very limited research exploring 2D CNN applications in wind speed prediction. In this paper, we aim to develop a novel 2D CNN-based accurate wind speed forecasting technique using three-year historical wind speed data recorded from 2020 to 2022 at Saskatoon International Airport in Saskatoon, Canada.

The main contributions of this paper include

1) A novel 2D CNN-based wind speed forecasting technique is proposed for an hour-ahead wind speed prediction.
2) Temporal Context Capture: By taking into account the weather data from a small number of previous hours, the 2D CNN effectively captures the temporal context and dependencies in wind speed, which leads to more accurate predictions as the model learns patterns and trends over time.
3) Spatial Feature Extraction: The 2D CNN efficiently extracts spatial features from the multidimensional weather data, and captures complex relationships between meteorological variables and wind speed. This overcomes the limitations of traditional techniques, which may struggle to identify such associations.
4) Automatic Feature Learning: The 2D CNN learns relevant features directly from the raw data, eliminating the need for manual feature engineering. This improves wind speed forecasts by leveraging the mode's ability to automatically learn discriminative features.
5) Nonlinear Relationships: The 2D CNN captures complex nonlinear relationships between weather variables and wind speed. By modeling intricate dependencies, the model uncovers patterns that may not be apparent with linear models or shallow machine learning algorithms, leading to enhanced forecasting accuracy.

The paper is arranged as follows: in Section II, the novel 2D CNN-based an hour ahead wind speed prediction method is proposed and its implementation procedure is explained; in Section III, historical wind speed datasets used in this study are discussed; in Section IV, the principles of 2D CNN and other four deep learning algorithms used for comparison are briefly introduced; the data preprocessing method, and the system training procedure and results analysis are provided in Sections V and VI; in Section VII, the proposed 2D CNN-based method is compared with other four deep learning-based wind speed forecasting; conclusions are drawn in Section VIII.

II. THE PROPOSED METHOD

In this paper, a novel 2D CNN-based wind speed forecasting method is proposed using historical wind speed along with other meteorological data. The 2D CNN model is pre-trained offline using a large dataset to learn complex patterns and extract informative features. Once trained, the model can be used online for real-time wind speed forecasting applications, efficiently scan a small number of weather data in past hours to generate accurate wind speed predictions for upcoming hours. This approach offers a fast and efficient decision-making process that can optimize the operation of wind energy systems, improve energy efficiency, and reduce operating costs. As shown in Fig. 1, the proposed method can be implemented in the following five steps:

Step 1: Define the problem and set up the environment. This step involves defining the problem of wind speed forecasting with specific objectives, input and output data, and performance metrics.

Step 2: Data collection and preprocessing. This step involves collecting and preprocessing relevant data for wind

TABLE 1. Unique features and advantages of 2d cnn compared to other machine learning and deep learning methods.

Property	2D CNN	ANN	SVR	MLP
Spatial Information Extraction	2D CNN can capture spatial dependencies in wind speed data effectively by treating it as an image	ANN might struggle to extract spatial information effectively [24]	SVR does not explicitly consider spatial dependencies in the data [25]	MLP might require additional preprocessing or feature engineering to incorporate spatial information [26]
Temporal Feature Learning	2D CNN can capture temporal features by incorporating a time dimension into the input data and effectively learning and exploiting temporal patterns in wind speed time series [27]	ANN can capture temporal patterns but may require additional design choices or preprocessing [24]	SVR can capture some temporal patterns, but it primarily focuses on modeling the statistical relationship between input and output variables [28]	MLPs are primarily designed for feed-forward tasks and are not naturally suited for capturing temporal dependencies in sequential data [26]
Automatic Feature Extraction	2D CNNs can automatically learn relevant spatial and temporal features from raw wind speed data, eliminating the need for explicit feature engineering [29]	ANN may require manual feature engineering, which can be time-consuming and may not capture all the relevant patterns in the wind speed data [30]	SVR relies on handcrafted features or kernels for capturing relevant patterns [31]	MLP may require manual feature engineering to identify and extract informative features [26]
Hierarchical Learning	2D CNN employ multiple layers with hierarchical feature representations, allowing them to capture complex spatial and temporal relationships in the wind speed data [32]	ANN can capture hierarchical relationships but may require additional design choices to model complex relationships effectively [33]	SVR does not inherently possess a hierarchical learning structure [30]	MLPs are not inherently designed for hierarchical learning, as they lack the explicit mechanisms to capture hierarchical relationships in data [30]
Transfer Learning and Pre-training	2D CNN can leverage transfer learning and pre-training techniques, allowing them to capture general spatial and temporal patterns from large datasets and fine-tune the model on specific wind speed data [32]	ANN can benefit from transfer learning and pre-training, but it may require more effort to adapt the pre-trained models to wind speed forecasting [33]	SVR does not typically incorporate transfer learning or pre-training techniques [30]	MLP can benefit from transfer learning and pre-training, but it may require more effort to adapt the pre-trained models to wind speed forecasting [30]

speed forecasting. The meteorological data, such as wind speed, temperature and pressure are collected and preprocessed to create appropriate input formats for the 2D CNN model.

Step 3: Model construction. A 2D CNN model is constructed to predict wind speed. The model architecture is designed to extract relevant features from input data. Different types of convolutional layers, pooling layers, and activation functions can be used to achieve this.

Step 4: Model training, validation and testing. The CNN model is trained and validated using a large dataset of preprocessed data. The training process involves optimizing the model's parameters based on the mean squared error loss function. The validation process is used to test the accuracy of the model and tune its parameters to minimize errors. Testing is performed using unseen historical data or real-time data from sensors and meteorological stations. The model can be evaluated by comparing predicted wind speeds to actual wind speeds, and error metrics, such as the mean absolute error and the root mean square error.

Step 5: Continuous monitoring and updating of the model. The CNN model should be continuously updated to ensure its accuracy in wind speed forecasting by collecting new data, retraining the model periodically, and adjusting its parameters as necessary.

III. HISTORICAL WIND SPEED DATASET

The data used in this paper are publically available at the Government of Canada website [34], and we picked the data measurement location as Saskatoon International Airport in Saskatoon, Canada for the data download. Several years of historical meteorological data from 2020 to 2023 recorded at Saskatoon International Airport in [34] serve as the datasets in this study. Saskatoon International Airport is located at the elevation of 504.10 m, the latitude of 52°10'15.000" N, and the longitude of 106°42'00.000" W [34]. The wind direction and wind speed were monitored ten meters above the ground.

The datasets include hourly data for 24 hours per day and 365 days per year with the following seven major parameters, which serve as features for the wind speed prediction (the measurement time refers to the Local Standard Time (LST)):

1) *Temperature*: The air temperature, expressed in degrees Celsius (°C).
2) *Dew Point Temperature*: The temperature at which cooling would cause the air to become saturated with liquid water, measured in degrees Celsius (°C).
3) *Relative Humidity*: The amount of water vapor in the air relative to the maximum amount it can store at that given temperature, measured in %.
4) *Wind Direction*: The real or geographic direction of the wind, measured in tens of degrees.

