



Spatio-Temporal Graph Deep Neural Network for Short-Term Wind Speed Forecasting

Mahdi Khodayar , *Student Member, IEEE*, and Jianhui Wang , *Senior Member, IEEE*

Abstract—Wind speed forecasting is still a challenge due to the stochastic and highly varying characteristics of wind. In this paper, a graph deep learning model is proposed to learn the powerful spatio-temporal features from the wind speed and wind direction data in neighboring wind farms. The underlying wind farms are modeled by an undirected graph, where each node corresponds to a wind site. For each node, temporal features are extracted using a long short-term memory network. A scalable graph convolutional deep learning architecture (GCDLA), motivated by the localized first-order approximation of spectral graph convolutions, leverages the extracted temporal features to forecast the wind-speed time series of the whole graph nodes. The proposed GCDLA captures spatial wind features as well as deep temporal features of the wind data at each wind site. To further improve the prediction accuracy and capture robust latent representations, the rough set theory is incorporated with the proposed graph deep network by introducing upper and lower bound parameter approximations in the model. Simulation results show the advantages of capturing deep spatial and temporal interval features in the proposed framework compared to the state-of-the-art deep learning models as well as shallow architectures in the recent literature.

Index Terms—Deep learning, wind speed forecasting, spatio-temporal features, long short-term memory, graph convolutional network, rough set theory.

I. INTRODUCTION

WIND ENERGY is known as a clean source of renewable energy with no pollution in its application [1]. The high variability of this renewable resource could cause considerable power fluctuation and instability in the power networks. Insufficiently accurate wind forecasts may lead to instability of the power network and eventually jeopardize the reliability and security of the power networks [2]. Wind speed forecasting is a crucial step to find the most economical solution for the operation of the power grid [3]. The energy produced by a wind generator is directly determined by the wind speed; therefore, accurate and efficient wind speed forecasting models would greatly benefit the wind energy prediction algorithms and consequently improves the reliability and security of the power grid [4], [5].

Manuscript received October 3, 2017; revised February 2, 2018; accepted April 3, 2018. Date of publication June 4, 2018; date of current version March 21, 2019. Paper no. TSTE-00904-2017. (Corresponding author: Mahdi Khodayar.)

The authors are with the Department of Electrical Engineering, Southern Methodist University, Dallas, TX 75205 USA (e-mail: mahdik@smu.edu; Jianhui@smu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSTE.2018.2844102

Since wind speed time series are generally chaotic and have stochastic characteristics, wind prediction is generally considered as a complex regression task [6]. In addition, wind prediction accuracy is heavily dependent on the generalization capability of the forecasting model; therefore, highly varying wind data requires complex nonlinear approaches for the forecasting tasks. In recent years, wind speed and wind power prediction approaches are mainly categorized with respect to the time horizon of the application [7]. These methods are mainly divided into three categories: 1) short-term wind prediction estimates the wind data of several minutes to hours ahead. This type of prediction benefits electricity market clearing, real-time grid operations and regulation, economic dispatch, and operational security in the electricity market [8]. 2) Medium-term forecasting focuses on time intervals in the range of several hours to a week. Unit commitment and reserve allocation decisions are made using this type of predictions. 3) Long-term forecasting refers to the prediction of wind velocity in the ranges of one week to a year or more. Maintenance, expansion planning, and the feasibility studies for the design of wind farms are dependent on this task [9].

Generally, there are four types of wind speed forecasting models proposed in the recent literature: 1) Persistence methods, which assume the wind data at a future time step has the same values as the current time instance. This assumption is similar to the smoothness assumption known in Machine Learning, where a function approximation model outputs the same results for two inputs close to each other in the input space. Although this assumption diminishes the future wind speed estimation precision for long time horizons, persistence models are accurate predictors for ultra-short horizons. 2) Physical methods, which are based on lower atmosphere or numerical weather prediction (NWP) that uses weather prediction data such as temperature, pressure, surface roughness, and obstacles. NWPs are generally executed on supercomputers because of their high computational complexity. The computation burden limits the usefulness of such methods for medium and long-term tasks [10]. 3) Statistical methods, which are low-cost and easily applicable techniques to capture the mathematical relationships in the time series data. The prediction accuracy of such methods declines with the increase in the length of the prediction time horizon. Statistical approaches include autoregressive (AR), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA) models. In [11], a hybrid ARMA model with wavelet transformation is introduced for short-term predictions. The research work presented in [12] proposes

several ARMA-based approaches for wind speed and wind direction data prediction. An ARIMA with a regression analysis approach is applied to real-life short-term wind forecasting in [13]. In [14], a robust combination of ARIMA and Support Vector Machines (SVMs) is designed for short-term prediction of wind speed data. The proposed approach has higher accuracy compared to the conventional ARIMA. 4) Artificial Intelligence (AI) methods apply Machine Learning and Pattern Recognition techniques such as Artificial Neural Networks (ANNs) and SVMs. A Support Vector Regression (SVR) framework is employed for short-term wind energy predictions in [15]. The capability of SVR-based forecast on the micro-scale level for one wind generation unit as well as the larger scale of whole wind park is highlighted. In [16], an optimal loss function is derived for heteroscedastic regression, and a novel SVR framework is developed for short-term wind forecasting based on learning tasks of Gaussian noise with heteroscedasticity. Most recently proposed ANN-based approaches design and learn shallow neural architectures with few levels of abstraction, such as Feed-forward ANN [17]–[18], recurrent ANN [19], RBF network [20]–[21], ridgelet ANN [22] and adaptive wavelet ANN [23]. In recent years, Deep ANNs have been introduced in the literature resulting in an improvement in the accuracy of time series predictions compared to shallow prediction models. In [24], a Deep Boltzmann Machine (DBM) is trained using a contrastive divergence algorithm for multi-period wind forecasting. The method leverages Gibbs sampling to train Restricted Boltzmann Machines (RBMs). Stacked Auto-encoder (SAE) models with greedy unsupervised feature learning methods are applied for short-term wind prediction in [25] and [26]. Simulation results show that increasing the number of hidden layers can improve the forecasting accuracy as long as the model does not encounter the overfitting problem. Such prediction results motivate wind prediction methodologies to capture high levels of abstractions in data to improve accuracy.

Although AI approaches have recently obtained better prediction results compared to statistical models, these methods have several drawbacks: 1) Shallow ANN models have weak generalization capability; thus, they are not able to effectively learn complex patterns from the highly varying wind data. Although wind time series datasets contain large amounts of data, these models cannot learn complex data abstractions. SVM-based approaches, as well as linear models, have the same problem because of the smoothness assumption. 2) Deep neural architectures proposed in the literature, i.e., DBM and SAE, do not consider the sequential characteristics of the wind data. 3) The ANN-based approaches merely capture wind patterns in the time-space and leverage temporal correlations to learn the temporal features from the time series data. It is shown in the literature that neighboring wind sites have high spatial correlations [27]. However, shallow ANNs are unable to efficiently capture spatial wind features, since high dimensional input vector requires a large number of free parameters that cannot be tuned efficiently in a shallow model using the gradient of the error function.

As there exists a strong correlation between wind sites located in a vicinity, spatio-temporal forecasting methodologies utilize

information collected from neighboring sites in order to improve the prediction accuracy of target sites. The Regime Switching Space-Time Diurnal method that leverages the off-site predictors to provide predictive cumulative distribution functions is presented in [28]. A multi-channel adaptive filter is used to forecast wind speed and direction using spatial correlations at multiple sites in [29]. Markov chain-based statistical methods that leverage graphical spatio-temporal learning-based model and statistical characteristics of aggregated wind generation are presented in [30] to forecast wind generation output. In this paper, a graph deep learning neural architecture is proposed to capture powerful spatio-temporal features of wind speed and wind direction in multiple neighboring wind sites. An undirected graph is designed using the wind speed and direction mutual information at each wind site with the rest of the wind sites. Temporal wind speed features are extracted using a Long Short-Term Memory (LSTM) network for each wind site [31]. A graph convolutional deep learning architecture (GCDLA) is further designed based on the graph theory, convolutional neural networks, and Rough Set Theory. The proposed architecture is based on the localized first-order approximation of spectral graph convolutions [32]. A convolutional network is learned using the temporal LSTM features of the graph nodes; thus, spatial features of the whole wind sites, i.e., graph nodes, are captured in an end-to-end fashion. In order to learn robust spatial and temporal features, Rough Set Theory [33] is incorporated into the proposed graph learning model to capture the interval features from the wind data. The extracted interval rough spatio-temporal features obtained from the GCDLA are used to tune the whole model in an end-to-end supervised manner using the gradient information of the regression loss. The contributions of this paper are as follows:

- 1) A novel graph-based convolution network is proposed for the problem of spatio-temporal wind speed prediction. Our proposed neural architecture learns spatial features with a high level of abstraction obtained from multiple wind farms. To the best of our knowledge, this is the first work that introduces a spatio-temporal deep learning methodology in time series prediction.
- 2) The proposed framework can better learn deep temporal features from the wind speed sequential data compared to DBN [24] and SAE [25], because the sequential nature of the data is modeled using recurrent LSTM units instead of having generative RBMs or discriminative auto-encoders for temporal feature extraction. Comparison of our method with DBN and SAE are provided in simulations.
- 3) The proposed GCDLA can better handle noise and uncertainties in the input data. The proposed architecture is further enhanced using the Rough Set Theory to capture interval deep features from the wind data. This is the first effort to incorporate the Rough Set Theory with a graph learning methodology.
- 4) As shown in simulation results, the proposed deep network can be trained in a semi-supervised learning setting. Previous spatio-temporal learning models in the literature including [34]–[36] should apply the supervision

information (i.e., the actual future wind speed values available in historical data) of all wind sites to capture wind data patterns; however, our proposed methodology can learn meaningful spatio-temporal features for a set of wind farms by merely observing a small subset of the underlying wind sites. The proposed network is scalable with respect to the number of observations; hence, the computational time can be dramatically reduced.

