IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 June 2025, accepted 4 July 2025, date of publication 24 July 2025, date of current version 30 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3586997

SURVEY

Spatiotemporal Wind Energy Forecasting: A Comprehensive Survey and a Deep Equilibrium-Based Case Study With StemGNN

LUIZA SCAPINELLO AQUINO[1], LAIO ORIEL SEMAN[2], VIVIANA COCCO MARIANI[3], LEANDRO DOS SANTOS COELHO[1], STEFANO FRIZZO STEFENON[1,4], AND GABRIEL VILLARRUBIA GONZÁLEZ[5]

[1]Graduate Program in Electrical Engineering (PPGEE), Federal University of Paraná (UFPR), Curitiba 80060-000, Brazil
[2]Department of Automation and Systems Engineering, Federal University of Santa Catarina (UFSC), Florianópolis 88040-900, Brazil
[3]Graduate Program in Mechanical Engineering (PGMec), Federal University of Paraná (UFPR), Curitiba 80060-000, Brazil
[4]Lisbon School of Engineering (ISEL), Polytechnical Institute of Lisbon (IPL), 1959-007 Lisbon, Portugal
[5]Expert Systems and Applications Laboratory, Faculty of Science, University of Salamanca, 37008 Salamanca, Spain

Corresponding author: Luiza Scapinello Aquino (luiza.scapinello@ufpr.br)

This work was supported in part by the National Council of Scientific and Technologic Development of Brazil (CNPq) under Grant 314389/2023-7-PQ, Grant 313169/2023-3-PQ, Grant 407453/2023-7-Universal, and Grant 442176/2023-6-Peci; in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES) under Grant 001; and in part by the Colaboración Consejería de Educación de la Junta de Castilla y León Grupo de Investigación through ESAL–Expert System and Applications Laboratory (ESALAB).

ABSTRACT Accurate spatiotemporal wind energy forecasting is essential for ensuring grid stability and maximizing the efficiency of renewable energy systems. This paper addresses the challenge of modeling the complex spatial and temporal dependencies inherent in wind power generation by presenting a comprehensive survey of existing spatiotemporal forecasting methods and introducing an innovative deep learning approach. The proposed model integrates a Graph Neural Network (GNN) to represent wind turbines as nodes within a graph, capturing spatial relationships, while a Deep Equilibrium Model (DEQ) enables equilibrium-based inference to handle highly nonlinear wind patterns. A Sequence-to-Sequence (Seq2Seq) architecture further manages temporal dependencies. The method was validated using a real-world dataset of wind power generation, outperforming baseline models across multiple forecast horizons and maintaining stable accuracy across short- and mid-term predictions. Results demonstrate that the proposed GNN with DEQ effectively models both spatial and temporal dynamics for Seq2Seq data, improving prediction accuracy while maintaining computational efficiency. This study highlights the potential of equilibrium-based spatiotemporal graph models for wind energy forecasting and provides a robust tool for better integration of wind power into modern power grids.

INDEX TERMS Wind energy, graph neural network, deep equilibrium, spatiotemporal forecasting.

I. INTRODUCTION

Due to its abundance and lack of pollution, wind energy has emerged as a crucial source of energy all over the world. The regional power grid may suffer from severe power fluctuations because of its high volatility. The safe and steady operation of the entire power system may be jeopardized by hidden threats from insufficiently accurate wind forecasts [1]. An efficient wind power forecasting approach is required to determine the most cost-effective way to operate the electricity system. This can assist the power dispatching department in organizing the generation plan properly, enhancing the grid's dependability and security [2].

Given the chaotic and stochastic characteristics of wind speed time series, its power forecasting is typically seen as a challenging undertaking. Some broad categories can be

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

used to group current wind power forecasting techniques. As an example, there is the idea of persistence techniques [3], which rely on the assumption that wind data do not change in a short period, as well as physical approaches, which use numerical weather prediction data, such as temperature, pressure, surface roughness, and impediments, to formulate the wind power forecasting problem. There are also statistical approaches, such as Autoregressive Integrated Moving Average (ARIMA) model grounded in the Box-Jenkins methodology [4], [5], which rely on statistical methods to analyze past data and make predictions about future outcomes.

Wind power data's nonlinearities and intricate geographical relationships are difficult to capture using conventional time series forecasting models. Artificial Neural Networks (ANNs) [6], and Deep Learning (DL) (hybrid methods [7], attention-based models [8], and using denoising techniques [9]) have demonstrated promising results in solving these difficulties, with techniques such as GNNs [10], DEQ [11], and Seq2Seq [12] emerging as efficient tools for modeling spatiotemporal data. Both techniques use optimization algorithms (social-spider optimization algorithm [13], annealing algorithm [14], and multi-objective optimization with non-dominated sorting genetic algorithm version II [15]) to obtain suitable hyperparameters and to acquire the complex relationship between input and output variables by analyzing large amounts of past data. GNNs are well-suited for modeling dynamic systems, particularly those that can be represented as graphs with evolving structures or attributes over time. Dynamic systems often involve entities and their interactions changing over time, and GNNs can effectively capture these temporal dynamics [16].

Spatiotemporal forecasting models leverage data from nearby spaces to quickly enhance projections from a target one [17]. Interest in models that can consider other spatially distributed data, in addition to historical ones about the subject of forecasting, has grown in recent years on account of the strong cross-correlation that exists between a target time series data and those of its neighboring space relatives [18]. The concept of these models suggests that the data positively affects forecasting accuracy if they are highly associated with the objective variable.

As a result, the research in the literature on spatiotemporal forecasting is primarily concerned with selecting the most insightful input data from a pool of candidate variables. Numerous real-world events are spatiotemporal, and accurately predicting these systems' futures based on past observations is crucial for a variety of practical applications, including smart city problems [19] and traffic [20], as well as photovoltaic power production [21] and even medical problems, for instance the ones regarding virus spread [22].

A development in this field is the application of unified graph formulations for spatiotemporal forecasting [23]. In this methodology, each recorded data sample is treated as a node within a graph, allowing GNNs to simultaneously learn spatial and temporal dependencies. This unified approach eliminates the need for separate temporal networks and naturally accommodates irregular time series data, varying sampling frequencies, and missing information. Studies have demonstrated that such frameworks outperform traditional architectures, especially in scenarios with incomplete data, by achieving lower errors in wind speed predictions [24].

In this paper, an approach for wind power forecasting using spatiotemporal data is proposed. Each wind turbine is represented as a node in a graph with GNNs to model the complex spatial relationships among different wind farms. To manage the temporal dependencies in the wind data, a Seq2Seq (Sequence-to-Sequence) model is applied to operate on the features processed by the DEQ. The DEQ represents a novel deep-learning paradigm that perceives the entire network as a single layer solved iteratively to equilibrium, making it efficient for modeling highly complex wind patterns. Furthermore, a latent correlation layer is introduced into the model, enhancing its ability to capture complex, non-linear dependencies in the data. Attention mechanisms are also incorporated, highlighting the most discriminative features in the time-series data for making accurate predictions.

The approach is validated using a real-world wind power generation dataset. Experimental results indicate that the model outperforms state-of-the-art wind power forecasting methods, illustrating its effectiveness in capturing spatial and temporal dependencies. This study makes a contribution to the advancement of wind power forecasting approaches and offers a robust tool for more efficient integration of wind power into the power grid. The main contributions of this study are:

- A rigorous exploration and analysis of the application of spatiotemporal forecasting methodologies in various contexts drawn from an extensive literature review is presented.
- It is proposed a unique integration of DEQs into the StemGNN framework, enabling equilibrium-based inference over spatiotemporal graph data, an approach not previously explored in this context.
- The proposed model introduces a latent correlation layer and attention-guided Seq2Seq blocks in the frequency domain, enhancing temporal feature extraction and capturing nonlinear spatial dependencies.
- This paper provides a comparison against the original StemGNN, existing DEQ-based time series models, and state-of-the-art GNN-based forecasting methods, empirically demonstrating the performance and architectural advantages of the proposed approach.

The remainder of the paper is organized as follows: an overview of the fundamentals of spatiotemporal techniques is given in Section II. Section III reviews the related work on spatiotemporal series forecasting, highlighting the motivation for the proposed approach. Section IV presents

the description of GNN, along with information about DEQs. Section V describes the proposed method, dataset, and performance metrics used to evaluate the presented forecasting model. Section VI presents the results and discussion, while Section VII concludes the paper and discusses future research directions.

II. FUNDAMENTALS OF SPATIOTEMPORAL FORECASTING

Spatiotemporal models make use of time series data from several available places; therefore, rather than concentrating on the data from a single location, these models consider the spatiotemporal dependency structure in an area of interest [25]. These models integrate two distinct forecasting techniques, the first one being temporal modeling, in which projected future values are forecast using past data from the same point, and the second one refers to spatial modeling, in which the data are replaced at applications where there is a lack of information [26].

The main principle behind spatiotemporal forecasting is attached to the concept of inertia, meaning that the impacts of an occurrence at one point in a system may, over time, spread to its surroundings. To reiterate, any data coming from a different place can potentially help model a target variable at some time [27]. Therefore, spatiotemporal forecasting broadens classical time series forecasting or spatial interpolation approaches into space-time dimensions. For example, if the goal of the forecast is wind speed, not just timestamps of different past speeds, past latitudes, and temperatures can also be advantageous for variable forecasts.

The data, model structure, parametrization, and particular applications affect spatiotemporal forecasting accuracy [28]. Therefore, applications of spatiotemporal forecasting may handle uncertainty originating from both the model and the data applied [29]. There may be errors in the datasets from various causes, including measurement errors caused by external factors, instrumental insufficiency, and methodological uncertainty. External conditions may include temperature, solar radiance, and rain, which impact the accuracy of the data acquisition. What refers to instrumental errors is the inability to have flawless equipment containing inherent systematic and random instrumental errors [30].

Spatial and time-based elements are both included in spatiotemporal data. Three distinct characteristics define them: non-spatiotemporal, spatial, and temporal. The non-contextual features of things are indicated by the first attribute [31]. The spatial characteristics define the location, range, and shape of the items. The timestamps and durations of processes are marked by temporal attributes, representing the vector or raster layers of a spatial object.

Spatial and temporal data can be categorized as discrete or continuous observations depending on how they were gathered. Events and data trajectories are cases of discrete data types and can be exemplified in Figure 1, whereas point reference and raster data are continuous data types [32], as shown in Figure 2.