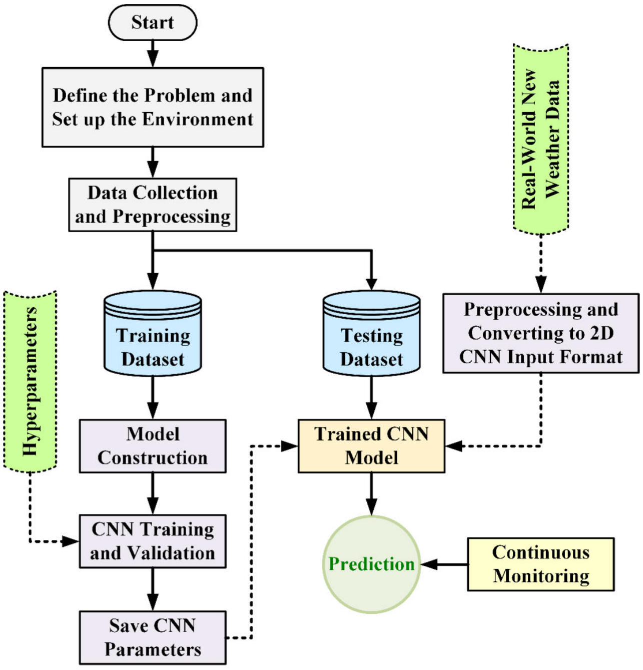
Start
Define the Problem and Set up the Environment
Data Collection and Preprocessing
Real-World New Weather Data
Hyperparameters
Training Dataset
Testing Dataset
Preprocessing and Converting to 2D CNN Input Format
Model Construction
Trained CNN Model
CNN Training and Validation
Prediction
Continuous Monitoring
Save CNN Parameters

FIGURE 1. The flow chart of the proposed 2D CNN-based wind speed forecasting method.

5) *Visibility*: The distance at which objects of an appropriate size may be seen, recognized, measured in kilometers (km).
6) *Station Pressure*: The station elevation's atmospheric pressure, measured in kilopascals (kPa).
7) *Wind Speed*: The speed of air movement, measured in kilometers per hour (km/h).

Each of the yearly datasets in 2020, 2021 and 2022 has 8,760 rows of data. The developed models are eventually tested using the first three months (January to March) data in the 2023 dataset. Fig. 2 shows physical features of the data in January and February 2022 with 1,416 measurement hours (the horizontal axis).

Fig. 3 illustrates the correlation between meteorological measurements. According to this heatmap correlation graph, temperature has a positive correlation of 0.1 with wind speed, indicating that as the temperature increases, there is a slight tendency for wind speed to increase. This can be attributed to that higher temperatures often lead to increased atmospheric instability, and thus, result in stronger air movements and wind speeds. Dew point temperature shows a very weak positive correlation of 0.04 with wind speed, which reflects the moisture content in the air, and may have a minor influence on wind speed. Relative humidity has a negative correlation of −0.18 with wind speed, suggesting that higher relative humidity levels are associated with slightly lower wind speeds. This is because higher humidity often indicates a more stable atmospheric condition, which can limit the intensity of wind patterns. Visibility shows a negative correlation of −0.073 with wind speed, implying that the reduced visibility may be associated with slightly higher wind speeds. This can be attributed to fog or heavy precipitation, which often occur in turbulent conditions and can lead to increased wind speeds. Station pressure demonstrates a moderate negative correlation of −0.25 with wind speed. Lower pressure systems are often accompanied by stronger wind patterns due to variations in atmospheric circulation, resulting in increased wind speeds [34].

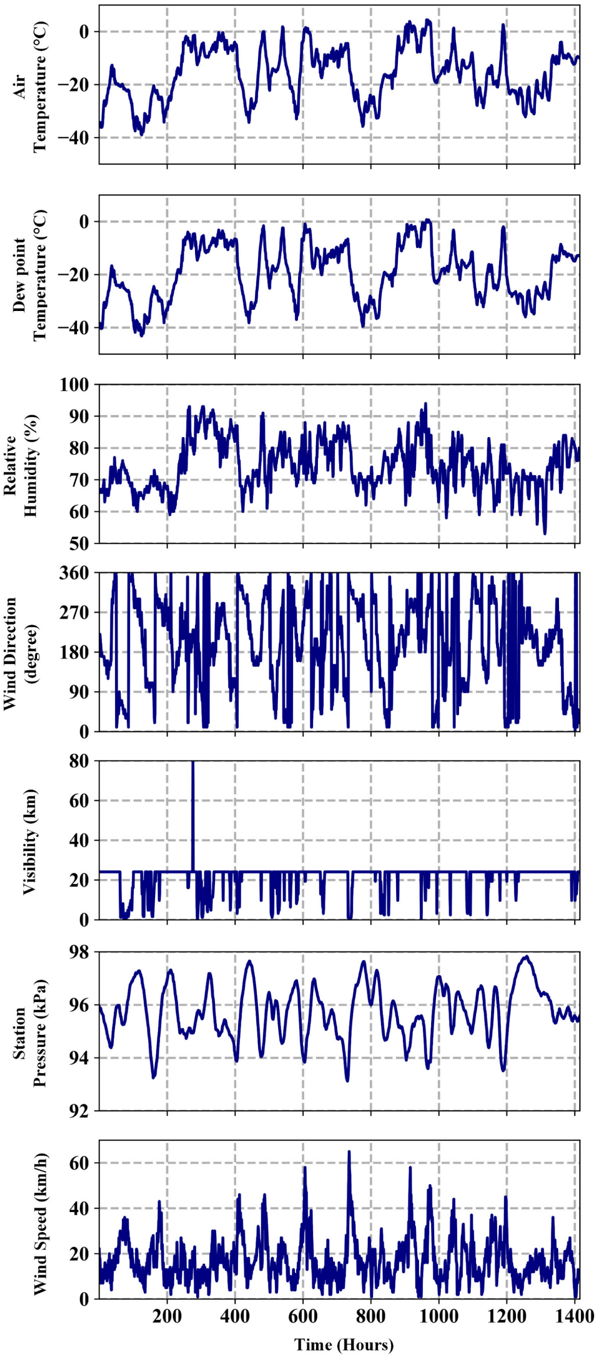

FIGURE 2. Physical parameter variations for the data measured in January and February, 2022.

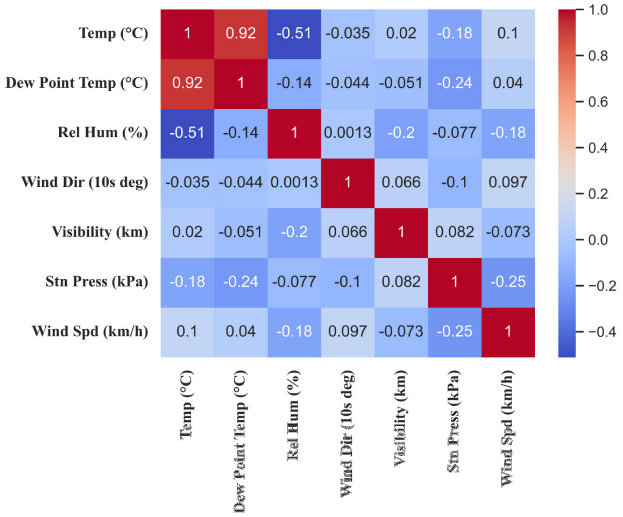

FIGURE 3. The correlation values between weather observations.