The rest of the paper is organized as follows, Section II describes the problem formulation of spatio-temporal wind prediction. In Section III, the proposed graph deep learning model is explained. The temporal and spatial feature learning as well as the interval feature extraction by applying Rough Set Theory are discussed in this section. The numerical results are discussed in Section IV. The conclusions are presented in Section V.

II. PROBLEM FORMULATION

The objective of the spatio-temporal wind forecasting is to capture the relations between the historical wind data of surrounding wind farms and the future values of the target wind farm. Suppose there exists a set of N wind farms in a region. Each wind farm i has wind speed and wind direction measurements denoted by v_t^i and r_t^i for time step t , respectively. The distance between the i th and j th wind farms is given by d_{ij} and the symmetric distance matrix D contains elements d_{ij} for all $1 \leq i, j \leq N$. The wind speed data of the whole set of wind farms at time t is denoted by $V_t = \langle v_t^1, v_t^2, \dots, v_t^N \rangle$ and the corresponding wind direction vector is $R_t = \langle r_t^1, r_t^2, \dots, r_t^N \rangle$. The proposed framework for time series forecasting is considered as a combination of functions F and T_i . Here, T_i is a nonlinear function that maps the wind speed values of previous time steps at the wind farm i to the temporal feature space. This mapping is obtained by the LSTM network corresponding to the i th wind farm. Let us denote the number of time steps of the historical time series applied for the prediction task by L . The proposed model learns a mapping F from the space of T_i 's to the space of V_{t+K} where K is the length of the forecasting horizon. Thus, the prediction framework has the following form:

$$\begin{aligned} &F(T_1(v_{t-L+1}^1, v_{t-L+2}^1, \dots, v_t^1; \theta_1), \\ &T_2(v_{t-L+1}^2, v_{t-L+2}^2, \dots, v_t^2; \theta_2), \dots, \\ &T_N(v_{t-L+1}^N, v_{t-L+2}^N, \dots, v_t^N; \theta_N); \zeta) = \hat{V}_{t+K} \end{aligned} \quad (1)$$

Here, \hat{V}_{t+K} is the estimation obtained for the vector of actual wind speed values V_{t+K} . θ_i is the set of LSTM parameters corresponding to the i th wind farm. One advantage of this formulation is that the LSTM parameters $\Theta = \{\theta_1, \theta_2, \dots, \theta_N\}$ can be shared in order to speed up the training procedure. ζ is the set of free parameters in the proposed graph deep learning model which contains a convolutional network for capturing spatial features using temporal features, i.e., T_i 's.

III. PROPOSED METHODOLOGY

In this section, the extraction of interval spatio-temporal deep features of the proposed graph feature learning method for

short-term wind prediction is explained. First, the graph construction for modeling the set of underlying wind farms is discussed. Then, the temporal feature extraction is described using Long Short-Term Memory networks, and the rough spatial feature learning using convolution neural networks is explained in detail.

A. Graph Modeling

Wind speed and direction time series obtained from surrounding wind farms in a region have high spatial correlations, and a graph structure is designed to leverage such relationships. In order to model the underlying wind farms, a graph $G = (V, E)$ is assumed. Here, V is the set of N nodes, each corresponding to a wind farm, and E is the set of edges.

A node i is connected to some node j if and only if i and j are neighbors. The neighborhood relationship is defined using the mutual information of i and j in the historical data. Suppose for wind farm i , there exists H historical wind speed time series values $v^i(t)$ and wind direction $r^i(t)$ with $t \in [1, H]$. For each wind farm i , a two-dimensional historical signal $h_{2 \times H}^i$ is obtained that contains wind speed and direction records. These measurements are considered for all N neighboring wind sites. In order to define E for each pair of nodes i and j , the mutual information between h^i and h^j , $MI(h^i, h^j)$ is computed. Similar to the correlation computation method in [25] and [26], first the mutual information is computed for the speed and direction time series inside h^i and h^j separately. Here, the mutual information $MI(h^i, h^j)$ represents the average of wind speed and direction correlation values corresponding to nodes i and j . If $MI(h^i, h^j) > \tau$ for some $\tau > 0$, then $(i, j) \in E$; otherwise, there is no connection between two nodes. The adjacency matrix A is a symmetric square matrix of size N , in which each element $A(i, j)$ reflects the closeness of two wind farms i and j . The adjacency matrix is defined as the following:

$$A(i, j) = \begin{cases} e^{-d_{ij}} & \text{if } MI(h^i, h^j) > \tau \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

B. Temporal Feature Learning

Assume the wind speed values corresponding to time steps $\tilde{t} \leq t$ are available at time step t to tune a regression model for estimating the target function V_{t+K} with forecasting horizon $K > 0$. An L -length time window $[t-L+1, t-1, t]$ is considered for the wind speed time series to capture the temporal wind data features. For wind farm i , temporal features are extracted from the input vector to learn the mapping $T_i(v_{t-L+1}^i, v_{t-L+2}^i, \dots, v_t^i; \theta_i)$ by tuning parameters θ_i . In the proposed framework, a supervised recurrent LSTM network model [31] is applied to learn deep temporal features θ_i in each graph node $i \in V$. Unlike earlier proposed state-of-the-art deep architectures that applied DBN [24] and SAE [26] for temporal feature extraction, the proposed model captures the sequential nature of the wind time series.

The LSTM network consists of memory blocks that contain memory cells with self-connections that are able to store the temporal state of the model. In addition to memory units, the

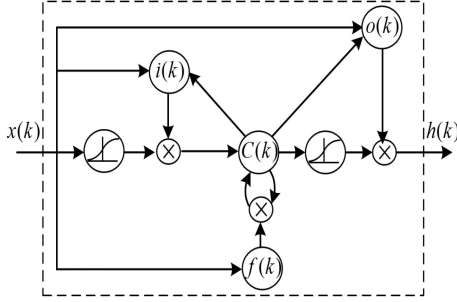


Fig. 1. Structure of the LSTM memory block.

LSTM block has the following special multiplicative computational units: 1) input gate that controls the input activations fed to the memory cell; 2) output gate that controls the output flow of the cell activations into the subsequent memory block; 3) forget gate that scales the corresponding cell's temporal state before adding it back to the cell as an input signal applying a self-recurrent connection. This gate can control and reset the memory content adaptively. Fig. 1 shows a memory block. Here, $x(k)$ is the input at time step k and $h(k)$ is the output of the previous block. $i(k)$, $f(k)$, and $o(k)$ are the signals corresponding to the input, forget, and output gates, respectively. The gates' output signals are computed by:

$$\begin{aligned} i(k) &= g(W_i x(k) + U_i h(k-1) + b_i) \\ f(k) &= g(W_f x(k) + U_f h(k-1) + b_f) \\ o(k) &= g(W_o x(k) + U_o h(k-1) + b_o) \end{aligned} \quad (3)$$

where $g(\cdot)$ is the nonlinear activation function of the input, output, and forget signals. In this paper, a sigmoid unit is considered for this function and the parameter space consists of a set of tunable weight matrices $Q = \{W_i, U_i, W_f, U_f, W_o, U_o\}$ and bias vectors $B = \{b_i, b_f, b_o\}$.