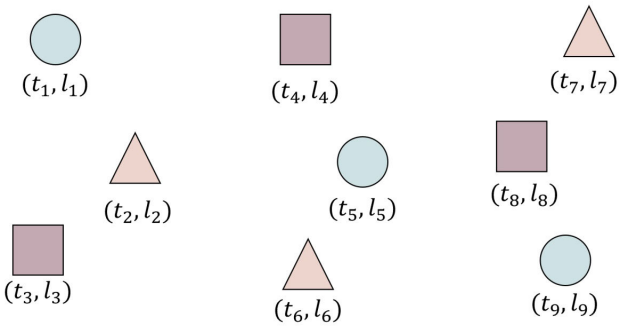


FIGURE 1. Numerous sorts of spatiotemporal occurrences carried out at various times and places.

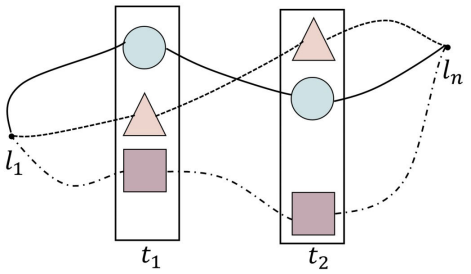


FIGURE 2. Spatiotemporal trajectories between locations at different times.

A. METHODOLOGIES APPLIED TO SPATIOTEMPORAL FORECASTING

Spatiotemporal forecasting methodologies focus on predicting the state of phenomena that have spatial and temporal dimensions. For example, weather forecasting [33], pollution levels [34], crime rate [35], and disease spread [36] are typical spatiotemporal problems. The key characteristic of spatiotemporal forecasting methodologies is the inclusion of both the chronological sequence of data and the spatial dependencies between different locations [37]. Hence, capturing both the progression over time and the interdependencies across space. Moreover, they can handle large, complex data effectively, particularly with the advances in computational power and data storage capabilities [38].

These methodologies can also accommodate nonlinear relationships. Advanced spatiotemporal methods, such as those based on machine learning approaches, can capture these nonlinearities. In addition, incorporating contextual or auxiliary information, such as temporal trends, seasonal cycles, or extra variables, into the forecasting model can aid the predictions [39]. Feature selection and dimensionality reduction are also important components in such cases [40].

The evolution of spatiotemporal forecasting methodologies over the years can be summarized in statistical models, beginning with AutoRegressive Moving-Average

model (ARMA) [41] and ARIMA model [42], and its variants, as is the case of AutoRegressive Integrated Moving Average with eXogenous inputs (ARIMAX) [43], that could simultaneously handle spatial and temporal data. Other models important to mention are the seasonal ARIMA with and without exogenous inputs, i.e., SARIMAX [44] and SARIMA [45] models, respectively. Next, there were the spatiotemporal statistical models, such as SpatioTemporal ARIMA (STARIMA) [46] and vector autoregressive [47] models. These models are presented in Figure 3.

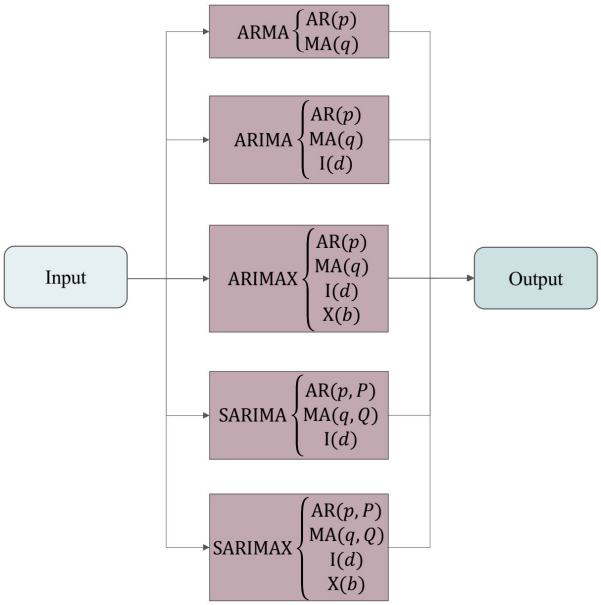


FIGURE 3. Spatiotemporal forecasting representation chart: statistical models.

Machine Learning models, on the other hand, provide a data-driven approach that allows for complex nonlinear relationships between predictors and response variables. Models such as Support Vector Machine (SVM) [48], Random Forest (RF) [49], and Gradient Boosting Machines (GBMs) [50] have been widely used. Recently, with the success of DL in various domains, models like Convolution Neural Networks (CNNs) [51], Recurrent Neural Networks (RNNs) [52], Long Short-Term Memory (LSTM) networks [53], GNNs [54], and hybrid methods [55] have been employed. Ultimately, hybrid and ensemble methods are being developed to leverage the strengths of different models [56]. These can combine traditional statistics with machine learning or deep learning models to improve forecasting accuracy and can be seen in Figure 4.

Recent advancements in industrial fault detection emphasize the limitations of traditional global- and local-based methods [57], which often assume independence among local variable groups, a condition rarely met in real-world scenarios. To address this, a novel mutual stacked autoencoder has been proposed.

This architecture enhances local details, global structures, and residual associations by incorporating a multi-task entropy-aided loss function. Evaluations across eleven datasets demonstrate its superior performance and component efficacy, offering a robust solution for accurate multivariable industrial fault detection [58].

B. ACCURACY OF FORECASTING MODELS

Incorporating spatiotemporal data can improve the accuracy of forecasting models by providing them with more holistic and relevant information, due to the inherent spatial and temporal dependencies in many real-world phenomena that interact with their surroundings. For example, wind patterns are influenced by both temporal factors, such as diurnal and seasonal cycles, and spatial factors, namely topography and proximity to geographical features.

Also, wind conditions in one location can influence or be influenced by conditions in neighboring areas. Spatiotemporal data can help in detecting anomalies or unusual events that may disrupt regular patterns. For instance, abnormal temperature spikes and other factors like temperature, humidity, and air pressure can be identified by comparing the current data with historical spatial and temporal patterns.

The model can learn from a larger collection of features that are provided by spatiotemporal data. Because of this, the model may be able to identify complex, nonlinear interactions that would be missed by simpler models. Furthermore, models that have been trained on spatiotemporal data frequently generalize well to novel contexts. This is because they are less likely to be taken aback by new data as they are exposed to a greater range of circumstances during training.

III. RELATED WORK ON WIND ENERGY

A vital component of the renewable energy industry is wind power forecasting, which enables ideal energy management and grid integration. The current wind power prediction techniques can be divided into physics, statistics, and machine learning approaches [59]. The first one uses historical data to develop forecasting models based on time series analysis and regression techniques. Physical models depend on numerical weather prediction models to forecast meteorological variables. In addition, machine learning approaches such as ANNs, SVMs, and ensemble learning models have drawn more and more attention in recent years, given their promise to increase accuracy and adaptability [60].

To get the best performance, the work of Aly [61] makes use of various fusions of the Recurrent Kalman Filter, Fourier series, wavelet neural networks, and ANN evaluated in twelve distinct hybrid models, as it combines deep learning hybrid models with clustered segments to enhance system performance overall. Lui et al. [62] proposed a Jaya algorithm-based SVM model for short-term wind speed forecasting. While Shahid et al. [63] proposed a framework

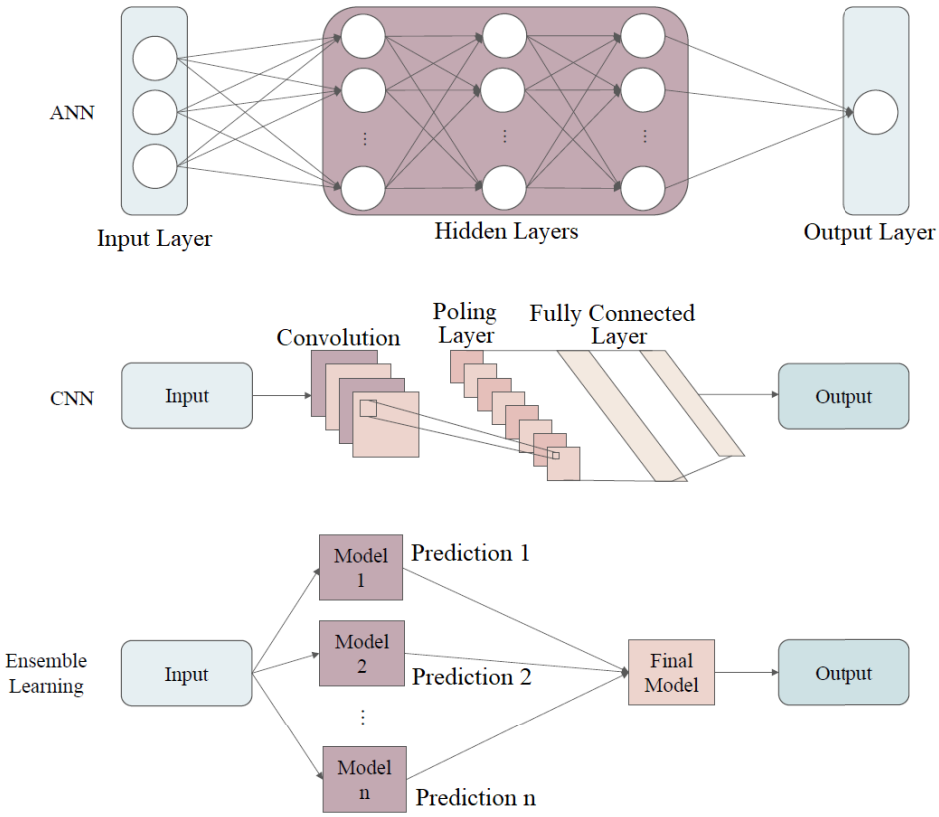


FIGURE 4. Spatiotemporal forecasting representation chart: machine learning models.

consisting of a genetic algorithm with LSTM for short-term wind power prediction.

Liu et al. [64] proposed the use of discrete wavelet transform and LSTM networks to anticipate wind energy. Therefore, the dynamic behavior of the wind power time series may be successfully displayed by the LSTM network thanks to its architecture. At the same time, the discrete wavelet transform is used to divide the non-stationary wind power time series into more predictable and stationary components. Santamaría-Bonfil et al. [65] proposed a hybrid methodology based on Support Vector Regression for wind speed forecasting.

Spatiotemporal dependencies must be taken into account in wind power forecasting to capture underlying trends better. Wu et al. [66], for the prediction of ultra-short-term wind power using CNNs and LSTM, presented a model based on a spatiotemporal correlation model. Li and Armandpour [67] used historical wind speed data and a wind turbine's spatial location in relation to other turbines to predict the amount of wind power that will be produced. Lu et al. [68], to anticipate the output wind power from several wind farms, proposed a framework for ultra-short-term forecasting based on spatiotemporal analysis, multi-output support vector machines, and grey wolf optimizer.