Despite weak individual correlations between the selected parameters and wind speed, the utilization of these parameters in wind speed forecasting through a 2D CNN offers advantages, including multivariate analysis and exploration of complex relationships. Incorporating multiple parameters enables the model to capture the combined effects and potential interactions among these variables. While their individual correlations may be weak, the collective information they provide can offer valuable insights for wind speed prediction.

2D CNN can capture nonlinear relationships and patterns that may not be visible through simple correlation analysis. It can effectively learn the complex dependencies between the input parameters and wind speed, leading to improved forecast accuracy.

Fig. 4 illustrates the pair-plot graph, which provides a comprehensive visualization of the relationship between the parameters (temperature, dew point temperature, relative humidity, wind direction, visibility, station pressure) and wind speed. The diagonal section of the graph presents the distribution of each parameter individually, allowing for an in-depth analysis of their frequency distributions and the identification of noteworthy patterns or outliers. Notably, the majority of recorded wind speeds in the dataset are concentrated around 18 km/h; while the station pressure falls mostly within the range of 96 kPa.

In the off-diagonal section of the graph, scatter plots portray the pairwise relationships between each parameter and wind speed. These scatter plots enable the detection of potential correlations or dependencies among the variables. By closely examining the scatter plots, it becomes possible to recognize patterns indicative of linear or nonlinear relationships, thereby revealing the strength and direction of the connections between the parameters and wind speed.

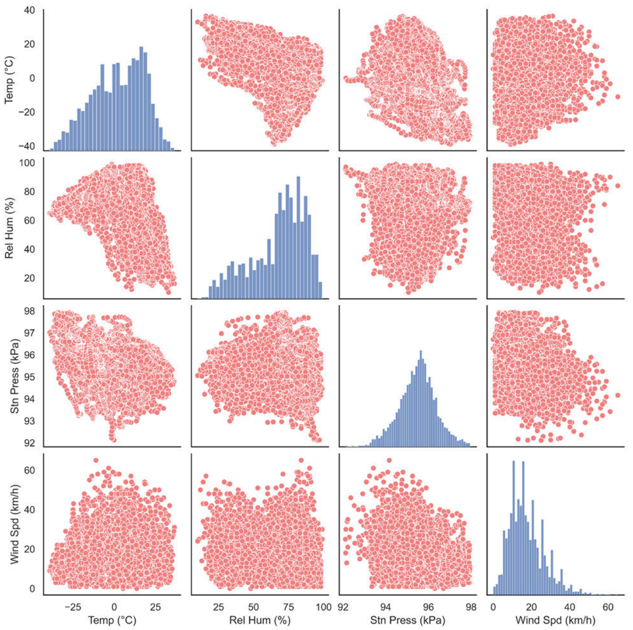

FIGURE 4. The pair plot of the weather information which has the highest correlation to wind speed.

IV. FUNDAMENTAL THEORY OF DEEP LEARNING APPROACHES IN TIME SERIES PREDICTION

In this paper, a novel 2D CNN-based short-term wind speed prediction method is proposed, which is further compared with other four deep learning-based methods, 1D CNN, LSTM, MLP, and Rough Autoencoder (RAE), for validation. In this section, the principles of all five deep learning methods are briefly introduced.

A. 2D CNN

2D CNN is a powerful deep learning algorithm that is widely used in the image analysis and recognition tasks. It can also be used for time series forecasting, such as stock prices or weather data, by converting the time series data into a 2D image format [35]. Generally, this conversion is achieved through the sliding window technique, which generates a sequence of two-dimensional images. Each image corresponds to a slice of the time series data [36]. The generated images can be either color or grayscale. For a color image, it refers to a 3D tensor with three pixel values in each coordinate, and the size of (width, height, 3); while for a grayscale image, it refers to a 2D tensor with one pixel value per position with the size of (width, height, 1). When creating an image database from a time series dataset, it is common to have just one value in each position. As a result, each image is typically classified as either a 2D tensor or grayscale [37]. The image dataset is then used to train the system along with its corresponding output labels. This enables the system to predict the value of a future time step accurately.

A 2D CNN architecture typically consists of convolutional layers, pooling layers, and fully connected layers.

The convolutional layers are responsible for extracting the main features using a set of filters to identify relevant patterns in the input data. The pooling layers then downsample the feature maps generated by the convolutional layers, which reduces computation and prevents overfitting. Finally, the fully connected layers combine the extracted features to make a prediction. There are numerous variations and modifications to this basic 2D CNN architecture that can be explored depending on the particular study and data being processed. An illustration of 2D CNN is shown in Fig. 5.

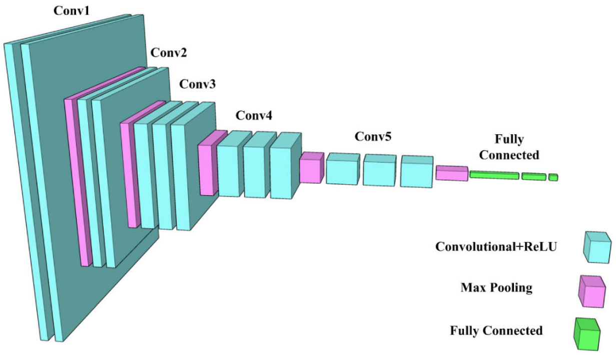

FIGURE 5. A graphical representation of a 2D CNN architecture.

B. 1D CNN

1D CNN is often used to analyze time series/sequential data and is suitable for classification, regression, and anomaly detection due to its capability to automatically learn features from raw data. The standard components of 1D CNN are the input layer, convolutional layers, pooling layers, and fully connected layers. The input layer receives the sequential data, such as time series or text, prepares it for processing, and the input data is slid over by a filter. Convolutional layers compute the dot products at each point to extract local patterns or features. Pooling layers summarize the data and downsample the output of convolutional layers, and the output of pooling layers is sent to the fully connected layers, which then turns it into a final output for classification or regression prediction [38]. Depending on the particular study and data being processed, the architecture of 1D CNN and the layer count may change. An illustration of 1D CNN is shown in Fig. 6.

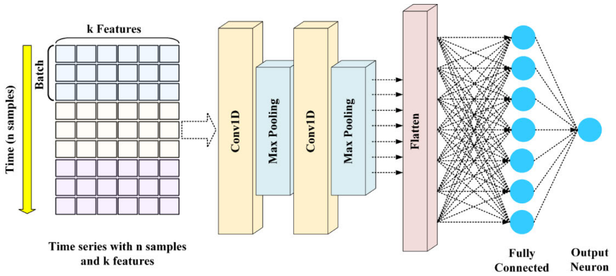

FIGURE 6. The graphical representation of a 1D CNN.