The new state of the memory cell $C(k)$ is computed by partially forgetting the existing memory content $C(k-1)$ and adding the new memory content $\tilde{C}(k)$:

$$\begin{aligned} \tilde{C}(k) &= \tilde{g}(W_c x(k) + U_c h(k-1) + b_c) \\ C(k) &= f(k) \cdot C(k-1) + i(k) \cdot \tilde{C}(k) \end{aligned} \quad (4)$$

Here, $\bar{Q} = \{W_c, U_c, b_c\}$ are the free parameters of the memory cell. A hyperbolic tangent function is utilized for the memory cell state activation \tilde{g} . In the LSTM training process associated with each node i , the parameters tuned for the corresponding wind farm are $\theta_i = Q \cup B \cup \bar{Q}$. Here, $x = v_k^i$, $k \in [t-L+1, t]$ corresponding to each time step k is considered as the input for our temporal feature learning procedure. The temporal model contains L memory blocks, and the hidden state of the last block, $h(L) = T_i$, is the temporal data representation for the i th wind site.

C. Spectral Graph Convolutions

The spectral convolutions [37] on G can be written as the multiplication of temporal features of all N nodes in V , that is $F = \langle T_1, T_2, \dots, T_N \rangle \in \mathbb{R}^N$, with a filter $\psi_\theta = \text{diag}(\theta)$

where $\theta \in \mathbb{R}^N$ is the vector of filter parameters in the Fourier domain. This convolutional operation can be written as:

$$\psi_\theta * F = U \psi(\theta) U^T F \quad (5)$$

where U is the eigenvector matrix for the normalized graph Laplacian $L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Gamma U^T$ with Γ as a diagonal matrix which contains the eigenvalues of L . Here, I is the identity matrix, A is the adjacency matrix of G , and $D_{ii} = \sum_j A_{ij}$. Using this notation, the graph Fourier transform of F is $U^T F$. The filter ψ_θ can be represented as a function of the eigenvalues of the normalized Laplacian; thus, the filter can be denoted by $\psi_\theta(\Gamma)$. Since the matrix multiplication of the eigenvector matrix U in (5) has $O(N^2)$ time complexity and the Eigen decomposition of L can result in very expensive computations [32], $\psi_\theta(\Gamma)$ is estimated using Chebyshev polynomials [37] $P_j(x)$:

$$\psi_\omega(\Gamma) \approx \sum_{j=0}^J \omega_j P_j \left(\frac{2}{\lambda_{\max}} \Gamma - I \right) \quad (6)$$

where λ_{\max} is the maximum eigenvalue in L and $\omega \in \mathbb{R}^J$ is the Chebyshev coefficients. The Chebyshev polynomials are computed recursively by:

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_{j+1}(x) &= 2xP_j(x) - P_{j-1}(x) \end{aligned} \quad (7)$$

Using (5), the convolution of temporal features extracted by the LSTM networks, i.e., $\psi_\theta * F$, can be rewritten as:

$$\psi_\omega * F \approx \sum_{j=0}^J \omega_j P_j \left(\frac{2}{\lambda_{\max}} L - I \right) F \quad (8)$$

This equation is a polynomial with order J since the approximation contains J Chebyshev terms. Therefore, (8) is a J -localized polynomial; that is, only the temporal features extracted from the nodes that are at most J steps away from a central node are applied in the convolution of G . Using (8) a convolutional network can be defined to learn from the graph data applying convolution operations on the nodes in set V .

D. Rough Set Theory

Rough Set Theory is a mathematical tool introduced by Pawlak [33] to address uncertain and inexact knowledge. $S = \langle U, A, V, f \rangle$ is a 4-length tuple representing an information system. Here, U is a finite non-empty set which is the universe of primitive objects and A is the non-empty set of attributes. Every attribute $a \in A$ has a domain V_a and V is the union of all attribute domains in S . Using the above notations, $f : U \times A \rightarrow V$ is defined as the information function; thus, $\forall u \in U \forall a \in A : f(u, a) \in V_a$.

Let us assume $M \subseteq A$ is a set of attributes, the objects u_1 and u_2 are indiscernible from each other in the information system S , if and only if $\forall m \in M : f(u_1, m) = f(u_2, m)$. For any concept set $X \subseteq U$ and attribute set M , the Rough Set Theory defines two types of estimations for X using the knowledge in M ;

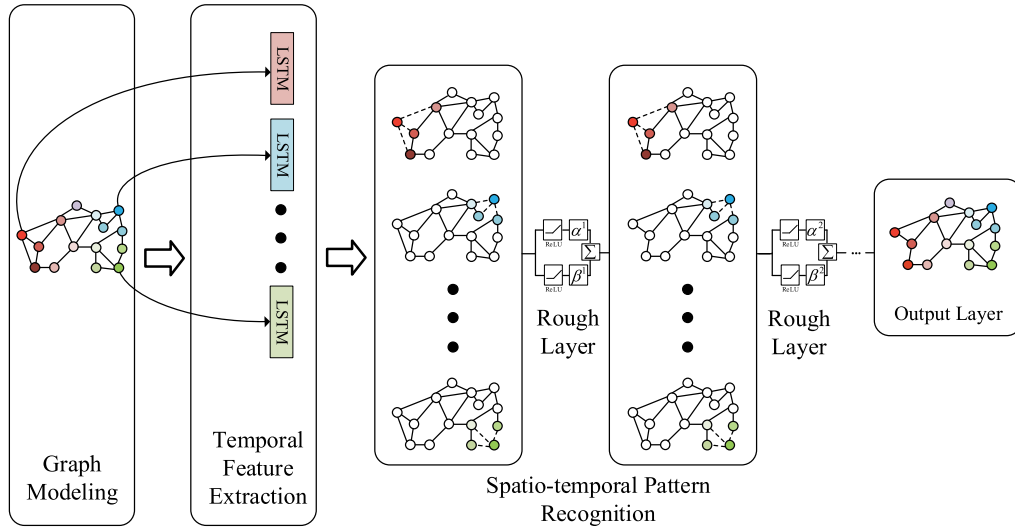


Fig. 2. Structure of graph convolutional deep learning architecture with rough feature extraction.

the M-upper bound estimation \overline{MX} and the M-lower bound estimation \underline{MX} . These estimations are computed by:

$$\begin{aligned}\overline{MX} &= \cup\{O \in U|M : O \subseteq X\} \\ \underline{MX} &= \cup\{O \in U|M : O \subseteq X\}\end{aligned}\quad (9)$$

The concept set X has also an M-boundary region computed by:

$$BND_M(X) = \overline{MX} - \underline{MX} \quad (10)$$

\underline{MX} represents the set of all objects in the universe of primitive objects that can be surely labeled as a member of X using the knowledge of attributes in M . The upper-bound estimation \overline{MX} shows the set of objects that can possibly be labeled as a member of X and the M-boundary contains the objects that cannot be certainly labeled as a member of the concept set. If $\overline{MX} = \underline{MX}$, that is $BND_M(X) = \emptyset$, then X is called a crisp (exact) set with respect to the attribute set M ; otherwise, X is a rough set.

E. Graph Convolutional Deep Learning Architecture

Fig. 2 shows the structure of the proposed deep spatio-temporal feature learning model on a graph of wind farms. As presented in the graph model, the wind farm pairs with highly correlated wind speed and direction time series are formed as members of E , and the adjacency matrix A is computed using (2).

A neural network is designed based on the graph theory and graph convolutions. The proposed neural network is incorporated into the Rough Set Theory in order to extract meaningful interval spatio-temporal features from the underlying wind data. The proposed Graph Convolutional Deep Learning Architecture consists of several convolutional layers of the form of (8) stacked together with interval rough feature learning layers that apply Leaky Rectified Linear Unit (ReLU) activations. The

ReLU activation for an input x is defined as:

$$f(x) = \begin{cases} \max(x, 0) & \text{if } x > 0 \\ mx & \text{otherwise} \end{cases} \quad (11)$$

where $m \leq 1$ is the leakage coefficient that makes the parameters tunable using the gradient of the supervised error function even if the input value is negative. In contrast to sigmoidal activations, this activation does not saturate for large input values; thus, the derivative of (10) is non-zero for small values; hence, the model can be tuned more effectively compared to sigmoidal networks.