The work of Zhao et al. [69] explored a hybrid deep learning model comprised of variable mode decomposition, CNN, and Gated Recurrent Unit (GRU), in which the wind speed sequence volatility is significantly decreased with Variational Mode Decomposition (VMD), while the GRU can directly extract temporal features from historical input data. The CNN automatically extracts complex spatial features from the wind power data. Wilms et al. proposed employing a convolutional LSTM to include both spatial and temporal dependencies gleaned from regionally dispersed wind forecasts and time series. WMD was also applied for signal denoising in [70] for wind speed forecasting and in [71] for streamflow series forecasting, showing promising results.

GNNs have emerged as a powerful tool for modeling structured data, particularly in the context of spatiotemporal forecasting. GNNs are well-suited for this task, as they can effectively model the spatial relationships among geographically distributed wind farms while considering the temporal dynamics. Several studies have demonstrated the potential of GNNs in wind power forecasting, such as the work of Dong et al. [72], which proposed a directed graph convolutional structure-based spatiotemporal convolutional network, and the temporal characteristics of wind power were also characterized using a temporal CNN.

Wu et al. [73] proposed a multidimensional spatiotemporal GNN for wind speed prediction. Li et al. [74] presented an adaptive spatiotemporal fusion GNN for short-term power forecasting of multiple wind farms. Bentsen et al. [75] applied a GNN to extract spatial connections and learn temporal correlations using various update functions, and many neural network designs were used to implement these update functions.

The application of DEQs in the proposed model is grounded on its unique capability to address complex systems by treating the entire network as a single layer solved iteratively to equilibrium. This revolutionary approach was first introduced by Bai et al. [11], who demonstrated its effectiveness in tackling various tasks by integrating it into different architectures. DEQs have been further explored by Bai et al. [76] in optical flows. DEQs are also used in imaging applications, such as the case of the work of Zhang et al. [77] in medical imaging and in the work of Zhao et al. [78] for snapshot compressive imaging.

A. BIBLIOMETRIC SURVEY ON SPATIOTEMPORAL WIND FORECASTING

Numerous studies have explored various aspects of wind prediction, seeking to enhance accuracy. This introduction sets the stage for a bibliometric study that delves into related works on wind energy forecasting with a specific focus on spatiotemporal dynamics. By employing bibliometric analysis across diverse repositories, such as Web of Science (WoS), Scopus, and Dimensions, aiming to systematically map and evaluate the existing body of literature. Through this exploration, the research seeks to uncover prevalent themes, identify influential authors, journals, and countries, and offer a comprehensive understanding of the evolving landscape in spatiotemporal wind energy forecasting.

Spatiotemporal Wind Forecasting was the search term used for WoS, Scopus, and Dimensions when the search was conducted in February 2025. Each database potentially contained overlapping or irrelevant studies. Filtering was performed to ensure relevance and avoid redundancy, as seen in Figure 5. Filtration was carried out according to the PRISMA guidelines [79], however, the order of the steps was modified to facilitate the authors' management of each database separately before conversion. After the initial search, the papers were filtered by date, therefore, only the most recent studies were considered pertinent, and then studies deemed irrelevant to the survey's scope were excluded through a thorough screening process. Duplicate entries, resulting from papers appearing in multiple databases, were also removed. Afterward, all repositories were compared, covering publications in the last few years, when spatiotemporal forecasting was more actively applied.

In the field of wind energy forecasting with a spatiotemporal emphasis, the role of influential papers is pivotal in influencing innovation. This bibliometric study explores the scholarly landscape, displaying the contributions of key articles in the field. Therefore, Table 1 presents the 10 most relevant papers in this area. The table shows a variety of approaches used in the research, such as fully connected LSTM, ensemble learning, wavelet transformation, variational Bayesian, along deep learning. The number of citations serves as a quantitative measure of the impact each paper has had within the academic community, measuring its significance in the realm of spatiotemporal wind forecasting.

The five most productive countries in the related field, as measured by the number of publications from the WoS, Scopus, and Dimensions, are listed in Table 2, sorted by paper counts and considering paper citations. China emerges as a dominant force in the three repositories, consistently securing the top position in terms of both scientific document production and citations. The inclusion of the United States of America (USA), Germany, India, and the United Kingdom in the top five countries across all repositories reflects the global collaboration and diverse geographical distribution of research efforts in spatiotemporal wind forecasting.

1) KEYWORKS ANALYSIS

The most popular terms used by authors in their papers are displayed in this section using VOSviewer, a widely used information visualization tool. In Figure 6, the occurrence of the most common keywords in WoS is presented, where words representing predictions and forecasting models are enlarged, meaning more commonly used. The same occurs in Figure 7, in which the most common keywords in text from Scopus are shown. Dimensions, however, do not provide keyword information.

The visual representations depicted in the images provide a comprehensive overview of the most frequently used keywords within the repositories WoS and Scopus, focusing specifically on terms that have garnered more than 20 occurrences. Scopus exhibits a higher frequency of keywords surpassing the 20-occurrence threshold in comparison to WoS. These visualizations not only underscore the distinctiveness in the utilization of specific terms within the two databases but also offer valuable insights into the evolving trends and emphases within the scholarly literature encapsulated by Scopus and WoS.

Moreover, upon closer examination of the visualizations, it becomes evident that certain keywords exhibit a notable convergence between the repositories, such as ''machine learning,'' ''wind power,'' ''wind speed,'' and ''prediction'' are prominently featured in both images, transcending the numerical disparities in keyword occurrences. This shared emphasis on specific terms underscores their universal significance within the academic discourse encapsulated by both databases.

In VOSviewer, color clusters typically represent thematic groupings of keywords based on their co-occurrence patterns. Each color corresponds to a cluster, and keywords within the same cluster are more closely related to each other in terms of how often they appear together in documents. These clusters

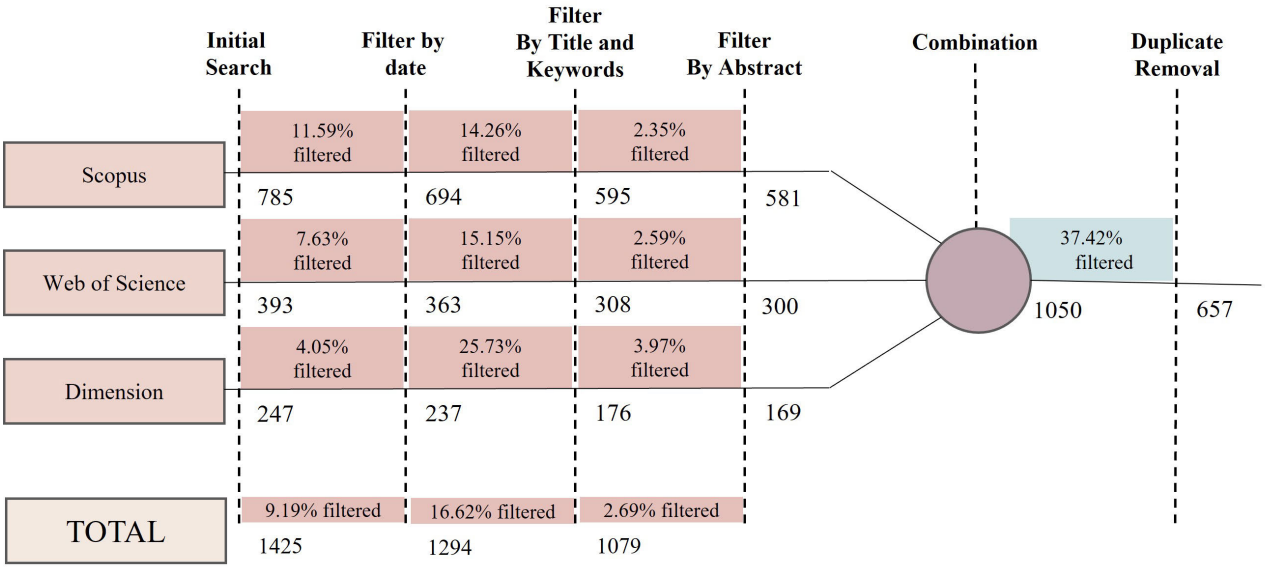


FIGURE 5. Filter map study.

TABLE 1. Top 10 most influential papers.

Paper	Applications	Approach	# Citations
Khodayar and Wang, 2018 [80]	Wind Speed	Deep GNN	436
Zhao *et al.*, 2019 [81]	Air Pollution	Fully Connected LSTM	414
Heinemann and Kramer, 2016 [82]	Wind Power	Ensemble Learning	262
Zhu *et al.*, 2019 [83]	Wind Speed	CNN + LSTM	216
Chen *et al.*, 2019 [84]	Wind Speed	CNN + LSTM	194
Liu *et al.*, 2020 [85]	Wind Speed	Variational Bayesian	185
Zhao *et al.*, 2018 [86]	Wind Power	Autoregressive	173
Shahid *et al.*, 2020 [87]	Wind power	wavelets LSTM	166
Zhu *et al.*, 2018 [88]	Wind Speed	Deep CNN	143
Zhang *et al.*, 2021 [89]	Wind Speed	CNN + Attention Mechanism	99

TABLE 2. Top 5 countries publishing about spatiotemporal wind forecasting.

Repository	Country	# Documents	# Citations
WoS	China	165	1774
	United States of America	95	1723
	Germany	19	290
	India	17	185
	United Kingdom	15	235
Scopus	China	298	4062
	United States of America	188	4845
	United Kingdom	39	983
	India	37	395
	Germany	30	920
Dimensions	China	87	123
	United States of America	61	99
	United Kingdom	21	34
	Germany	13	25
	India	8	7

can reveal thematic areas or topics that have a higher degree of interconnectedness in the analyzed literature.

IV. FUNDAMENTALS ON GRAPH NEURAL NETWORKS AND DEEP EQUILIBRIUM MODEL

This section provides a detailed exploration of GNNs and DEQs, powerful frameworks for learning from structured and sequential data. First, the fundamental concepts of GNNs are introduced, highlighting their ability to process graph-structured information and capture complex relational patterns. Then, the DEQs are explained, focusing on their implicit modeling approach and advantages over traditional deep networks.

A. GRAPH NEURAL NETWORK

A GNN is a highly effective machine learning framework designed for processing data represented as graphs [90]. GNNs extend the capabilities of CNNs to handle graph-structured data, and they have demonstrated remarkable performance in various tasks, including node classification [91], link prediction [92], and graph classification [93].