C. LSTM

The LSTM is a type of Recurrent Neural Network (RNN) that contains a memory cell within the hidden layer to manage the memory information of the time series data. The memory cell is governed by three gates: the forget gate, the input gate, and the output gate. The forget gate determines how much information from the previous time step should be retained; the input gate controls how much information from the current time step should be added to the memory cell; and the output gate determines how much of the memory cell's current state should be propagated to the next layer in the network. This gating mechanism enables the LSTM to selectively retain or discard information from past time steps, enabling it to remember long-term dependencies in the data.

Compared to traditional RNNs, the LSTM can overcome the gradient disappearance issue, which hinders the network's ability to learn and remember long-term dependencies. The utilization of the memory cell and gating mechanism in the LSTM allows for the management of memory and forgetting of past and current information. The LSTM network's structure is shown in Fig. 7, which explains that the gates are controlled by the sigmoid (σ) function ranging from 0 to 1, 0 means that no information should pass through the gate, and 1 indicates that all information should be permitted to pass through the gate. To address the gradient disappearance issue, the hyperbolic tangent function is used to ensure that the network's parameters are within a reasonable range [39], [40].

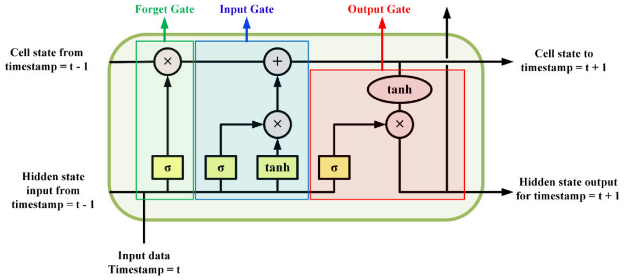
Forget Gate
Input Gate
Output Gate
Cell state from timestamp = t - 1
Cell state to timestamp = t + 1
×
+
tanh
×
×
σ
σ
tanh
σ
Hidden state input from timestamp = t - 1
Hidden state output for timestamp = t + 1
Input data Timestamp = t

FIGURE 7. The schematic diagram of LSTM.

D. MLP

The MLP is a type of neural network, where the information flows in one direction (the feedforward neural network) from the input layer to the output layer through multiple layers of interconnected neurons. Each neuron takes inputs from the previous layer's neurons and produces outputs that are sent to the next layer's neurons. The input layer receives the input data, such as wind speed measurements, and the input data is sent to hidden layers. The hidden layers process the input data and produce the transformed output data that is more suitable for the final output layer. The output layer generates the final prediction, which in this case is the predicted wind speed. The hidden layers of an MLP are responsible for learning complex relationships between input features and output prediction.

During the training process, the weights between the neurons in each layer are adjusted by minimizing the difference between predicted and actual outputs. This process is repeated iteratively until the error is minimized and the MLP produces accurate predictions [41]. Fig. 8 provides a basic visual representation of the MLP structure.

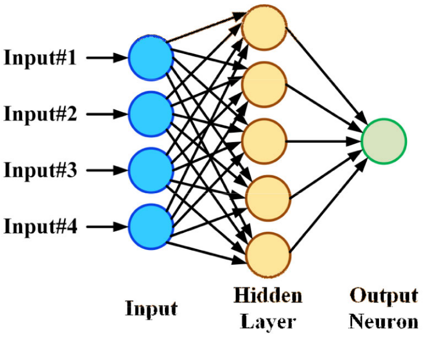

FIGURE 8. The MLP basic model.

E. ROUGH AUTOENCODER

The Rough Autoencoder (RAE) is a specialized neural network variant tailored for handling uncertain data effectively by combining the rough set theory with autoencoders, making RAE very valuable for wind speed forecasting. The architecture of RAE includes an encoder-decoder structure with a unique rough set layer for handling uncertain data patterns. Training involves minimizing the reconstruction error to ensure RAE represents and reconstructs uncertain data efficiently [42].

RAE excels with low-quality data, which are common in wind speed forecasting, as RAE can robustly handle noises and incompleteness in the data. It automates features selection, which eliminates the need for the manual feature engineering. In wind speed forecasting, RAE can preprocess uncertain meteorological data, enhance the model accuracy and mitigate data imperfections, leading to advanced data handling for more precise predictions [43].

V. DATA PREPROCESSING AND HYPERPARAMETERS

In this section, the essential data preprocessing steps are described, ensuring data consistency and an optimal format for our 2D CNN model. This process involves feature normalization, data transformation into 2D images, resizing to an ideal image size, and alignment of input and output vectors. These steps establish the groundwork for accurate wind speed predictions.

A. PERFORMANCE EVALUATION INDICES

There is a range of factors to take into account while assessing the performance of a regression model. The commonly used regression model evaluation metrics include Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), they can be calculated as follows [6] and [38]:

$$MAE = \frac{\sum_{i=1}^{N} \left| y\left(i\right) - \hat{y}(i) \right|}{N} \tag{1}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left[y\left(i\right) - \hat{y}(i) \right]^2}{N}} \tag{2}$$

where N is the number of data points, $y(i)$ is the observed wind speed data at the i^{th} sample of the time vector, and $\hat{y}(i)$ is the forecasted wind speed value. A lower value of these indices indicates a better model. They provide a thorough evaluation of the effectiveness of a regression model.

B. STANDARDIZATION AND IMAGE DATASET GENERATION

To train the proposed model, a portion of the input data vector (X) and the output true labels vector (y) (the training dataset) must be provided to 2D CNN based on the system standard feed. The following three stages must be implemented to accomplish this task:

Stage 1: Since the features differ in the sign and range of the values within the dataset, these features must be firstly normalized to standardize all the data input for 2D CNN. In the context of convolutional operation, the recommended number format is the float values between −1 and 1, which is the best number format that internal layers of 2D CNN can recognize. To accomplish this, a normalization function, known as "maxabs_scale", is used from the preprocessing class in the Python Scikit-learn package to normalize the data. "maxabs_scale" is a scaling technique by dividing each feature vector by its maximum absolute value, which ensures their values are within the range between −1 and 1:

$$X_{scaled} = X \big/ \max(abs\left(X\right)) \tag{3}$$

where X is the original feature vector, and X_{scaled} is the scaled feature vector. This technique is particularly useful for sparse matrices by preserving its sparsity, and it is also robust to outliers in the data.

Stage 2: Using a 2D CNN, the wind speed at a particular date and hour can be estimated by analyzing the seven meteorological data (the temperature, dew point temperature, relative humidity, visibility, station pressure, wind direction, and wind speed) from an arbitrary number of data measured at previous hours. The preferred image size for 2D CNN would be identical in width and length. Since seven meteorological data were measured hourly, the most optimal image size would be 7 × 7, i.e., each sliding window sweeps seven lines of the dataset containing seven normalized features. Therefore, we can forecast the wind speed at the 8th hour by analyzing the data measured at the previous seven hours.