Limiting the layer-wise convolution operation to $J = 1$ in (8) results in a linear function with respect to the normalized graph Laplacian L . Having such a linear formulation, the maximum eigenvalue of L can be estimated by $\lambda_{\max} = 2$ since the neural network can easily adapt to this change when trained on the training samples. Having these approximations, (8) can be written as:

$$\begin{aligned}\psi_\omega * F &\approx \omega_0 P_0(L - I)F + \omega_1 P_1(L - I)F \\ &= \omega_0 F + \omega_1 (L - I)F \\ &= \omega_0 F - \omega_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} F\end{aligned} \quad (12)$$

Here, ω_0 and ω_1 are two tunable filter parameters. These parameters are shared among all nodes. In order to avoid the overfitting problem, ω_0 and ω_1 are replaced by a single filter parameter $\varpi = \omega_0 = -\omega_1$; therefore, (12) can be computed by:

$$\psi_\omega * F \approx \varpi (I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) F \quad (13)$$

In order to deal with the uncertainties in the wind data, we are motivated to incorporate the Rough Set Theory with the deep spatio-temporal feature extractor. Interval filter parameters, ϖ_U and ϖ_L , are introduced as the upper- and lower-bound parameters, respectively. The proposed spatio-temporal feature learning model with rough pattern recognition, has an upper-bound

and lower-bound approximation for (13) computed as:

$$\begin{aligned} O_{\max} &= \max(\varpi_U(I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})F, \varpi_L(I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})F) \\ O_{\min} &= \min(\varpi_U(I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})F, \varpi_L(I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})F) \end{aligned} \quad (14)$$

The total output of the k th layer of the proposed spatio-temporal feature extraction model is calculated by: $O^k = \alpha^k O_{\max}^k + \beta^k O_{\min}^k$ where $0 \leq \alpha^k$ and $\beta^k \leq 1$ are the rough coefficients that set the contributions of the upper-bound and lower-bound activations to the latent representation. O_{\max}^k and O_{\min}^k are the rough approximations of O^k . Applying the renormalization trick [37] to address numerical instabilities and exploding/vanishing gradients of the supervised error signal, the upper-bound approximation for the k th layer is rewritten as:

$$O_{\max}^k = \max(\varpi_U M O^{k-1}, \varpi_L M O^{k-1}) \quad (15)$$

where $M = \tilde{D}^{-\frac{1}{2}}(A + I)\tilde{D}^{-\frac{1}{2}}$, and \tilde{D} is defined as $\tilde{D}_{ii} = \sum_j (A + I)_{ij}$. Similar formulation can be written to compute O_{\min}^k , the lower-bound approximation of O^k .

Assume that the output signal obtained from the LSTM network is $F = \langle T_1, T_2, \dots, T_N \rangle \in \mathbb{R}^N$, the forward propagation of the proposed spatio-temporal deep network with L layers is computed as:

$$\begin{aligned} F^l &= \alpha^l \max(f(MF^{l-1}W_U^l, MF^{l-1}W_L^l)) \\ &+ \beta^l \min(f(MF^{l-1}W_U^l, MF^{l-1}W_L^l)) \quad 1 \leq l \leq L \end{aligned} \quad (16)$$

where $F^0 = F$ is the temporal feature vector obtained from the LSTM networks, $Y = F^L$ is the output of the graph convolutional deep learning architecture, M is a pre-processed matrix computed in (15), W_U^l and W_L^l are the upper-bound and lower-bound rough parameters in layer l .

The proposed deep learning model can be trained in an end-to-end manner using the gradients of the supervised squared error function $\frac{1}{K} \sum_{k=1}^K (O(k) - T(k))^2$ where K is the total number of training samples available in the training wind dataset, $O(k)$ is the output of GCDLA for the k th sample, and $T(k)$ is its corresponding desired output which is the actual wind speed value vector for all the nodes in V .

IV. NUMERICAL RESULTS

A. Dataset

In this case study, 145 wind sites located in the northern states of the US are considered to train and evaluate the proposed prediction model. The data is obtained from the Eastern Wind Integration Dataset [38] that consists of wind speed for 1,326 simulated wind generation sites. Fig. 3 shows the locations of the wind sites under study. The dataset consists of 6 years of wind speed and wind direction measurements from 2007 to 2012 with 5-min intervals. Each year contains 105,120 data samples. The 2007-2011 datasets are considered for graph modeling and supervised training of GCDLA as well as model validation. The validation set consists of 20% of the available data. The 2012 wind data is used to evaluate the efficiency of the proposed ap-

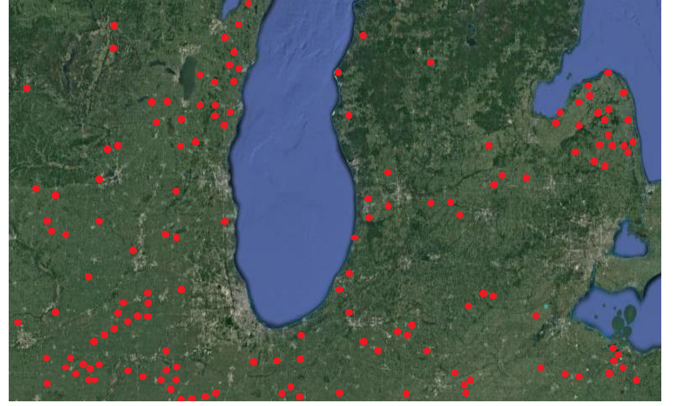


Fig. 3. Locations of the underlying wind sites obtained from the eastern wind dataset.

proach by comparing the performance of the proposed GCDLA model to other recently proposed deep learning and shallow feature extraction architectures. The wind speed time series are forecasted for a certain horizon; however, the time step in which the forecasting is updated is 5-minutes. This means that every 5-minutes the wind speed forecast is updated for the rolling forecasting horizon.

B. Evaluation Metrics

The root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are calculated as three evaluation metrics to assess the forecasting performance:

$$\begin{aligned} RMSE &= \frac{1}{N} \sum_{n=1}^N \left(\sqrt{\frac{1}{M} \sum_{m=1}^M e_{nm}^2} \right) \\ MAE &= \frac{1}{N} \sum_{n=1}^N \left(\frac{1}{M} \sum_{m=1}^M |e_{nm}| \right) \\ MAPE &= \frac{100}{N} \sum_{n=1}^N \left(\frac{1}{M} \sum_{m=1}^M \left| \frac{e_{nm}}{a} \right| \right) \end{aligned} \quad (17)$$

Here, e_{nm} is the prediction error of the n th wind site for the m th test sample. The total number of test samples is denoted by M and a represents the actual wind data.

C. Implementation Details

The proposed GCDLA model is implemented using TensorFlow [39] with GPU acceleration to speed up the training process. The model is implemented on a multi-GPU computer system with two NVIDIA GTX980 graphics cards and a 3.2 GHz CPU.

In the graph modeling procedure, the mutual information threshold is set to $\tau = 0.4$. The time window length L for the LSTM's input is chosen from the set $\Omega = \{20, 25, 30, 35, 40, 45, 50\}$. Batch size of the training procedure can range from 40 to 100 with 10 as a gap. The learning rate of the batch gradient descent algorithm is set to 10^{-3} and the number of training epochs is 80. The number of hidden layers

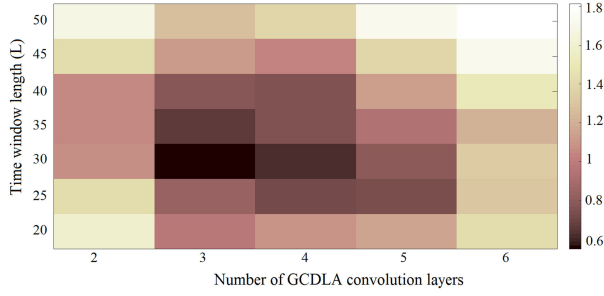


Fig. 4. Validation RMSE of the spatiotemporal model using different configurations.

k in the spatio-temporal feature learning model can range from 2 to 6. The best configuration is chosen after running the model on the validation set for 100 times. Fig. 4 shows the validation RMSE of hourly wind speed forecasting for various settings with the optimal batch size of 80. As shown in this figure, the RMSE is significantly decreased when the number of hidden layers is increased; however, large numbers of layers can lead to a noticeable decline in the performance due to overfitting on the training samples.

In this study, the optimal GCDLA architecture with respect to the validation accuracy consists of three convolutional layers with a time window length of 30 steps. The Chebyshev polynomial order of this architecture is $J = 2$; hence, the temporal features extracted from the nodes that are at most two steps away from a central node are applied in the convolution of G in (8). The maximum eigenvalue of L for the corresponding model is $\lambda_{\max} = 2$ and the optimal time window length is 45 steps.

D. Experimental Settings

Two experimental settings, supervised and semi-supervised, are conducted to evaluate the performance of the proposed methodology:

- 1) **Supervised Learning:** In this experiment, the true labels (actual wind speed values) of all nodes in G are observed in order to train GCDLA. Here, the network utilizes the supervision information obtained from all nodes to estimate the speed values of all input nodes simultaneously.
- 2) **Semi-Supervised Learning:** The true labels of the regression problem, i.e., actual wind speed values of each node, can be propagated using the proposed filters in (12). Hence, GCDLA can capture spatio-temporal patterns in a semi-supervised fashion on any graph-structured data. GCDLA only requires observing labels of a small portion of nodes in G . This method propagates the captured information of the wind data patterns in the supervised nodes through the edges of our graphical model to label both observed and unobserved nodes.