Formally, let $G = (V, E)$ denote a graph, where $V = \{v_1, v_2, \ldots, v_n\}$ represents the set of nodes and $E \subseteq V \times V$

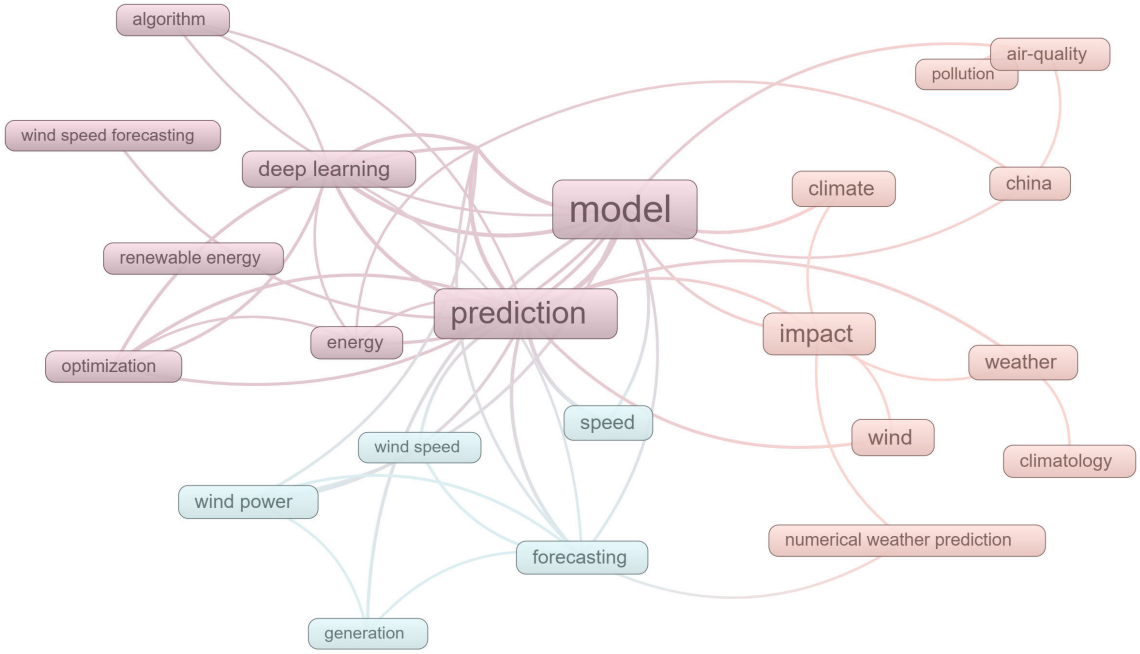


FIGURE 6. Most used keywords in web of science.

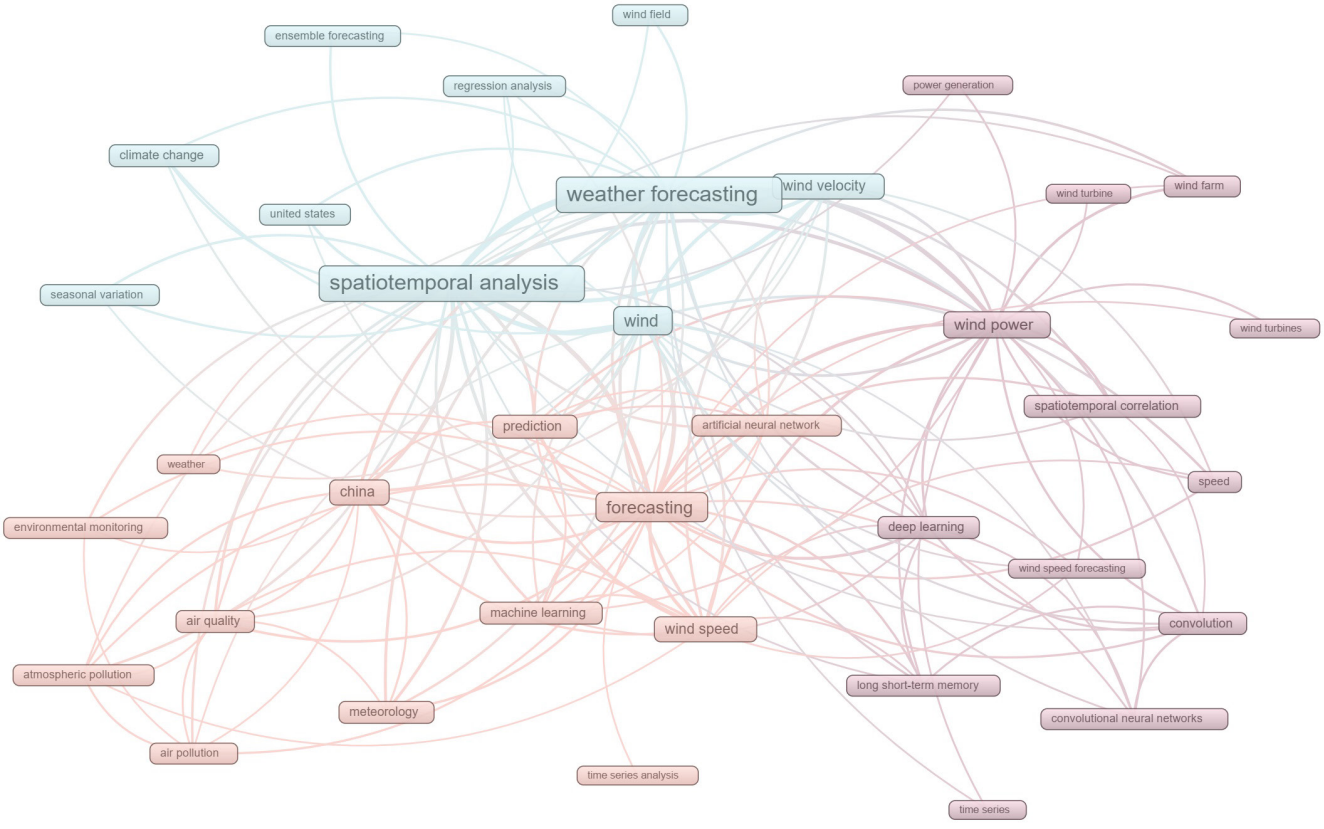


FIGURE 7. Most used keywords in scopus.

represents the set of edges. Each node $v_i \in V$ is associated with a feature vector $\mathbf{x}_i \in \mathbb{R}^d$. A GNN processes the graph by iteratively updating node embeddings $\mathbf{h}_i^{(t)} \in \mathbb{R}^p$, where t denotes the iteration or layer of the GNN. The updated

embeddings are computed based on the node's own features and the embeddings of its neighbors using the following equation:

$$\mathbf{h}_i^{(t+1)} = \text{ReLU}\left(\mathbf{W}^{(t)} \cdot \text{agg}^{(t)}(\mathbf{h}_j^{(t)} : v_j \in \mathcal{N}(v_i)) + \mathbf{b}^{(t)}\right), \tag{1}$$

where $\mathcal{N}(v_i)$ represents the set of neighbors of node v_i, $\mathbf{W}^{(t)} \in \mathbb{R}^{p \times p}$ and $\mathbf{b}^{(t)} \in \mathbb{R}^p$ are learnable parameters at layer t, and $\text{agg}^{(t)}$ is an aggregation function that combines the embeddings of neighboring nodes. Commonly used aggregation functions include summation, mean, and maximum.

The GNN operates for T iterations, and the final node embeddings are utilized for downstream tasks such as node classification:

$$\mathbf{z}_i = \text{softmax}\left(\mathbf{W}^{(T)} \cdot \mathbf{h}_i^{(T)} + \mathbf{b}^{(T)}\right). \tag{2}$$

The model parameters are learned by minimizing a loss function $\mathcal{L}(\mathbf{z}_i, y_i)$, where y_i represents the ground truth label associated with node v_i.

1) GRAPH CONVOLUTIONAL NETWORKS

A Graph Convolutional Network (GCN) is a type of neural network architecture designed to work directly on graph-structured data [94]. Instead of requiring the graph to be transformed into a vector representation, it operates directly on the graph structure, propagating information across the nodes of the graph to generate contextually relevant node embeddings.

The central operation of a GCN is rooted in the concept of message passing, where each node forwards its feature vector to all its connected neighbors [95]. These nodes, in turn, update their respective feature vectors based on the received messages. Mathematically, this process can be explained as follows.

Suppose A is the adjacency matrix representing the graph, and X is a matrix where each row corresponds to the feature vector of a node. The functioning of a GCN layer can then be represented as:

$$\tilde{A} = A + I \tag{3}$$

Here, I represents the identity matrix, the addition of which introduces self-loops to the graph. Then,

$$D = \text{diag}(\tilde{A} \cdot \mathbf{1}) \tag{4}$$

In this equation, $\mathbf{1}$ is a vector of ones, and $diag(\cdot)$ is a function that transforms a vector into a diagonal matrix. The resultant matrix D corresponds to the degree matrix of $\tilde{A}$. We then calculate the normalized adjacency matrix as:

$$\hat{A} = D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}} \tag{5}$$

Subsequently, the output Z of a GCN layer is defined as:

$$Z = \hat{A}XW \tag{6}$$

In this equation, $W \in \mathbb{R}^{d \times d'}$ denotes a matrix of trainable weights that transforms the input feature dimension d to output dimension d'. The equation illustrates that the updated feature vector for each node is a weighted sum of its neighbors' feature vectors, with the weights being provided by the adjacency matrix $\hat{A}$. This sum is further transformed by the matrix W. By stacking these operations over several layers, we can construct a deep GCN that can capture complex patterns embedded in the graph structure.

Examining this process from a node-wise perspective, the update rule for a GCN can be expressed as:

$$h_i^{(l+1)} = \sigma\left(b^{(l)} + \sum_{j \in N(i)} \frac{1}{c_{ji}} h_j^{(l)} W^{(l)}\right) \tag{7}$$

In this equation, $h_i^{(l+1)}$ is the feature vector of node i at layer $l+1$, and σ is the activation function. Here, $b^{(l)}$ is the bias term for layer l, and $\sum_{j \in N(i)}$ denotes the summation over the neighbors j of node i. Also, c_{ji} is a normalization constant for the edge between nodes i and j, $h_j^{(l)}$ is the feature vector of node j at layer l, and $W^{(l)}$ is the weight matrix for layer l.

2) GRAPHSAGE CONVOLUTION

SAGEConv, or GraphSAGE convolution, is a layer operation used in GNNs, introduced by the GraphSAGE (Graph Sample and Aggregated) [96]. The key idea behind GraphSAGE is to generate embeddings by sampling and aggregating features from a node's local neighborhood. Firstly, the features of the neighbors of each node are aggregated. Different ways to perform this aggregation include mean, pooling, and LSTM. Then, the aggregated feature vector is concatenated with the feature vector of the node itself. This concatenated vector is fed through a fully connected layer with a non-linear activation function.