The sliding window moves down one line at a time, scanning all the data in the dataset to create new images. The sliding window continues to move forward until it reaches one line left to the end line of the dataset (as there is no further data available for estimation beyond the last point). As a result, there will be a total of 8,753 grayscale images generated from the existing time series database, and the final X shape will be in a cubic image dataset format of (8,753, 7, 7, 1).

The optimization of 2D convolutional operations is facilitated by the image dataset's expansion. Although the image dimension selection seems somewhat arbitrary, an extreme enlargement may cause heavy processing burden. Larger images can capture intricate details, and enable the detection of complex, high-level features by 2D CNN filters. The information loss may occur when using smaller images, particularly for complex time series data. Filter kernels and pooling operations can systematically reduce the feature image size by half through each layer. Therefore, using smaller images requires fewer convolution layers with constrained filters and frequent pooling, reduces forecast accuracy, and increases risks of overfitting and regression failures. These concerns can be mitigated by using larger images, but excessively large images donot necessarily guarantee superior regression and prediction accuracy, and can substantially increase computational burdens. Caution should be exercised to achieve a careful balance between image dimensions and computational resources. In this paper, an optimal image size of 64 × 64 has been chosen, which is further extended by the conventional interpolation technique, "bilinear".

Ultimately, in this paper, the final dimension of the input image dataset for the 2D CNN training is chosen to be (8,753, 64, 64, 1). Fig. 9 shows a generated grayscale image by sliding window technique from the time series data in this study.

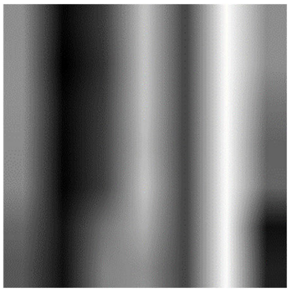

FIGURE 9. A grayscale image generated from the time series data.

Stage 3: In the proposed method, the estimation of the eighth hour wind speed is based on utilizing the first seven samples of the dataset (representing the weather information from the initial seven hours). As a result, the output vector (y) for the training sequence should begin with the eighth sample of actual wind speed data and continue to the end, so the y vector is 8, 753 × 1 in size. This vector is considered the true label for the regression model. Thus, the length of the feature vector matches the length of the output vector.

C. 2D CNN HYPERPARAMETERS TUNING

TABLE 2. The layers architecture of the proposed 2D CNN.

Layer (Type)	Output Shape	No. of Parameters
Conv2D	(64,64,16)	160
MaxPooling2D	(32,32,16)	0
Conv2D	(32,32,32)	4640
MaxPooling2D	(16,16,32)	0
Conv2D	(16,16,64)	18496
MaxPooling2D	(8,8,64)	0
Conv2D	(8,8,128)	73856
MaxPooling2D	(4,4,128)	0
Dropout	(4,4,286)	0
Flatten	(2048)	0
Dense	(32)	6558
Dense	(1)	33
Total Parameters	162,753	
Trainable parameters	162,753	

Since the 2D CNN algorithm processes images and requires a larger processing volume of CPU, a batch size of 50 and the epoch number of 100 are chosen. TABLE 2 displays the topology for 2D CNN layers to predict an hour-ahead wind speed based on TensorFlow's suggestion and trial-and-error.

To train the system. "Adam" is the most effective optimizer algorithm for 2D CNN, which is a stochastic gradient descent technique based on the adaptive estimate of the first-order and second-order moments. This approach works well for issues with plenty of data and parameters as it is computationally efficient, requires minimal memory and is invariant to the diagonal rescaling of gradients. The learning rate of the Adam optimizer is set to 0.001 [44]. In this paper, the CNN architecture is enhanced by increasing the filter count from 16 to 64 in each convolutional layer with a 3 × 3 kernel size. Various pooling window sizes, notably 2 × 2, were tested to optimize the feature map downsizing. Dropout layers are introduced with a tuned dropout rate of 0.2 to prevent overfitting. "ReLU" and its variants serve as activation functions for hidden layers, introducing crucial non-linearity. These adjustments aim to improve the network's ability to capture intricate spatial patterns while preventing overfitting, resulting in a more robust model for wind speed prediction using meteorological data. MAE and RMSE, which compute the mean absolute error and the root mean squared error between the true y and predicted y, respectively, are specified as the metrics functions in (1) and (2). Generally, 80% of an entire shuffled X and y vector is used as the training dataset (X_train and y_train), and the rest 20% is used as the validation dataset (X_val and y_val).

Algorithm 1 provides a clear illustration of the procedure, starting from importing the meteorological data in a spreadsheet format into the coding platform, to preparing the training and validation datasets for feeding into the 2D CNN model.

VI. SYSTEM TRAINING AND RESULT ANALYSIS

The proposed 2D CNN-based wind speed prediction technique is trained using X_train and y_train in 2020, 2021, and 2022 datasets, using the hyperparameters mentioned above, and its performance is measured using X_val and y_val for the same years. Fig. 10 provides a graphical representation on

Algorithm 1 Training and Validation Dataset Generation

```
Input:   Dataset of hourly climate recorded in a
         spreadsheet format
Output:  Training and validation dataset
df = convert to data frame (Input)
(row = n, column = k) = size (df)
x = df [Temp, Dew Point Temp, Rel Hum, Wind
   Dir, Visibility, Stn Pres, Wind Speed]
x = maxabs_scale (X [row = 1 to n-1, column = 1
   to k])
shape (X) = (8759, 7)
y = df [Wind Speed] [row = 8 to n]
shape (y) = (8753, 1)
sliding window shape = (7, 7)
initial 3D matrix dataset = zeros vector (8753, 7, 7)
for i = 1 to i = 8753
    k = sliding window scans x[row = i to i + 7,
       column = 7]
    initial 3D matrix dataset [i] = k
Endfor
X = initial 3D matrix dataset
Shape (X) = (8753, 7, 7)
For grayscale image dataset, the dimension of X
should be expanded:
expand X dimension to (8753, 7, 7, 1)
For preparing image dataset to fed 2D CNN the
image size should be stretched by the nearest or
bilinear technique:
resize X shape to (8753, 64, 64, 1)
Final X shape = (8753, 64, 64, 1)
Final y shape = (8753, 1)
This condition should be always met:
 if length (X) == length (y)
 continue
 else print (verify the dimensions of the initial
 vectors)
(X_train, y_train), (X_val, y_val)
= train_test_split function from scikit-learn
(X, y, test size = 20%, shuffle = True)
training dataset = tensor integrating (X_train,
y_train)
validation dataset = tensor integrating
(X_val, y_val)
compile 2D CNN model (optimizer = Adam, loss
= MSE, metrics = [MAE, RMES])
Ready for training and validation of 2D CNN
model
```

how training and validation metrics evolve with the increasing number of epochs for the 2022 dataset. Additionally, TABLE 3 offers insights into the training time of the 2D CNN and presents performance evaluation indices for the trained model. Notably, the model achieves an average MAE of approximately 3.66 km/h on the validation dataset over a span of three years. This MAE is relatively low, particularly when contrasted with the average wind speed observed throughout the year.