E. Supervised Learning

In this study, the proposed GCDLA is trained using the data obtained from all nodes in the modeled graph. The input size of neural networks and the number of nonlinear mappings (hidden

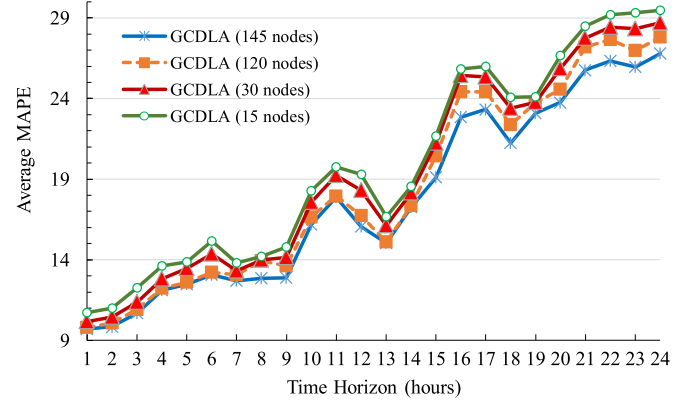


Fig. 5. Average RMSE (m/s) of GCDLA using various number of supervised input nodes.

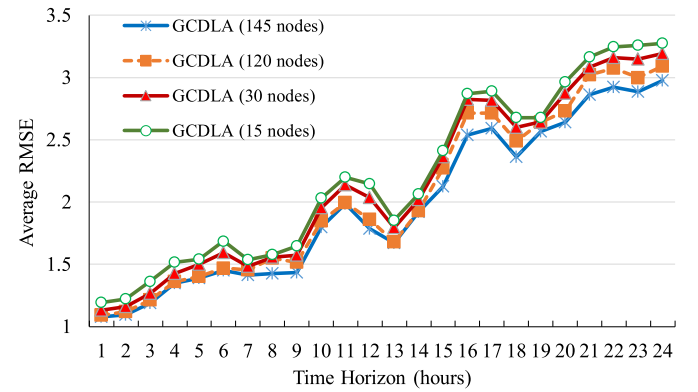


Fig. 6. Average MAPE of GCDLA using various number of supervised input nodes.

layers) in the architecture are hyper-parameters of GCDLA. As discussed in Section IV. C, the number of hidden layers is selected according to the performance of GCDLA on the validation set. In order to show the effect of the number of input wind sites (graph size) on GCDLA's performance, we randomly select $N \in \{15, 30, 120, 145\}$ wind sites to train and evaluate GCDLA for 100 runs. In each run, the proposed methodology is trained and evaluated on the wind data corresponding to a subset of nodes in graph G . Each run consists of an independent validation to find the optimal set of parameters.

Figs. 5 and 6 show the average RMSE and MAPE reported for various time horizons, respectively. When the number of supervised observable nodes is increased, the model obtains more accurate spatial information from the region of interest; hence, the performance is enhanced. For instance, when 6-hr ahead prediction is considered, GCDLA with 15 observable nodes has a MAPE of 15.17, while this error reaches to 13.20 and 13.05 for 120 and 145 supervised nodes, respectively. Moreover, for 6-hr ahead prediction task, the RMSE of the model decreases by 13.09% as the input size is increased from 15 nodes to 145 nodes. This improvement is more considerable for larger time horizons as the prediction will require more supervised information when the problem complexity grows. Although having larger modeled graphs helps GCDLA to capture more powerful deep

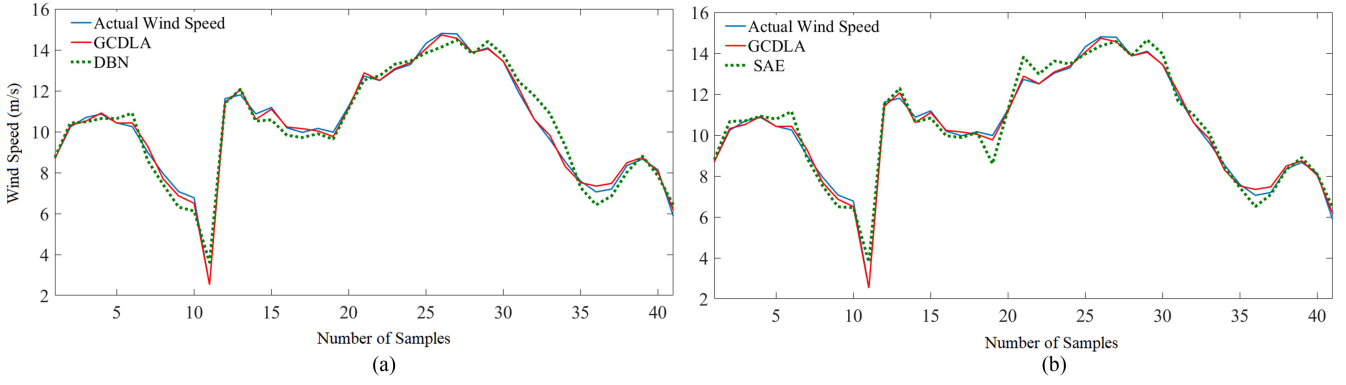


Fig. 7. Output of GCDLA, DBN, and SAE with the actual wind speed values for 40 samples of May 25th 2012. (a) Comparison with DBN. (b) Comparison with SAE.

spatio-temporal features, smaller architectures still have high prediction accuracy compared to the state-of-the-art deep learning models. Fig. 7 shows the hourly forecasting performance of GCDLA with only 15 input nodes on 40 test samples of May 25th 2012 compared to recently proposed deep learning models, i.e., Deep Belief Network (DBN) [24] and Stacked Auto-encoder (SAE) [25], [26]. As shown in Fig. 7(a) and (b), GCDLA follows the actual wind speed values with higher accuracy compared to DBN and SAE. The largest absolute prediction error of GCDLA in this period is 0.29 m/s while applying DBN and AE leads to 1.09 m/s and 1.38 m/s errors, respectively. Although both DBN and GCDLA capture complex nonlinear features with high abstractions, GCDLA leads to noticeably higher accuracy, especially when the prediction time horizon is expanded. As shown in Fig. 7(b), GCDLA is able to accurately estimate the peaks while the error rate is increased at peaks in AE when the wind data has relatively high variations.

F. Comparison With State-of-the-art Methodologies

In order to further investigate the prediction performance of our methodology, the proposed deep spatio-temporal feature learning model is compared with recently proposed shallow and deep feature extraction techniques. Single-model and hybrid approaches presented below are used as benchmarks to highlight the performance of the GCDLA.

1) *Single-Model Methodologies*: In this study, shallow architectures including feed-forward neural network (FFNN) [17], [18], time delay neural network (TDNN), and nonlinear autoregressive neural network (NARNN) [9], as well as recently introduced deep feature learning models such as Deep Belief Network [24] and Stacked Auto-encoder neural network [25], [26] are compared with the proposed GCDLA. The Persistence (PR) model [2] is further presented as a benchmark due to its decent performance for ultra-short-term and short-term predictions.

Tables I and II show the average test RMSE and MAE for short term wind speed forecasting with 10-min, 30-min, 1-hr, 2-hr, and 3-hr ahead forecasting tasks. PR leads to accurate ultra-short-term forecasts; however, for the extended prediction time horizons, the corresponding RMSE and MAE grow very fast

TABLE I
RMSE OF FORECASTING METHODS (M/S)

Method	Time Step				
	10-min	30-min	1-hr	2-hr	3-hr
Persistence	0.703	1.071	1.679	1.940	2.321
FFNN	0.670	0.920	1.302	1.521	1.714
TDNN	0.592	0.881	1.112	1.239	1.477
NARNN	0.570	0.820	1.105	1.178	1.251
SAE[26]	0.491	0.780	0.884	0.906	0.938
DBN[24]	0.487	0.774	0.845	0.872	0.913
GCDLA	0.431	0.702	0.721	0.781	0.807

TABLE II
MAE OF FORECASTING METHODS (M/S)

Method	Time Step				
	10-min	30-min	1-hr	2-hr	3-hr
Persistence	0.583	0.871	1.403	2.007	2.179
FFNN	0.442	0.821	0.983	1.439	1.641
TDNN	0.430	0.803	0.941	1.241	1.377
NARNN	0.398	0.793	0.902	1.001	1.214
SAE[26]	0.351	0.539	0.711	0.792	0.958
DBN[24]	0.349	0.518	0.682	0.740	0.912
GCDLA	0.301	0.491	0.533	0.692	0.784

due to the smoothness assumption. When the forecast horizon is extended, the linear correlation of future wind speed values and historical data is decreased; hence, PR is not a reliable technique for forecasts with longer horizons.

As shown in Table I, the recurrent shallow architectures, i.e., TDNN and NARNN, have smaller RMSE and MAE compared to FFNN. TDNN has 11.6% less RMSE in 10-min ahead forecasting. It also improves the FFNN's MAE by 2.71%. NARNN has generally better performance compared to TDNN with 6.29% and 8.79% RMSE and MAE improvements compared to TDNN on average over all prediction tasks. For the 3-hr ahead prediction task, a significant 15.30% RMSE and 11.83% MAE improvement is observed for the NARNN architecture. This is due to the recurrent signals that can help the model to capture meaningful temporal features from the underlying wind time series.