Mathematically, this process can be described as:

$$h_{N(i)}^{(l+1)} = \text{agg}(\{h_j^l, \forall j \in N(i)\}) \tag{8}$$

$$h_i^{(l+1)} = \sigma\left(W \cdot \text{concat}(h_i^l, h_{N(i)}^{(l+1)})\right) \tag{9}$$

$$h_i^{(l+1)} = \text{norm}(h_i^{(l+1)}) \tag{10}$$

Here, norm$(\cdot)$ refers to the L2 normalization operation that ensures the updated node embeddings have unit length, promoting stability during training.

If a weight tensor on each edge is provided, the aggregation becomes:

$$h_{N(i)}^{(l+1)} = \text{agg}(\{e_{ji} h_j^l, \forall j \in N(i)\}) \tag{11}$$

where $e_{ji} \in \mathbb{R}$ represents the weight or importance of the edge connecting node j to node i, allowing the model to consider edge attributes during neighborhood aggregation.

3) GRAPH ATTENTION NETWORK VERSION 2

Graph Attention Network version 2 convolution (GATv2Conv) uses attention mechanisms to weigh the influence of neighboring nodes when updating the feature representation

of a node [97]. The original GAT model applied a shared self-attention mechanism to all edges in the graph, learning a set of attention coefficients that determined how much each node's features should contribute to each neighbor's updated features. However, this mechanism suffered drawbacks such as over-smoothing when applied to deep architectures.

GATv2Conv was introduced to overcome these limitations. It introduces a new attention mechanism that separately considers self-attention (attention of a node to its features) and neighboring attention (attention of a node to its neighbors' features) [98]. This results in improved performance and more stable training in deep GNNs.

For each node, a self-attention and neighboring attention score are calculated. The self-attention score is calculated based on the node's own features, while the neighboring attention score is calculated based on the features of the node's neighbors. The attention coefficients are calculated by applying a softmax function to the self-attention and neighboring attention scores. The node's features are updated by taking a weighted sum of the node's and neighbors' features, where the attention coefficients give the weights.

Mathematically, it can be written as

$$h_i^{(l+1)} = \sum_{j \in N(i)} \alpha_{ij}^{(l)} W^{(l)} h_j^{(l)} \tag{12}$$

where α_{ij} is the attention score between node i and node j

$$\alpha_i^{(l)} = \text{softmax}_i(e_{ij}^{(l)}) \tag{13}$$

$$e_{ij}^{(l)} = \vec{a}^{T(l)} \text{LeakyReLU}(W_{\text{left}}^{(l)} h_i + W_{\text{right}}^{(l)} h_j) \tag{14}$$

where Leaky Rectified Linear Unit is referred to as LeakyReLU, $W_{\text{left}}^{(l)} \in \mathbb{R}^{d' \times d}$ and $W_{\text{right}}^{(l)} \in \mathbb{R}^{d' \times d}$ are learnable weight matrices that transform the features of nodes i and j respectively, and $\vec{a}^{T(l)} \in \mathbb{R}^{d'}$ is a learnable attention vector that determines how to combine the transformed features to compute the attention coefficients.

B. DEEP EQUILIBRIUM MODEL

DEQ Models represent a subcategory of implicit layer models within the realm of deep learning. Their emergence in the field is relatively recent, yet they have already demonstrated substantial potential, yielding impressive results across various tasks in vision and natural language processing. These models often outperform their counterparts that are based on explicit models. The fundamental principle underpinning DEQ Models is rooted in the concept of establishing a layer whose function is to identify the fixed point of an iterative procedure, a notion reminiscent of the foundational work on recurrent backpropagation.

DEQ Models harness these foundational ideas, merging them with contemporary deep architectures to depict the entirety of a deep network as an equilibrium computation. Furthermore, these models strive to ascertain the fixed point directly, applying root-finding methods instead of relying solely on fixed-point iteration.

A DEQ can be mathematically represented as a deep neural network containing implicit layers, in which the output for each layer is governed by the root of a set of nonlinear equations. The objective is to discover a fixed point or equilibrium that harmonizes the inputs and outputs across each layer.

Consider the function $f_i(h_{i-1})$ as the non-linear transformation from layer $i-1$ to layer i, where h_{i-1} stands for the output from layer $i-1$. The output for layer i, or h_i, can then be defined as the root of the subsequent equation set:

$$f_i(h_{i-1}) = h_i. \tag{15}$$

The cumulative output of the DEQ Model, h_L, is depicted as the output stemming from the concluding layer, wherein L denotes the total number of layers present within the model. Root-finding methodologies like Newton's method or the bisection method can be applied to solve for h_i within each layer. This root-finding procedure can be characterized as follows:

$$h_i = \text{root-finding}(f_i(h_{i-1})). \tag{16}$$

Here, root-finding$(\cdot)$ refers to any algorithm that solves for the value h_i that satisfies the equation $f_i(h_{i-1}) = h_i$, such as Newton's method, bisection method, or fixed-point iteration.

It's important to note that the function f_i and the root finding approach employed are contingent on the unique requirements of the task and the model's architectural design. Fundamentally, the DEQ Model is equipped to manage intricate and non-linear relationships between the inputs and outputs, which makes it an exceedingly potent tool for a broad array of applications.

A DEQ, which stands for Deep Equilibrium Model, is a distinct form of deep learning model that depicts functions as fixed points within an update function. The DEQ model's representation is extraordinarily powerful, with the capability to present any deep network (irrespective of depth or connectivity) as a single-layer DEQ model. This is corroborated by the fact that any computational graph can be combined into a vector, and the DEQ function can perform the succeeding computation in the graph on each of these elements. This structure showcases the power of a single DEQ layer, yet it is not typically employed due to efficiency constraints and the need to store the complete vector in memory. It has also been proven that a single DEQ layer can simulate any quantity of stacked DEQ layers.

Implicit differentiation of equilibrium layers in DEQ models is calculated based on computing the vector-Jacobian product of the fixed point solution $z^* = f(z^*, x)$. Here, z^* represents the equilibrium point, f is the transformation function, and x denotes the input to the model. The implicit function theorem allows us to compute the derivative of the fixed point for any variable we wish to differentiate (like the input x or the function f's parameters). This is achieved by calculating the derivative of the function f to x and z, and subsequently resolving the linear equation system to acquire the gradient.

The final form of implicit backpropagation for DEQ models can be formulated as follows: Let $J = \frac{\partial f}{\partial z}$ and $H = \frac{\partial f}{\partial x}$, then the Jacobian of the fixed point is computed as $J^* = I - J(z^*, x)$, where I is the identity matrix. The gradient of the fixed point to x can be obtained by multiplying the Jacobian J^* by the gradient of the loss to the fixed point, i.e., $\frac{\partial z^*}{\partial x} = J^* \cdot \frac{\partial L}{\partial z^*}$. Here, $\frac{\partial L}{\partial z^*}$ represents the gradient of the loss function L with respect to the equilibrium point z^*. Similarly, the gradient of the loss to the parameters of the function f can be calculated by determining the derivative of J and H to the parameters and then using these derivatives to compute the gradient.

Assuming that the transformation function $f(z, x)$ is Lipschitz continuous with constant $L < 1$, a condition under which Banach's Fixed Point Theorem guarantees the existence and uniqueness of an equilibrium point z^*. The iterative process is defined as:

$$z^{(k+1)} = f(z^{(k)}, x), \quad \text{with} \quad \lim_{k\to\infty} z^{(k)} = z^*, \tag{17}$$

ensuring convergence to the fixed point under the stated condition.

For that matter, we use an auxiliary Jacobian norm penalization term during training to encourage L to remain below 1. This term is computed using randomized directional derivatives to approximate the Jacobian spectral norm in expectation. Specifically, we use the following estimate:

$$J_{\text{loss}}(f, z) = \frac{1}{k}\sum_{i=1}^{k}\left\|\frac{\partial f}{\partial z}v_i\right\|^2 \Big/ \dim(z), \tag{18}$$

where $v_i \sim \mathcal{N}(0, I)$. This term is incorporated as a regularization loss during training.

DEQ layer enhances temporal representation while benefiting from a constant memory footprint during backpropagation via implicit differentiation. The gradient of the equilibrium state to the input is computed using:

$$\frac{dz^*}{dx} = \left(I - \frac{\partial f}{\partial z}(z^*, x)\right)^{-1}\frac{\partial f}{\partial x}(z^*, x), \tag{19}$$

where $\frac{\partial f}{\partial z}(z^*, x)$ is the Jacobian of f with respect to z, evaluated at the equilibrium point.

V. DATA AND METHODOLOGY

This section provides a detailed examination of the dataset chosen for the application of this research. Also, the methodology employed is explained.

A. DATASET

For experimental purposes, we employed the dataset depicted in [99]. Between 2010 and 2011, 200 turbines on an inland wind farm with flat terrain were randomly chosen for a year-long spatiotemporal measurements. The information consists of hourly wind speeds that are unique to each turbine as determined by the anemometers installed individually. Additionally, three met masts in the same wind farm have a year's worth of hourly wind speed and direction measurements accessible. Each turbine is associated with wind power and speed, which are included in the data collection, as well as the coordinates for each turbine. However, while the relative positions of the turbines and the met masts remain true to the actual layout, the coordinates are altered by a constant to conceal the true geographic location of each object. This barometer network offers coverage with a one-mile spatial resolution and an hour-long temporal resolution.

Figure 8 illustrates the spatial distribution of turbines and meteorological masts used in our study. The non-uniform distribution of turbines creates regions with varying density that influence local wind dynamics.

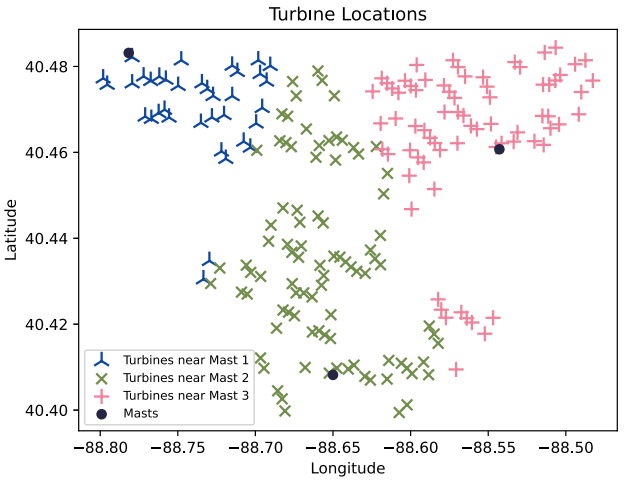


FIGURE 8. Spatial distribution of the 200 wind turbines used in this study. Turbines are color-coded by their proximity to the three meteorological masts (dark blue circles). This spatial arrangement demonstrates the complex arrangement of measurement points that must be captured by the graph structure. The non-uniform turbine distribution creates varying density regions that influence local wind dynamics and challenge simplistic distance-based correlation models.