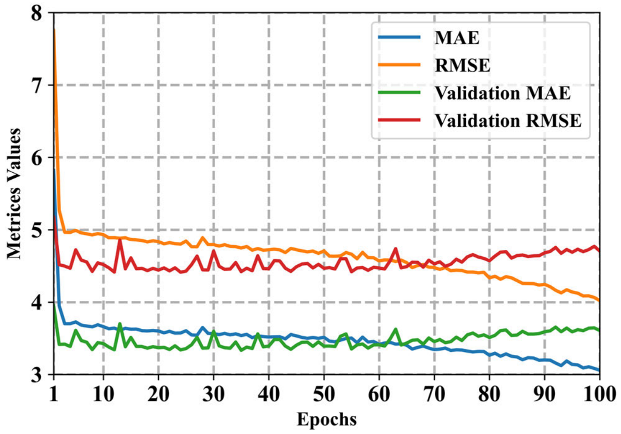

FIGURE 10. Training and validation metrics (MAE and RMSE) against epochs for the 2022 dataset.

TABLE 3. Performance evaluation of the trained system.

Dataset	Training Time (s)	Validation MAE	Validation RMSE
2020	302.671	3.619	4.696
2021	316.162	3.836	4.986
2022	298.565	3.520	4.633

Fig. 11 shows the performance of the proposed regression model through a hybrid graph, including a scatter plot of the measured wind speeds vs. the forecasted values, and a histogram plot of true labels and predictions, integrated with a Kernel Density Estimate (KDE) graph for three years. According to the performance evaluation of the trained model and the distribution of the predicted values, the best fit is found in 2022 with minimal outliers, unusual data, and nonnormality.

The trained model based on the 2022 dataset can be used to predict an hour-ahead wind speed for the first three months of 2023. The performance of the model is evaluated using MAE and RMSE, which are calculated as 3.314 and 4.296, respectively. The predicted values are compared to the measured values for January, February, and March separately, and the results are presented in Fig. 12. The comparison shows a good match between the forecasted and measured data, indicating that the model is performing well.

VII. COMPARING WITH OTHER DEEP LEARNING METHODS

The datasets collected at Saskatoon International Airport have both spatial and temporal information. The spatial information includes the airport location, wind sensors, and the topography, elevation, and surrounding land use. The temporal information includes the time of day, day of the week, and month of the year [45], [46]. 2D CNN has strength in capturing spatial features; LSTM and 1D CNN are better

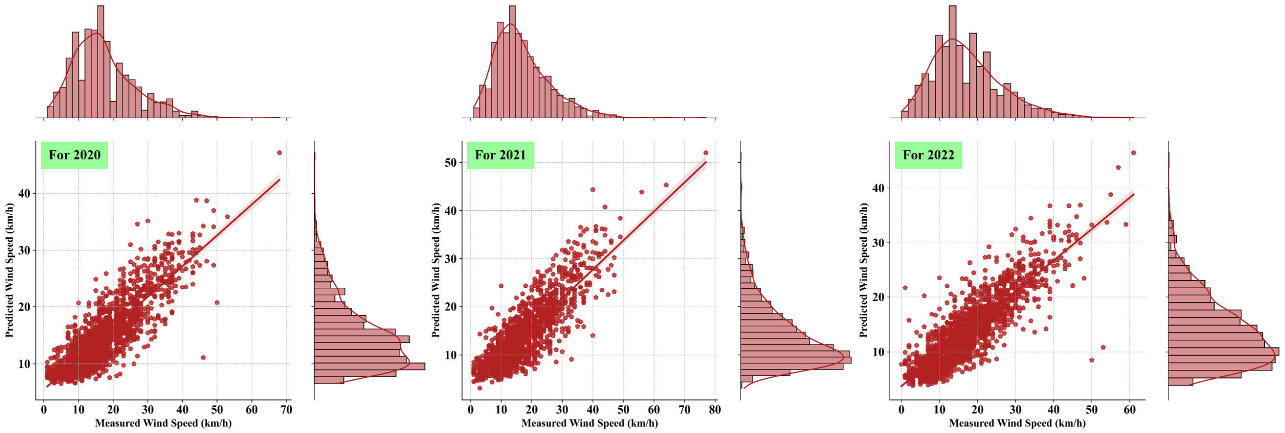

FIGURE 11. Scatter and histogram plots of the measured and predicted wind speed for the 2020, 2021, and 2022 datasets.

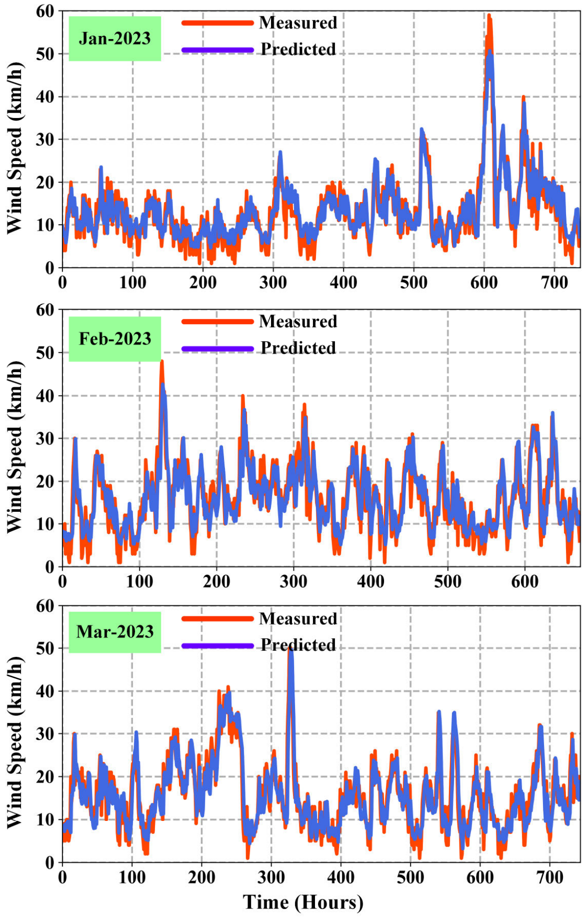

FIGURE 12. Wind speed prediction for the first three months of 2023 based on the 2022 regression model.

suited to capture temporal dependencies; RAE excels when handling the data with uncertainties and enhancing the forecast accuracy with noisy or incomplete data; while MLP is a simpler approach, it may not perform as well as other methods for complex datasets. Various algorithms should be assessed for the given datasets, and the one yields the best results should be chosen [45].

TABLE 4. Performance assessment of the proposed 2D CNN-based approach vs. 1D CNN, LSTM, MLP, and RAE techniques using the 2022 dataset.