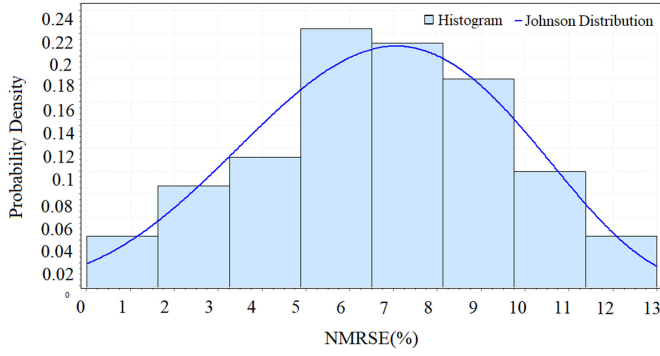


Fig. 8. Histogram of NMRSE percentage with the probability density function of the fitted Johnson distribution.

In recent years, two deep learning architectures, DBN [24] and SAE [25], [26], have been proposed for short term wind prediction. SAE applies a stack of auto-encoders to learn a nonlinear manifold from the underlying wind data. Similar to Generalized Principal Component Analysis (GPCA) utilized in the SVR methodology [6], the auto-encoders in [25] and [26] reduce the dimensionality of the historical wind speed data in order to recover useful data patterns for a linear regression (LR) model. The architecture contains several layers of sigmoid units trained using a greedy layer-by-layer unsupervised algorithm. The parameters of SAE, as well as the LR weights, are further fine-tuned applying a stochastic gradient descent optimizer with an L2 regularization error term for the weights. The DBN architecture contains several Restricted Boltzmann Machines trained using the Contrastive Divergence method [24] with Gibbs sampling to tune Bernoulli RBMs. The RMSE and MAE results in Tables I and II show the better accuracy of DBN in all time horizons. DBN outperforms SAE by 2.48% in RMSE and 3.98% in MAE. The higher accuracy of DBN shows the superiority of RBMs to effectively capture the probability distribution of the underlying wind data using generative techniques. The proposed spatio-temporal deep feature learning model, GCDLA, has 0.431 m/s RMSE in 10-min ahead wind prediction. This error is 11.49% lower than the average RMSE obtained by the state-of-the-art deep feature learning model, i.e., the DBN in [24]. GCDLA has also a MAE of 0.301 m/s which is significantly smaller than the MAE obtained by the DBN.

In order to further investigate the reliability of GCDLA, the Normalized Root Mean Squared Error (NRMSE), which is the RMSE in (17) normalized by the range of observed data, is computed to provide a scale-independent error measure. Fig. 8 shows the NMRSE percentage histogram as well as the probability density function of the fitted distribution. Applying the Kolmogorov-Smirnov statistical test [26], the Johnson distribution with a mean of 6.68% and standard deviation of 2.91% is fitted to the corresponding histogram data.

2) *Hybrid Methodologies*: Hybrid approaches make use of multiple feature extraction and regression methodologies for the prediction task. In this work, GCDLA is compared to three recently introduced hybrid methods presented below:

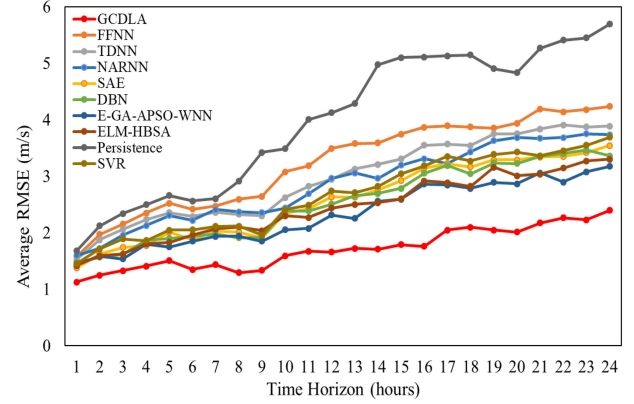


Fig. 9. Average RMSE comparison of single-model and hybrid methodologies.

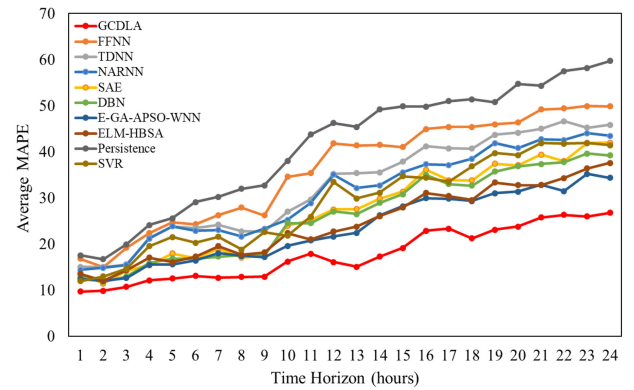


Fig. 10. Average MAPE comparison of single-model and hybrid methodologies.

- 1) E-GA-APSO-WNN [35]: This approach utilizes Ensemble Empirical Mode Decomposition (EEMD) to reduce uncertainties in wind data. Genetic Algorithm (GA) incorporated with Particle Swarm Optimization (APSO) is used as an optimization method for the parameter update process in a Wavelet Neural Network (WNN). GA and APSO help to find better local solutions for the network parameters; however, the time complexity of such evolutionary algorithms is very high.
- 2) ELM-HBSA [36]: A hybrid algorithm for short-term wind speed forecasting using a compound structure of Extreme Learning Machine (ELM) based on feature selection and parameter optimization using hybrid backtracking search algorithm (HBSA) is developed in [36]. The proposed approach effectively learns highly nonlinear characteristics of wind speed signals and outperforms ARIMA.
- 3) SVR [6]: The SVR-based methodology presented in [6] is also tested as a machine learning approach on the underlying regression problem. In this method, GPCA is employed to automatically discover patterns in the wind data while SVR is utilized for wind data regression to generate the final forecast.

Figs. 9 and 10 depict the average RMSE and MAPE of single-model approaches such as PR, FFNN, TDNN, NARNN, SAE,

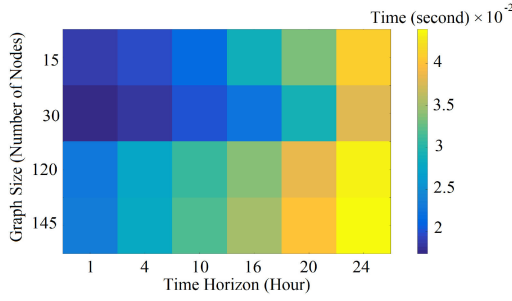


Fig. 11. Average training time of GCDLA using different number of input nodes with the change of the forecasting horizon (1-hr to 24-hr ahead forecasts).

DBN with the hybrid methodologies including E-GA-APSO-WNN, ELM-HBSA, and SVR. As shown in these plots, hybrid methodologies generally show better performance (smaller RMSE and MAPE) compared to single-model approaches. FFNN, TDNN, and NARNN are empirical risk minimization models in which the training process aims to find the best fits on the training samples. As a result, there may be cases where the individual error can be significantly high on the testing data, while SVR is a structural risk minimization model that can reduce the overfitting risk to a certain extent. Therefore, SVR has stronger individual error control ability than the single-model approaches. Moreover, SVR utilizes kernel tricks that lead to capturing features in high dimensional spaces while maintaining low computational cost.

Although deep single-model methods, DBN and SAE, are able to capture high abstractions in wind data which lead to better accuracy compared to SVR, other hybrid methodologies, E-GA-APSO-WNN and ELM-HBSA, still obtain better results. Among hybrid models, E-GA-APSO-WNN has generally the lowest RMSE and MAPE as a result of applying evolutionary computational techniques that find solutions (network parameters) close to the optimal solution of the regression problem. Despite the high accuracy of E-GA-APSO-WNN, this method cannot be tuned online due to very high time complexity of GA and APSO, while all single-model techniques including GCDLA are optimized both in offline and online settings. GCDLA outperforms E-GA-APSO-WNN by 0.32 m/s RMSE in 1-hr ahead prediction, which is increased to 0.78 m/s RMSE when the forecast horizon is extended to 24 hours. The superiority of GCDLA is also presented in Fig. 10. Here, MAPE increases with a faster rate for E-GA-APSO-WNN compared to GCDLA when the time horizon is extended.