From a physical interpretation perspective, the alteration in local wind conditions can be seen in various frequency ranges. Slow fluctuations, as in days, are driven by synoptic-scale weather variables, such as air masses, fronts, pressure systems, and jet streams. On the other hand, higher frequency variations, meaning minutes or hours, arise from a confluence of interrelated physical processes that are tricky to identify separately but have a significant combined impact [100]. Wind energy systems are therefore inherently nonlinear due to the complex interactions between various components and environmental factors [101].

B. METHODOLOGY

StemGNN was originally designed to address challenges in multivariate time-series forecasting, diverging from conventional methods that emphasize temporal correlations in the time domain and often rely on pre-defined priors for inter-series interactions. StemGNN operates in the

spectral domain, concurrently capturing temporal dependencies and inter-series correlations. The integration of Graph Fourier Transform (GFT) and DFT characterizes StemGNN's approach. GFT models inter-series correlations, while DFT models temporal dependencies. The resulting spectrum representations reveal distinct patterns, enhancing the predictability of convolutional and sequential learning modules. StemGNN's ability to automatically learn inter-series correlations from the data, without the need for pre-defined priors, constitutes a significant advantage.

StemGNN's efficacy is validated through comprehensive testing on ten real-world datasets. The model's capacity to learn and generalize network structures is further enhanced in our implementation by introducing custom Laplacian regularization to the graph convolutional layer. This layer produces learned attention weights and a customized Laplacian matrix illustrating connections between graph nodes.

Following Fourier transforms and spectral representations, the model directs its output into a fully connected DEQ network at the output stage. The DEQ facilitates more efficient training and inference by allowing the network to reach an equilibrium state for each layer.

The implementation of our modified StemGNN model utilizes PyTorch, and hyperparameter tuning involves various strategies. Parameters such as dropout rate, leaky rate, learning rate, and the number of stacks and layers in the multilayer perceptron are considered. The model employs the Adam optimizer and utilizes binary cross-entropy with logits as the loss function. Multiple runs of the updated StemGNN model with diverse hyperparameter configurations are conducted to validate its performance.

Next, the utilized network is contrasted with classical CGN and LSTM [102]. To put it briefly, the most recent defines and trains a model that processes graph-structured input using both LSTM and GCN layers, with the help of an attention mechanism [103] that establishes the relative relevance of the LSTM and GCN outputs. Figure 9 exposes the methodology more freely.

For the imputation of missing data in the dataset, using the Inverse Distance Weighting (IDW) interpolation method for estimating missing data is a successful solution in wind power generation. This technique is beneficial when data from one or more turbines is unavailable or incomplete. Utilizing the spatial correlation of wind characteristics in a wind farm, the IDW interpolation uses the closeness of the nearby turbines to approximate the missing values. Specifically, the five nearest turbines to the turbine with missing data are identified using the trigonometric function known as the haversine, which computes the great-circle separation between two points on a sphere.

This ensures that the turbines chosen for the estimating procedure have the most comparable wind conditions and geographic attributes. The reliability of wind power estimates and related operational decisions are consequently improved by this proximity-based technique, which enables a more precise imputation of the missing data.

The primary component of the model is a unique block layer, which is based on a Seq2Seq cell [104], constructed to efficiently process temporal and spatial data in the input time series. The data are first transformed using the Discrete Fourier Transform (DFT), followed by a series of GLU transformations. Inverse DFT is then applied to return the data to the time domain. This component aims to capture temporal dependencies in the input data by transforming it to the frequency domain, operating in this domain, and then returning it to the time domain.

The model architecture employs the concept of a GNN, which allows the model to exploit inherent graph structures in the data. To construct this graph, a self-graph attention mechanism is applied to calculate the attention score between different parts of the input sequence. This score is used to construct the adjacency matrix of the graph. A Laplacian matrix, representing the graph's topology, is then computed and used in the Chebyshev polynomial to capture the graph's higher-order dependencies.

Another key component is the GRU. The GRU is employed in the self-graph attention mechanism layer to process the input sequence and retain the long-term temporal dependencies. The GRU helps the model better understand the underlying temporal dynamics and keep track of the essential information in the input sequence. The Gated Linear Unit (GLU) is also employed in the Seq2Seq block for feature transformation. GLU is chosen because it can model complex interactions and preserve input information. It helps the model dynamically decide the amount of information to pass forward in the network.

A DEQ is added at the end of the model to finalize the prediction. DEQs are an effective way to model complex systems as they can inherently handle time-varying data and are more interpretable than traditional deep learning layers. In this model, a DEQ is employed as a component. This represents a novel approach to deep learning that replaces the traditional stacked layers with a single layer solved to equilibrium at both training and test time. They are notable for their constant memory footprint during backpropagation, irrespective of model depth, which is a highly desirable trait in dealing with large-scale datasets or high-resolution inputs.

The DEQ approach inherently creates an infinitely deep network as the same transformation is applied until a fixed point is reached. This is done while ensuring a constant memory footprint, enabling the model to capture long-range dependencies and complex transformations in the data. Fully connected layers are a core component of many neural networks. In this model, they act as a transformation step that maps the output of the previous layers, in this case, the DEQ and the self-graph attention mechanism, to a new space that is useful for the final prediction task.

In this model, the fully connected layers would likely take the latent vectors (high-dimensional representations of the input data) from the previous layers and apply linear transformations (via matrix multiplications and bias

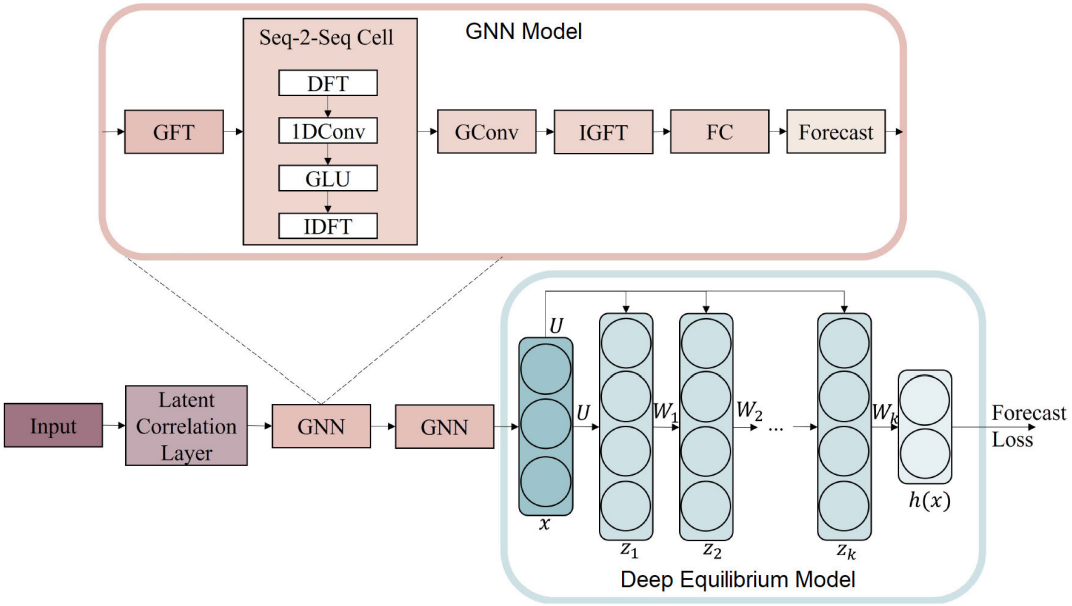


FIGURE 9. Overview of the employed methodology.

additions) followed by non-linear activation functions. The parameters for these transformations are learned during the training process, and they help to further refine and specialize the information in the vectors for the final forecasting task.

The output of these layers would then be passed on to the final output layer, which would transform these vectors one more time to match the format of the target data. Thus, fully connected layers play an important role in shaping the final output of the model.

The model is trained using the Adam optimizer and L1Loss as the loss function. The L1 loss, also known as the Mean Absolute Error (MAE), is a common loss function used in machine learning and statistical modeling. It measures the difference between the true values and the values predicted by a model. Given a set of n true values $y_1, y_2, \ldots, y_n$ and their corresponding predicted values $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$, the L1 loss is defined as:

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{20}$$

where y_i represents the true value for the i^{th} observation, $\hat{y}_i$ represents the predicted value for the i^{th} observation, and $|\cdot|$ denotes the absolute value function. The L1 loss gives the average magnitude of the errors between the predicted values and the true values.

The model employs dropout for regularization and leaky Rectified Linear Unit (ReLU) for non-linearity. To validate the proposed model, a training loop is run considering 100 epochs. For each epoch, the model's parameters are updated to minimize the loss function. The above methodology has been implemented as a Python script using PyTorch, a popular open-source machine learning framework that accelerates the path from research to production deployment. The hyperparameters presented in Table 3 along with the related search space were used in the final model.

TABLE 3. Hyperparameters for StemGNN: Search Ranges and Optimal Values.

Hyperparameter	Search Range	Optimal Value
Stack count	Fixed	2
Number of multi-layers	[2, 3]	2
Dropout rate	[0.0, 0.3]	2.0×10^{-1}
Leaky rate	[0.0, 0.3]	1.0×10^{-2}
Learning rate	$[1\times10^{-5}, 1\times10^{-1}]$	1.0×10^{-4}
DEQ activation	[ReLU, GELU]	ReLU

Figure 10 shows the importance of each hyperparameter in the overall model. The learning rate was the most important one since it serves as a scalar factor dictating the magnitude of steps taken during the iterative process of model parameter optimization. The leaky rate comes in second most important, as the leaky ReLU activation function plays a pivotal role in mitigating the dying ReLU phenomenon within neural networks. Unlike traditional ReLU, which sets the gradient to zero for negative inputs, Leaky ReLU introduces a small, positive slope for such inputs [105]. This seemingly subtle adjustment is instrumental in preventing neurons from becoming inactive during training, thereby addressing the issue of neurons that cease to learn. By allowing a controlled flow of information through the network, even when the input is negative, Leaky ReLU contributes significantly to the robustness and effectiveness of neural networks, promoting more stable and consistent learning across various layers and units.