Method	Training Time (s)	Validation MAE	Validation RMSE
1D CNN	78.959	4.262	4.696
LSTM	204.389	3.626	4.734
MLP	51.219	3.871	4.809
RAE	83.092	3.601	4.923
2D CNN	298.565	3.520	4.633

In this section, a comparison is made between the proposed 2D CNN-based method and the methods using the four deep learning techniques, 1D CNN, LSTM, MLP, and RAE, and assessed through the metrics, MAE and RMSE, using a portion of the 2022 dataset, as shown in Table 4 (the wind speed is measured in km/h).

The 2D CNN model exhibits the strongest performance, with the lowest MAE of 3.520 km/h and RMSE of 4.633 km/h, among the five deep learning methods, which indicates that the proposed 2D CNN-based method provides superior accuracy in the short-term wind speed prediction.

VIII. CONCLUSION

Accurate wind speed forecasting is essential to ensure proper planning and operation of wind farms and power grids with high wind power penetration. In this paper, a novel short-term 2D CNN-based wind speed prediction method is proposed. Several years of historical wind speed and other meteorological data from 2020 to 2023 measured in Saskatoon, Canada are used to train, validate and test the proposed model. With an average MAE of 3.66 km/h, the trained 2D CNN model can predict one hour-ahead wind speed in less than a few milliseconds. By comparing the proposed 2D CNN-based method with four deep learning methods, 1D CNN, LSTM, MLP, and

RAE, the proposed method shows superior performance in short-term wind speed prediction. It can improve economic and reliable operations of wind farms in Saskatchewan and beyond.

REFERENCES

[1] Y. Wang, Z. Zhou, A. Botterud, and K. Zhang, ''Optimal wind power uncertainty intervals for electricity market operation,'' *IEEE Trans. Sustain. Energy*, vol. 9, no. 1, pp. 199–210, Jan. 2018.

[2] *Wind Energy*, International Renewable Energy Agency (IRENA), Masdar City, United Arab Emirates, 2011. [Online]. Available: https://www.irena.org/Energy-Transition/Technology/Wind-energy

[3] S. D. Ahmed, F. S. M. Al-Ismail, M. Shafiullah, F. A. Al-Sulaiman, and I. M. El-Amin, ''Grid integration challenges of wind energy: A review,'' *IEEE Access*, vol. 8, pp. 10857–10878, 2020.

[4] H. S. Dhiman and D. Deb, ''A review of wind speed and wind power forecasting techniques,'' 2020, *arXiv:2009.02279*.

[5] M. Santhosh, C. Venkaiah, and D. M. Vinod Kumar, ''Current advances and approaches in wind speed and wind power forecasting for improved renewable energy integration: A review,'' *Eng. Rep.*, vol. 2, no. 6, Jun. 2020, Art. no. e12178.

[6] X. Zhao, S. Wang, and T. Li, ''Review of evaluation criteria and main methods of wind power forecasting,'' *Energy Proc.*, vol. 12, pp. 761–769, 2011.

[7] X. Wang, P. Guo, and X. Huang, ''A review of wind power forecasting models,'' *Energy Proc.*, vol. 12, pp. 770–778, 2011.

[8] L. Xiao, W. Shao, M. Yu, J. Ma, and C. Jin, ''Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system forecasting,'' *Appl. Energy*, vol. 198, pp. 203–222, Jul. 2017.

[9] C. D. Zuluaga, M. A. Álvarez, and E. Giraldo, ''Short-term wind speed prediction based on robust Kalman filtering: An experimental comparison,'' *Appl. Energy*, vol. 156, pp. 321–330, Oct. 2015.

[10] Z. Tian, G. Wang, and Y. Ren, ''Short-term wind speed forecasting based on autoregressive moving average with echo state network compensation,'' *Wind Eng.*, vol. 44, no. 2, pp. 152–167, Apr. 2020.

[11] E. Cadenas, O. A. Jaramillo, and W. Rivera, ''Analysis and forecasting of wind velocity in Chetumal, Quintana Roo, using the single exponential smoothing method,'' *Renew. Energy*, vol. 35, no. 5, pp. 925–930, May 2010.

[12] W. S. Cleveland and S. J. Devlin, ''Locally weighted regression: An approach to regression analysis by local fitting,'' *J. Amer. Stat. Assoc.*, vol. 83, no. 403, pp. 596–610, Sep. 1988.

[13] Y. Zhang, S. Gao, J. Han, and M. Ban, ''Wind speed prediction research considering wind speed ramp and residual distribution,'' *IEEE Access*, vol. 7, pp. 131873–131887, 2019.

[14] K. Yunus, T. Thiringer, and P. Chen, ''ARIMA-based frequency-decomposed modeling of wind speed time series,'' *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 2546–2556, Jul. 2016.

[15] M. U. Yousuf, I. Al-Bahadly, and E. Avci, ''Short-term wind speed forecasting based on hybrid MODWT-ARIMA-Markov model,'' *IEEE Access*, vol. 9, pp. 79695–79711, 2021.

[16] B. Jena, D. Swain, and A. Tyagi, ''Application of artificial neural networks for sea-surface wind-speed retrieval from IRS-P4 (MSMR) brightness temperature,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 3, pp. 567–571, Jul. 2010.

[17] J. Reynolds, M. P. Clarizia, and E. Santi, ''Wind speed estimation from CYGNSS using artificial neural networks,'' *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 708–716, 2020.

[18] Q. Hu, S. Zhang, M. Yu, and Z. Xie, ''Short-term wind speed or power forecasting with heteroscedastic support vector regression,'' *IEEE Trans. Sustain. Energy*, vol. 7, no. 1, pp. 241–249, Jan. 2016.

[19] Y. Ren, P. N. Suganthan, and N. Srikanth, ''A novel empirical mode decomposition with support vector regression for wind speed forecasting,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1793–1798, Aug. 2016.

[20] C. Guo, W. Ai, X. Zhang, Y. Guan, Y. Liu, S. Hu, and X. Zhao, ''Correction of sea surface wind speed based on SAR rainfall grade classification using convolutional neural network,'' *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 321–328, 2023.

[21] M. Li, Z. Zhang, T. Ji, and Q. H. Wu, ''Ultra-short term wind speed prediction using mathematical morphology decomposition and long short-term memory,'' *CSEE J. Power Energy Syst.*, vol. 6, no. 4, pp. 890–900, Dec. 2020.

[22] C. Jiang, Y. Mao, Y. Chai, M. Yu, and S. Tao, ''Scenario generation for wind power using improved generative adversarial networks,'' *IEEE Access*, vol. 6, pp. 62193–62203, 2018.

[23] W. Zhao, Y.-M. Wei, and Z. Su, ''One day ahead wind speed forecasting: A resampling-based approach,'' *Appl. Energy*, vol. 178, pp. 886–901, Sep. 2016.

[24] M. Mishra and M. Srivastava, ''A view of artificial neural network,'' in *Proc. Int. Conf. Adv. Eng. Technol. Res. (ICAETR)*, Aug. 2014, pp. 1–3.