G. Time Complexity Analysis

The time complexity of machine learning methodologies is a key factor considered when devising new computational algorithms for real-world problems. A proposed algorithm should have a running time less than the response time needed for the underlying problem. In this study, the online training time of the proposed GCDLA model is computed with the change of the modeled graph size N . Fig. 11 shows the online training time of GCDLA as a function of graph size and the forecasting time horizon. As shown in this plot, the running time increases

TABLE III
ACCURACY AND TRAINING TIME OF SEMI-SUPERVISED GCDLA FOR 6-HR AHEAD PREDICTION WITH RESPECT TO PERCENTAGE OF OBSERVABLE NODES

Percentage of Observable Nodes	MAPE	RMSE (m/s)	Offline Training time (min)	Online Training time (s)
20	21.49	2.15	15.12	0.017
40	18.76	2.03	16.95	0.018
60	15.01	1.56	20.04	0.023
80	14.83	1.39	25.19	0.028
100	13.05	1.35	28.43	0.033

TABLE IV
ACCURACY AND TRAINING TIME OF SEMI-SUPERVISED GCDLA FOR 12-HR AHEAD PREDICTION WITH RESPECT TO PERCENTAGE OF OBSERVABLE NODES

Percentage of Observable Nodes	MAPE	RMSE (m/s)	Offline Training time (min)	Online Training time (s)
20	33.28	2.83	19.04	0.016
40	28.19	2.29	20.12	0.019
60	22.53	1.85	29.41	0.025
80	19.94	1.73	33.09	0.029
100	16.05	1.66	35.72	0.032

as the forecast horizon is extended. This is because of the increase in the complexity of the problem which leads to larger architectures and more number of tunable parameters to update at each run of the proposed algorithm. The maximum update time is corresponding to the 24-hr ahead prediction using a graph of size 145 and at each run, all parameters are updated in 0.048 seconds. This period is negligible for online applications of this methodology. The running time corresponding to smaller time horizons or smaller input graph sizes, is less than this amount as the problem complexity decreases.

H. Semi-Supervised Learning

One major contribution of our proposed model is the ability to train the deep neural network in a semi-supervised approach. In contrast to the supervised setting, here, only a percentage of the number of nodes need to be observable during the training time while the rest of the nodes can be labeled during the test time. This characteristic helps the model decrease the offline training time (which is the time complexity bottleneck for neural network-based methodologies) while the accuracy is not sacrificed. Tables III and IV show the average MAPE and RMSE of GCDLA in our semi-supervised settings for 6-hr and 12-hr ahead forecasting tasks, respectively. As shown in Table III, the increase in the percentage of the observable nodes reduces the MAPE and RMSE, and increases the offline and online training times. Therefore, considering the negative correlation among the performance indices (i.e., RMSE and offline training time) a tradeoff can be reached by adjusting the percentage of the observable nodes. A similar notion is presented in Table IV for 12-hours ahead prediction. It is worth noting that the online and offline training times are negligible compared to the forecasting time steps in practice.

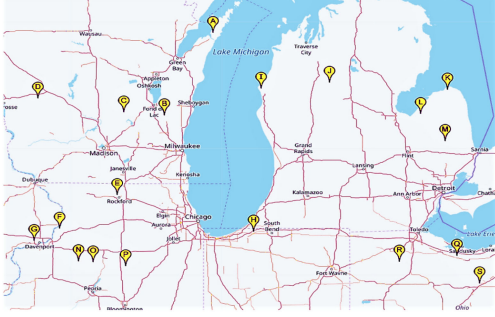


Fig. 12. Map of the studied region with 19 wind sites considered to investigate the spatial feature extraction.

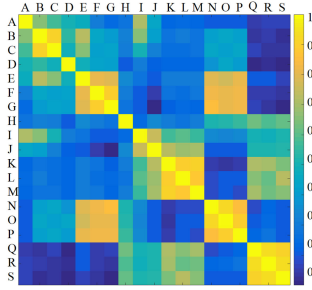


Fig. 13. Normalized mutual information matrix.

I. Spatial Outcomes

In order to show the effectiveness of our spatio-temporal methodology in capturing meaningful spatial features, actual and predicted wind speed values in 19 points of the studied region are investigated during the test time. The corresponding wind sites are denoted by alphabetical symbols in Fig. 12. As shown in this plot, the points are selected from various locations with a large diversity of distances in the Euclidean space. For instance, the points N and O are both in the north area of Peoria city with a relatively small distance with each other (14.32 miles), while E (near Rockford city) and Q (near Sandusky city) have a large distance of 435.46 miles from one another. Fig. 13 depicts the normalized mutual information matrix MI computed in our graph modeling process. As shown in this plot, nodes with lower spatial distance have higher historical wind correlation and larger weights in the adjacency matrix of G . In Fig. 13, there are three clusters of nodes in which the wind sites have highly correlated historical wind speed data: $\{G, F, N, O, P\}$, $\{K, L, M\}$, and $\{Q, R, S\}$. In our graph modeling procedure, the MI entries corresponding to node pairs inside each cluster are large while the correlation is decreased when MI is computed for nodes assigned to different clusters. For instance, considering nodes K and M, we have $MI(K, M) = 0.91$ while for K and O we have a smaller correlation of $MI(K, O) = 0.14$.

In order to see the effectiveness of spatio-temporal graph learning, a similarity measurement is defined on the actual and also forecasted wind speed measurements. For each pair of wind sites (i, j) and for each time step t , we define the similarity matrix entry as $S(i, j, t) = \exp(-diff(i, j, t))$ where $diff(i, j, t) = |v_t^i - v_t^j|$. Here, v_t^i is the wind speed time series value obtained from either GCDLA or the actual wind speed

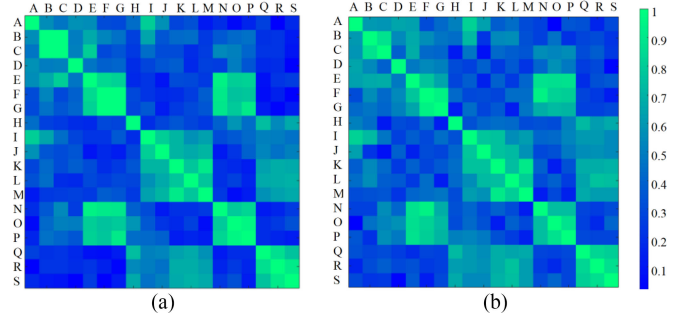


Fig. 14. Similarity matrices of actual wind speed and GCDLA output. (a) Similarity matrix of actual wind data. (b) Similarity matrix of GCDLA output.

measurement at wind site i in time step t . In Fig. 14(a), $S(i, j, t)$ is computed for the GCDLA's 3-hr ahead prediction outputs for all wind site pairs at 11:00 AM on April 3rd. Fig. 14(b) depicts similar results for the actual wind speed values for the same locations and time. As shown in this plot, each pair (i, j) has similar values in its corresponding entry in both similarity diagrams. Moreover, the patterns shown in Fig. 14(a) and (b) support the results presented in Fig. 13. If two data points are highly correlated, the corresponding entries have high actual/forecasted wind speed similarities in Fig. 14; however, the reverse may not necessarily be true. The comparison of the patterns in Fig. 14(a) and (b), shows that our proposed GCDLA can accurately capture the correlation among adjacent nodes in G using filters in (12). Moreover, comparing Figs. 13 with 14 justifies the utilization of mutual information as a reliable measure for the graph modeling in the proposed wind forecasting methodology.

V. CONCLUSIONS

In this paper, a novel spatio-temporal wind speed feature learning model is proposed based on Deep Learning, Graph Theory, and Rough Set Theory. The proposed architecture employs a recurrent LSTM model for temporal feature extraction of each wind site. The set of wind farms is modeled as a graph in which the nodes with high wind speed and direction correlations are connected by an edge. The extracted temporal features from each wind site are further fed to a novel graph convolutional deep learning architecture motivated by the localized first-order approximation of spectral graph convolutions. In order to learn robust features, Rough Set Theory is incorporated with the proposed framework by extracting interval upper-bound and lower-bound filtering parameters applied in the convolutional layers of GCDLA. The obtained spatio-temporal features can handle noise and uncertainties in the wind speed data. The results obtained by the proposed model on the Eastern Wind Dataset show significant RMSE and MAE improvement compared to shallow architectures and recent state-of-the-art Deep Learning architectures such as Deep Belief Networks and Stacked Auto-encoder networks.

REFERENCES

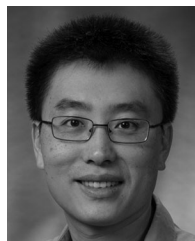
- [1] M. Ahlstrom, L. Jones, R. Zavadil, and W. Grant, "The future of wind forecasting and utility operations," *IEEE Power Energy Mag.*, vol. 3, no. 6, pp. 57–64, Nov./Dec. 2005.