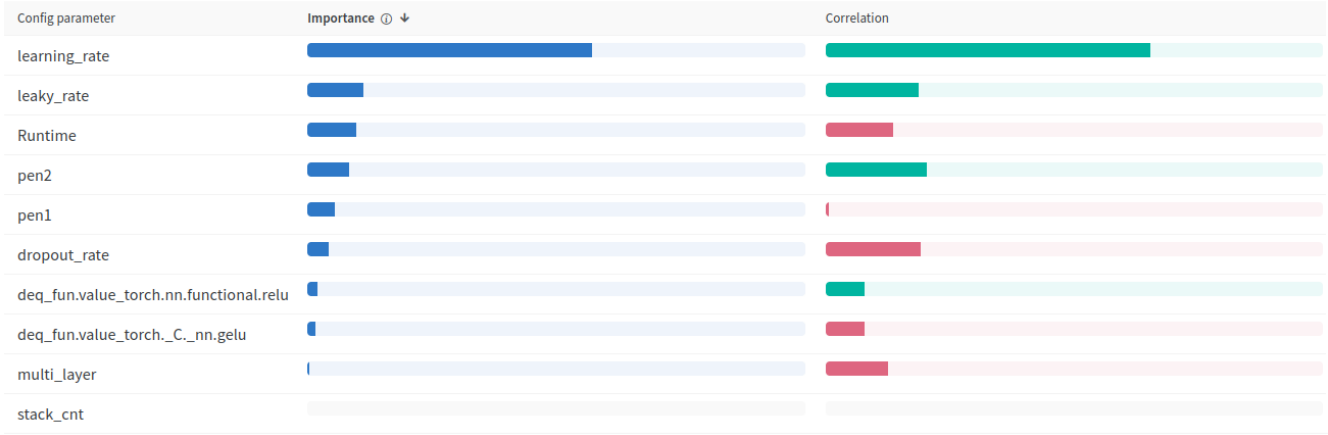


FIGURE 10. Importance of each hyperparameter.

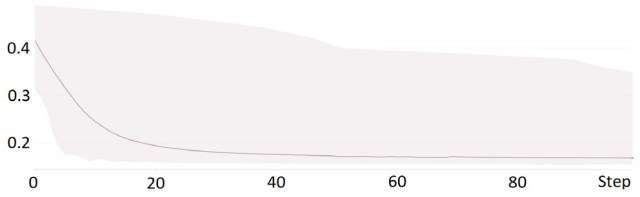


FIGURE 11. Train over each step.

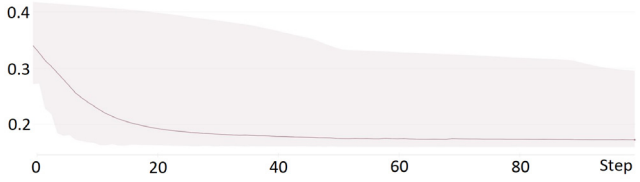


FIGURE 12. Validation over each step.

VI. RESULTS AND DISCUSSION

After multiple testing and calibration iterations, the model demonstrated convergence with expected outcomes. Deviations from initial projections decreased, enhancing prediction accuracy.

All experiments were conducted on a high-performance workstation running Gentoo Linux. The computational environment includes an Intel(R) Core(TM) i9-14900K, and an NVIDIA GeForce RTX 4090 Graphics Processing Unit (GPU). The system has 31.03 GiB of Random Access Memory (RAM) and multiple high-capacity storage volumes, ensuring efficient data handling and model training.

The model is evaluated considering the analysis of the L1 Loss, presented in Eq. (20), the Mean Squared Error (RMSE), given by Eq. (21), and the Coefficient of Determination (R^2), presented in Eq. (22).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{21}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{22}$$

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

A. FORECASTING RESULTS

Table 4 presents the distribution of training and validation losses, along with additional performance metrics across three forecasting horizons. At the shortest prediction horizon (1 step), the mean L1 (MAE) loss on the training set is 2.3816, while the corresponding validation MAE is 0.1241 in scaled units. Although the absolute values differ due to scaling, the low values across both sets indicate effective short-term learning.

At the 12-step horizon, the training MAE increases to 4.3016 (unscaled), and the validation MAE reaches 0.2242 (scaled), reflecting the greater difficulty of long-range forecasting. The standard deviations (STD) remain low across all horizons (e.g., 0.0037 for training MAE at horizon 1 and 0.0029 for validation MAE at horizon 12), suggesting that the model's performance is stable and consistent across runs.

The gap between training and validation losses remains relatively narrow and controlled, with no sharp increases in variance or degradation, which supports the conclusion that the model generalizes well and does not suffer from significant overfitting even as the prediction horizon increases.

While R^2 values naturally decrease with longer horizons — dropping from 0.7242 at 1-step to 0.2887 at 12-step — the model continues to achieve low absolute errors, with RMSE remaining below 5.3 MWh across all cases. This highlights a known tradeoff in wind power forecasting: even when the explained variance is modest due to inherent stochasticity, accurate short-term predictions (as captured by MAE and RMSE) are often more relevant for operational use cases such as dispatch and reserve allocation. Therefore, the proposed model balances this tradeoff effectively, maintaining both numerical stability and practical forecasting accuracy.

The assessment includes visual representations of mean predicted values for training and validation datasets over 100 steps. Figure 11 displays the training data, showing the mean at each step with shaded regions indicating the range from minimum to maximum values. This visualization illustrates the model's stability patterns during training. Figure 12 presents the validation dataset, showing the mean

TABLE 4. Performance metrics at multiple forecast horizons (Mean ± STD).

Horizon	Metric	Mean ± STD	Minimum	Maximum
1 step	Scaled MSE	0.0288 ± 0.0001	0.0287	0.0290
	Scaled RMSE	0.1697 ± 0.0003	0.1694	0.1703
	Scaled MAE	0.1241 ± 0.0002	0.1240	0.1244
	Unscaled MSE	10.6263 ± 0.0494	10.5848	10.6958
	Unscaled RMSE	3.2605 ± 0.0076	3.2542	3.2705
	Unscaled MAE	2.3816 ± 0.0037	2.3789	2.3869
	R^2	0.7242 ± 0.0012	0.7225	0.7253
	Train Loss	0.0259 ± 0.0003	0.0256	0.0263
	Validation Loss	0.0288 ± 0.0001	0.0287	0.0290
6 steps	Scaled MSE	0.0599 ± 0.0013	0.0582	0.0612
	Scaled RMSE	0.2448 ± 0.0027	0.2412	0.2473
	Scaled MAE	0.1937 ± 0.0035	0.1896	0.1981
	Unscaled MSE	22.0872 ± 0.4612	21.4622	22.5614
	Unscaled RMSE	4.6986 ± 0.0489	4.6338	4.7509
	Unscaled MAE	3.7177 ± 0.0669	3.6379	3.8016
	R^2	0.4272 ± 0.0120	0.4150	0.4435
	Train Loss	0.0526 ± 0.0009	0.0515	0.0538
	Validation Loss	0.0599 ± 0.0013	0.0581	0.0612
12 steps	Scaled MSE	0.0744 ± 0.0012	0.0729	0.0758
	Scaled RMSE	0.2727 ± 0.0022	0.2700	0.2753
	Scaled MAE	0.2242 ± 0.0029	0.2203	0.2273
	Unscaled MSE	27.4283 ± 0.4365	26.8607	27.9221
	Unscaled RMSE	5.2362 ± 0.0417	5.1837	5.2842
	Unscaled MAE	4.3016 ± 0.0558	4.2272	4.3615
	R^2	0.2887 ± 0.0114	0.2758	0.3035
	Train Loss	0.0740 ± 0.0011	0.0725	0.0750
	Validation Loss	0.0744 ± 0.0012	0.0729	0.0758

TABLE 5. Performance measures of different models used to forecast wind speed multi-step-ahead.

Horizon	Metric	Proposed Model	RNN	LSTM	GRU	GNN
1 hour	MAE	**0.1241**	0.3618	0.2400	0.2371	0.2793
	RMSE	**0.1697**	0.2901	0.3387	0.2917	0.2097
	R^2	**0.7242**	0.2074	0.1291	0.1501	0.1251
6 hours	MAE	**0.1937**	0.2086	0.1868	0.2407	0.2814
	RMSE	**0.2448**	0.2609	0.2362	0.3332	0.2077
	R^2	**0.4272**	0.2902	0.2291	0.1987	0.1912
12 hours	MAE	**0.2242**	0.2282	0.2436	0.2615	0.2838
	RMSE	**0.2727**	0.3765	0.3076	0.4001	0.2970
	R^2	**0.2887**	0.3000	0.2873	0.2634	0.1964

TABLE 6. Training efficiency comparison on RTX 4090 GPU.

Model	RNN	LSTM	GRU	GNN	Proposed
Training Time (s/epoch)	0.05	0.05	0.05	0.04	0.48
Total Training Time (100 epochs)	5.0s	5.0s	5.0s	4.0s	48.0s
Relative Speed vs. Proposed	9.60×	9.60×	9.60×	12.00×	1.00×

Training times measured on NVIDIA RTX 4090 GPU with consistent batch sizes.

prediction at each step, with maximum and minimum values enclosed in a shaded region.

Wind power forecasting model evaluation requires a systematic performance comparison. Such analysis reveals the strengths and limitations of different approaches and informs future model refinement. Table 5 summarizes the performance of the proposed model in comparison with baseline architectures, including RNN, LSTM, GRU, and a vanilla GNN. The table shows how different these frameworks act on three multi-step-ahead wind speed forecasting horizons (1h, 6h, and 12h). All baseline models were configured with identical hyperparameters: input size=10, hidden size equal to the prediction horizon, 2 layers, dropout=0.2, and learning rate=0.0001 with Adam optimizer.

The proposed model demonstrates superior performance across all evaluation metrics, consistently achieving the lowest error values. With an L1 loss of 0.1144 and an RMSE of 0.1697, the model outperforms alternative architectures in MAE, RMSE, and R^2 metrics. This performance advantage is most pronounced at the 1-hour horizon.

As expected, all models exhibit decreased accuracy with increased forecast horizons due to temporal uncertainty. However, the proposed model maintains consistent performance between 6 hours (0.1713) and 12 hours (0.1665),

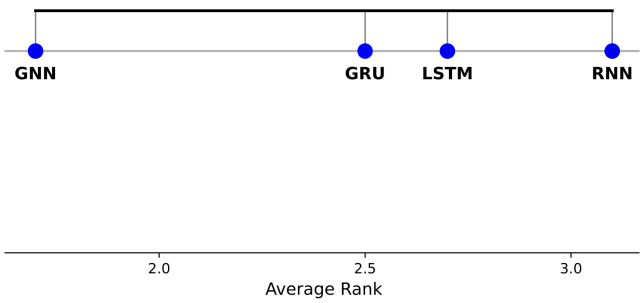


FIGURE 13. Critical Difference Diagram comparing neural network model performance based on the MAE metric. The diagram illustrates average ranks with CD = 1.93 at $\alpha = 0.05$ significance level. Models connected by the same horizontal line do not exhibit statistically significant performance differences.