[25] C. Cortes and V. Vapnik, ''Support-vector networks,'' *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[26] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, ''Supervised machine learning: A review of classification techniques,'' *Emerg. Artif. Intell. Appl. Comput. Eng.*, vol. 160, pp. 3–24, Oct. 2007.

[27] Y. LeCun, Y. Bengio, and G. Hinton, ''Deep learning,'' *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[28] K. S. Ni and T. Q. Nguyen, ''Image superresolution using support vector regression,'' *IEEE Trans. Image Process.*, vol. 16, no. 6, pp. 1596–1610, Jun. 2007.

[29] N. Aloysius and M. Geetha, ''A review on deep convolutional neural networks,'' in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2017, pp. 0588–0592.

[30] N. K. Chauhan and K. Singh, ''A review on conventional machine learning vs deep learning,'' in *Proc. Int. Conf. Comput., Power Commun. Technol. (GUCON)*, Sep. 2018, pp. 347–352.

[31] C. Lopez-Martin, S. Banitaan, A. Garcia-Floriano, and C. Yanez-Marquez, ''Support vector regression for predicting the enhancement duration of software projects,'' in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 562–567.

[32] R. Ramachandran, D. Rajeev, S. Krishnan, and P. Subathra, ''Deep learning—An overview,'' *Int. J. Appl. Eng. Res.*, vol. 10, no. 10, pp. 25433–25448, Jan. 2015.

[33] E. Grossi and M. Buscema, ''Introduction to artificial neural networks,'' *Eur. J. Gastroenterol. Hepatol.*, vol. 19, no. 12, pp. 1046–1054, 2007.

[34] (2023). *Historical Climate Data—Environment Canada*. [Online]. Available: https://climate.weather.gc.ca/historical_data/search_historic_data_e.html

[35] S. Mehtab and J. Sen, ''Stock price prediction using convolutional neural networks on a multivariate timeseries,'' 2020, *arXiv:2001.09769*.

[36] Y. Zhang, J. Liu, S. Liang, and M. Li, ''A new spatial–temporal depthwise separable convolutional fusion network for generating Landsat 8-day surface reflectance time series over forest regions,'' *Remote Sens.*, vol. 14, no. 9, p. 2199, May 2022.

[37] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, ''Deep learning for time series forecasting: A survey,'' *Big Data*, vol. 9, no. 1, pp. 3–21, Feb. 2021.

[38] A. Lawal, S. Rehman, L. M. Alhems, and Md. M. Alam, ''Wind speed prediction using hybrid 1D CNN and BLSTM network,'' *IEEE Access*, vol. 9, pp. 156672–156679, 2021.

[39] P. Prabha, V. Vanitha, and R. Resmi, ''Wind speed forecasting using long short term memory networks,'' in *Proc. 2nd Int. Conf. Intell. Comput., Instrum. Control Technol. (ICICICT)*, vol. 1, Jul. 2019, pp. 1310–1314.

[40] C. Huang, H. R. Karimi, P. Mei, D. Yang, and Q. Shi, ''Evolving long short-term memory neural network for wind speed forecasting,'' *Inf. Sci.*, vol. 632, pp. 390–410, Jun. 2023.

[41] M. Abbasipour, M. A. Igder, and X. Liang, ''A novel hybrid neural network-based day-ahead wind speed forecasting technique,'' *IEEE Access*, vol. 9, pp. 151142–151154, 2021.

[42] M. Khodayar, O. Kaynak, and M. E. Khodayar, ''Rough deep neural architecture for short-term wind speed forecasting,'' *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2770–2779, Dec. 2017.

[43] H. Jahangir, M. A. Golkar, F. Alhameli, A. Mazouz, A. Ahmadian, and A. Elkamel, ''Short-term wind speed forecasting framework based on stacked denoising auto-encoders with rough ANN,'' *Sustain. Energy Technol. Assessments*, vol. 38, Apr. 2020, Art. no. 100601.

[44] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' 2014, *arXiv:1412.6980*.

[45] M. Khodayar and J. Wang, ''Spatio-temporal graph deep neural network for short-term wind speed forecasting,'' *IEEE Trans. Sustain. Energy*, vol. 10, no. 2, pp. 670–681, Apr. 2019.

[46] D. S. Wilks, *Statistical Methods in the Atmospheric Sciences*. New York, NY, USA: Academic Press, 2011.

MOHAMMADHOSSEIN NAZEMI was born in Tehran, Iran, in September 1984. He received the B.Sc. degree in electrical engineering from the University of Tabriz, Tabriz, Iran, in 2008, and the M.Sc. degree in electrical engineering from Shahid Beheshti University (Abbaspour School of Engineering), Tehran, in 2020. He is currently pursuing the Ph.D. degree with the University of Saskatchewan, Saskatoon, SK, Canada. He has authored several journals and conference papers. His current research interests include power systems protection, electric machines fault detection, machine learning and deep learning, digital signal processing, statistical performance analysis, estimation, and classification.

SHAIKAT CHOWDHURY was born in Chattogram, Bangladesh. He received the B.Sc. degree in electrical engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2017. He is currently pursuing the master's degree with the University of Saskatchewan, Saskatoon, SK, Canada. From May 2017 to January 2018, he was a Lecturer with East Delta University, Chattogram. In February 2018, he joined North West Power Generation Company Ltd., Sirajgong, Bangladesh, as an Assistant Engineer. From March 2019 to August 2022, he was with the Bangladesh Power Development Board as an Assistant Engineer. His current research interests include renewable energy, power systems, machine learning, and power electronics.

XIAODONG LIANG (Senior Member, IEEE) was born in Lingyuan, Liaoning, China. She received the B.Eng. and M.Eng. degrees in electrical engineering from Shenyang Polytechnic University, Shenyang, China, in 1992 and 1995, respectively, the M.Sc. degree from the University of Saskatchewan, Saskatoon, Canada, in 2004, and the Ph.D. degree in electrical engineering from the University of Alberta, Edmonton, Canada, in 2013.

From 1995 to 1999, she was a Lecturer with Northeastern University, Shenyang, China. In October 2001, she joined Schlumberger (SLB), Edmonton, and was promoted to a Principal Power Systems Engineer with this world's leading oil field service company, in 2009. She was with Schlumberger for almost 12 years, until August 2013. From 2013 to 2019, she was with Washington State University, Vancouver, WA, USA, and Memorial University of Newfoundland, St. John's, NL, Canada, as an Assistant Professor and later an Associate Professor. In July 2019, she joined the University of Saskatchewan, where she is currently a Professor and the Canada Research Chair in Technology Solutions for Energy Security in Remote, Northern, and Indigenous Communities. She was an Adjunct Professor with the Memorial University of Newfoundland, from 2019 to 2022. Her current research interests include power systems, renewable energy, and electric machines.

Dr. Liang is a Registered Professional Engineer of Saskatchewan Province, Canada, a fellow of IET, and the Deputy Editor-in-Chief of IEEE Transactions on Industry Applications.