- [2] J. Jung and R. P. Broadwater, "Current status and future advances for wind speed and power forecasting," *Renewable Sustain. Energy Rev.*, vol. 31, pp. 762–777, 2014.
- [3] G. Osorio, J. Matias, and J. Catalo, "Short-term wind power forecasting using adaptive neuro-fuzzy inference system combined with evolutionary particle swarm optimization, wavelet transform and mutual information," *Renewable Energy*, vol. 75, pp. 301–307, 2015.
- [4] N. Chen, Z. Qian, I. Nabney, and X. Meng, "Wind power forecasts using gaussian processes and numerical weather prediction," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 656–665, Mar. 2014.
- [5] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1033–1044, May 2014.
- [6] Q. Hu, P. Su, D. Yu, and J. Liu, "Pattern-based wind speed prediction based on generalized principal component analysis," *IEEE Trans. Sustain. Energy*, vol. 5, no. 3, pp. 866–874, Jul. 2014.
- [7] T. G. Barbounis and J. B. Theoharis, "Locally recurrent neural networks for long-term wind speed and power prediction," *Neurocomputing*, vol. 69, no. 4–6, pp. 466–496, 2006.
- [8] W. Chang, "A literature review of wind forecasting methods," *J. Power Energy Eng.*, vol. 2, no. 4, pp. 161–168, 2014.
- [9] H. Azad, S. Mekhilef, and V. Ganapathy, "Long-term wind speed forecasting and general pattern recognition using neural networks," *IEEE Trans. Sustain. Energy*, vol. 5, no. 2, pp. 546–553, Apr. 2014.
- [10] S. Soman, H. Zareipour, O. Malik, and P. Mandal, "A review of wind power and wind speed forecasting methods with different time horizons," in *Proc. North Amer. Power Symp.*, 2010, pp. 1–8.
- [11] T. Tjing Lie, D. Kaur, N. K. C. Nair, and B. Vallès, "Wind speed forecasting using hybrid wavelet transform—ARMA techniques," *AIMS Energy*, vol. 3, no. 1, pp. 13–24, 2015.
- [12] E. Erdem and J. Shi, "ARMA based approaches for forecasting the tuple of wind speed and direction," *Appl. Energy*, vol. 88, no. 4, pp. 1405–1414, 2011.
- [13] O. Shukur and M. Lee, "Daily wind speed forecasting through hybrid KF-ANN model based on ARIMA," *Renewable Energy*, vol. 76, pp. 637–647, 2015.
- [14] J. Wang and J. Hu, "A robust combination approach for short-term wind speed forecasting and analysis – Combination of the ARIMA (Autoregressive Integrated Moving Average), ELM (Extreme Learning Machine), SVM (Support Vector Machine) and LSSVM (Least Square SVM) forecasts using a GPR (Gaussian Process Regression) model," *Energy*, vol. 93, pp. 41–56, 2015.
- [15] O. Kramer and F. Gieseke, "Short-term wind energy forecasting using support vector regression," in *Proc. Intl. Conf. Soft Comput. Models Ind. Environ. Appl.*, Salamanca, Spain, Apr. 2011, pp. 271–280.
- [16] 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.
- [17] E. Cadenas and W. Rivera, "Short term wind speed forecasting in La Venta, Oaxaca, México, using artificial neural networks," *Renewable Energy*, vol. 34, no. 1, pp. 274–278, 2009.
- [18] O. Shukur and M. Lee, "Daily wind speed forecasting through hybrid KF-ANN model based on ARIMA," *Renewable Energy*, vol. 76, pp. 637–647, 2015.
- [19] Q. Cao, B. Ewing, and M. Thompson, "Forecasting wind speed with recurrent neural networks," *Eur. J. Oper. Res.*, vol. 221, no. 1, pp. 148–154, 2012.
- [20] W. Zhang, J. Wang, J. Wang, Z. Zhao, and M. Tian, "Short-term wind speed forecasting based on a hybrid model," *Appl. Soft Comput.*, vol. 13, no. 7, pp. 3225–3233, 2013.
- [21] W. Kehe, Y. Yue, C. Bohao, and W. Jinshui, "Research of wind power prediction model based on RBF neural network," in *Proc. Int. Conf. Comput. Inf. Sci.*, 2013, pp. 237–240.
- [22] N. Amjadi, F. Keynia, and H. Zareipour, "Short-term wind power forecasting using ridgelet neural network," *Elect. Power Syst. Res.*, vol. 81, no. 12, pp. 2099–2107, 2011.
- [23] K. Bhaskar and S. Singh, "AWNN-assisted wind power forecasting using feed-forward neural network," *IEEE Trans. Sustain. Energy*, vol. 3, no. 2, pp. 306–315, Apr. 2012.
- [24] C. Zhang, C. Chen, M. Gan, and L. Chen, "Predictive deep boltzmann machine for multiperiod wind speed forecasting," *IEEE Trans. Sustain. Energy*, vol. 6, no. 4, pp. 1416–1425, Oct. 2015.
- [25] M. Khodayar, O. Kaynak, and M. Khodayar, "Rough deep neural architecture for short-term wind speed forecasting," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2770–2779, Dec. 2017.
- [26] M. Khodayar and M. Teshnehlab, "Robust deep neural network for wind speed prediction," in *Proc. 4th Iranian Joint Congr. Fuzzy Intell. Syst.*, 2015, pp. 1–5.
- [27] L. Xie, Y. Gu, X. Zhu, and M. Genton, "Short-term spatio-temporal wind power forecast in robust look-ahead power system dispatch," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 511–520, Jan. 2014.
- [28] T. Gneiting, K. Larson, K. Westrick, M. Genton, and E. Aldrich, "Calibrated probabilistic forecasting at the stateline wind energy center," *J. Amer. Statist. Assoc.*, vol. 101, no. 475, pp. 968–979, 2006.
- [29] J. Dowell, S. Weiss, D. Hill, and D. Infield, "Short-term spatio-temporal prediction of wind speed and direction," *Wind Energy*, vol. 17, no. 12, pp. 1945–1955, 2013.
- [30] M. He, L. Yang, J. Zhang, and V. Vittal, "A spatio-temporal analysis approach for short-term forecast of wind farm generation," *IEEE Trans. Power Syst.*, vol. 29, no. 4, pp. 1611–1622, Jul. 2014.
- [31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [32] D. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmonic Anal.*, vol. 30, no. 2, pp. 129–150, 2011.
- [33] Z. Pawlak, "Rough set theory and its applications to data analysis," *Cybern. Syst.*, vol. 29, no. 7, pp. 661–688, 1998.
- [34] A. Ezzat, M. Jun, and Y. Ding, "Spatio-temporal asymmetry of local wind fields and its impact on short-term wind forecasting," *IEEE Trans. Sustain. Energy*, to be published.
- [35] T. Filik, "Improved spatio-temporal linear models for very short-term wind speed forecasting," *Energies*, vol. 9, no. 3, 2016, Art. no. 168.
- [36] J. Dowell, S. Weiss, and D. Infield, "Kernel methods for short-term spatio-temporal wind prediction," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2015, pp. 1–5.
- [37] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. International Conference on Learning Representations*, Toulon, France, arXiv:1609.02907, 2016.
- [38] "Eastern wind integration and transmission study," EnerNex Corporation, Knoxville, TN, USA, 2011. [Online]. Available: <http://www.nrel.gov/docs/fy11osti/47078.pdf>. [Accessed on: Jan. 21, 2017].
- [39] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," arXiv:1603.04467, 2016.



Mahdi Khodayar (S'17) received the B.Sc. degree in computer engineering and the M.Sc. degree in artificial intelligence from Khajeh Nasir Toosi University of Technology, Tehran, Iran, in 2013 and 2015, respectively. He is currently working toward the Ph.D. degree in electrical engineering with Southern Methodist University, Dallas, TX, USA. His main research interests include machine learning and statistical pattern recognition. He was a Reviewer for reputable journals including the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS

ON FUZZY SYSTEMS, and IEEE TRANSACTIONS ON SUSTAINABLE ENERGY. He is currently focused on deep learning methodologies for pattern recognition in power systems.



Jianhui Wang (M'07–SM'12) received the Ph.D. degree in electrical engineering from Illinois Institute of Technology, Chicago, IL, USA, in 2007. He is currently an Associate Professor with the Department of Electrical Engineering, Southern Methodist University, Dallas, TX, USA. He is the secretary of the IEEE Power & Energy Society (PES) Power System Operations, Planning & Economics Committee. He is an Associate Editor of the *Journal of Energy Engineering* and an Editorial Board Member of the *Applied Energy*. He has held visiting positions in Europe, Australia, and Hong Kong, including a VELUX Visiting Professorship with the Technical University of Denmark. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON SMART GRID and an IEEE PES Distinguished Lecturer. He is also the recipient of the IEEE PES Power System Operation Committee Prize Paper Award in 2015.