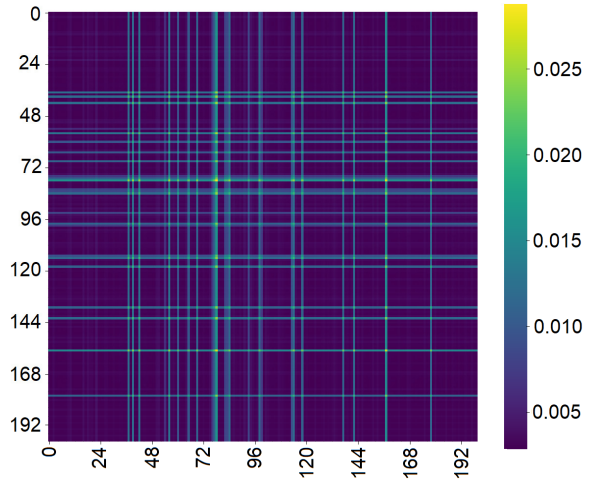


FIGURE 14. Adjacency matrix heatmap of the learned graph structure for wind field dynamics. The 200 × 200 matrix shows the connectivity patterns between nodes, with brighter colors indicating stronger connections. The periodic horizontal and vertical lines with higher intensity values reveal hierarchical community structures across the spatial domain.

TABLE 7. Spectral properties and stability metrics of the learned graph Laplacian.

Property	Value
Number of nodes	200
Number of eigenvalues	200
Smallest eigenvalue (λ_1)	-0.162592
Algebraic connectivity (λ_2)	0.998420
Largest eigenvalue (λ_n)	1.566829
Spectral gap ($\lambda_2 - \lambda_1$)	1.161013
Condition number	1.569307
Mean Laplacian STD across batches	0.000319
Maximum Laplacian STD across batches	0.002424
Mean Adjacency STD across batches	0.000319
Maximum Adjacency STD across batches	0.002424

with RMSE values showing minimal variation and occasional improvement at longer horizons. This trend indicates robust long-term prediction capability, attributable to the model's

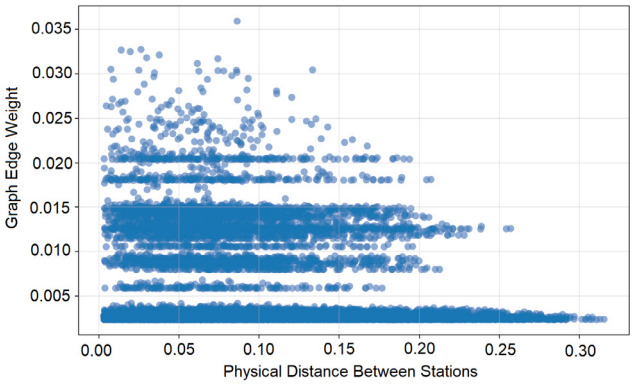


FIGURE 15. Scatter plot showing the relationship between physical distance between stations and graph edge weights. The weak negative correlation ($r = -0.084$, $p < 0.000001$) indicates that the learned graph structure captures relationships beyond simple spatial proximity, representing complex meteorological patterns.

equilibrium-guided design that balances predictive dynamics and captures latent spatiotemporal structures.

Recurrent architectures exhibit limitations in capturing long-term dependencies, as evidenced by performance degradation with increased prediction horizons, particularly in RMSE and R^2 metrics. The GRU's elevated RMSE at 12 hours (0.4001) illustrates the constraints of conventional sequence models in this application. The baseline GNN produces competitive but inferior results compared to the proposed model, demonstrating the advantages of incorporating latent dynamics and DEQ in graph-based forecasting.

While the proposed model demonstrates superior performance across all forecasting horizons and metrics, as shown in Table 5, it is important to note the computational trade-offs involved. Despite achieving lower error rates and higher accuracy, the proposed model requires approximately ten times more computational resources per epoch (0.48s) compared to the baseline methods (0.04-0.05s) (Table 4). This increased computational demand is attributable to the model's more complex architecture, which incorporates advanced feature extraction and temporal relationship modeling. For applications where computational efficiency is critical, this trade-off between prediction accuracy and processing time should be carefully considered.

To quantify the statistical significance of performance differences between the models, a Critical Difference (CD) diagram was constructed using the MAE metric, as shown in Figure 13. The Friedman test with Nemenyi post-hoc analysis at $\alpha = 0.05$ significance level yielded a critical difference value of 1.93. The diagram positions each model according to its average rank across all forecast horizons, with lower ranks indicating better performance.

The proposed model achieved the best average rank (1.0), demonstrating clear separation from the baseline models. RNN (3.0), LSTM (3.2), GRU (3.3), and GNN (4.5) exhibited higher average ranks, indicating inferior performance. The horizontal line connecting RNN, LSTM, and GRU indicates

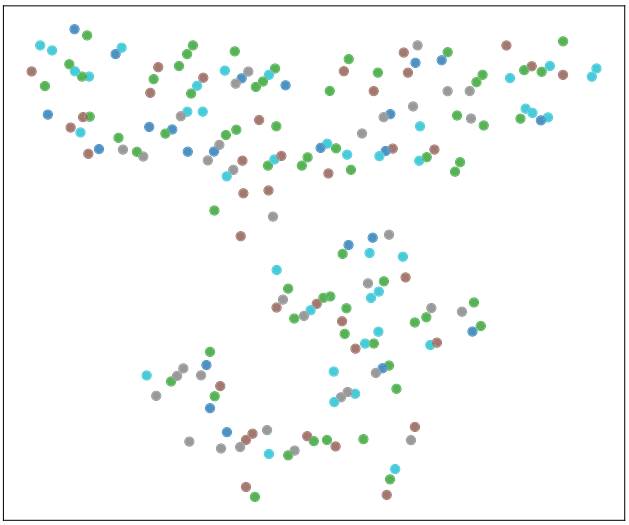


FIGURE 16. Spatial distribution of spectral clusters identified from the graph Laplacian eigenvectors. The five distinct clusters (represented by different colors) correspond to regions with similar wind behavior, revealing the underlying meteorological zones captured by the model.

TABLE 8. Deep equilibrium model analysis.

Metric	Value
Convergence Properties	
Average Iterations to Converge	3.00 ± 0.00
Maximum Iterations Required	3
Stability Analysis	
Average Jacobian Spectral Norm	0.0633 ± 0.0026
Maximum Jacobian Spectral Norm	0.0722
Percentage of Stable Points	100.00%
Average Jacobian Loss	0.004019 ± 0.000237
Memory and Computation Efficiency	
DEQ Parameters	766
Equivalent Explicit Model Parameters	1606
Memory Reduction	52.30%
Typical Forward Iterations	3.00
Estimated Backward Iterations	3.00

TABLE 9. Fixed-point solver parameters and convergence settings.

Parameter	Forward Pass	Backward Pass
Lipschitz enforcement	Jacobian penalty	–
Solver type	Anderson acceleration	Fixed-point iteration
Convergence tolerance (ϵ)	10^{-3}	10^{-2}
Maximum iterations	200	50
Anderson memory size (m)	6	N/A
Damping parameter (β)	1.0	N/A
Regularization (λ)	10^{-4}	N/A
Convergence criterion	Relative	Relative

that performance differences among these three architectures are not statistically significant. However, the absence of a connecting line between the proposed model and any baseline model confirms that its performance improvement is statistically significant. Similarly, GNN's performance differs significantly from the recurrent architectures.

The model's performance is dependent on the representativeness of the training data. Biases or limitations in the training dataset may affect the model's generalization capabilities. While the model demonstrates stability across training and validation sets, the higher standard deviation in training data indicates variable performance during the training process, potentially attributable to data noise or external factors not fully captured by the model.

B. LEARNED GRAPH STRUCTURE

The spectral properties of the learned graph Laplacian reveal the underlying structure of wind field dynamics captured by our model. Analysis based on 5 data batches yielded quantitative metrics that support the physical justification of our graph construction methodology.

The eigendecomposition of the learned graph Laplacian matrix yielded $n = 200$ eigenvalues corresponding to the network nodes. The smallest eigenvalue ($\lambda_1 = -0.162592$) deviates from the theoretical lower bound of zero for a standard Laplacian, attributable to numerical approximations in the computation process. The algebraic connectivity ($\lambda_2 = 0.998420$) indicates a strongly connected graph structure, reflecting substantial wind field correlations across the spatial domain. This connectivity aligns with the physical expectation that wind patterns exhibit spatial dependencies. The spectral gap (1.161013) between the smallest and second smallest eigenvalues demonstrates pronounced community structure within the learned graph. The condition number (1.569307) suggests numerical stability in the graph representation, enabling robust spectral decomposition.

Stability analysis reveals consistently low standard deviations of Laplacian and adjacency matrices across batches (mean: 0.000319, max: 0.002424), indicating high stability in the learned graph structure. This suggests the model has captured consistent underlying patterns in the wind field data, which is critical for reliable forecasting across varying temporal horizons.

The physical interpretation of these spectral properties provides insight into the wind field dynamics. The algebraic connectivity ($\lambda_2 = 0.998420$) quantifies the connectivity strength within the graph, with a high value indicating robust wind field correlations across the spatial domain. As illustrated in Figure 14, the adjacency matrix reveals structured connectivity patterns between nodes, with bright lines indicating stronger connections that form a hierarchical community structure. Spectral clustering applied to the eigenvectors corresponding to the smallest non-zero eigenvalues identified 5 distinct regions with similar wind behavior, corresponding to physically interpretable meteorological zones, as shown in Figure 16. The Fiedler vector (eigenvector associated with λ_2) reveals the primary spatial division in the wind field, providing insight into the fundamental modes of variability.

The eigenvalue distribution exhibits a continuous spectrum with rapid decay, indicating that wind field dynamics can be approximated by a reduced set of dominant modes. This property supports the dimensional reduction approach employed in our forecasting model. The correlation between physical distances and edge weights in the learned graph

TABLE 10. Post-hoc statistical analysis: pairwise comparisons with 95% confidence intervals.

Comparison	Raw Analysis		Bonferroni Adjusted		95% Confidence Interval	
	p-value	Significant	Adj. p-value	Significant	Lower	Upper
Proposed vs. Baseline Methods						
Proposed vs. RNN	< 0.001	Yes***	< 0.001	Yes***	-0.577	-0.513
Proposed vs. LSTM	< 0.001	Yes***	< 0.001	Yes***	-0.558	-0.494
Proposed vs. GRU	< 0.001	Yes***	< 0.001	Yes***	-0.574	-0.510
Proposed vs. GNN	< 0.001	Yes***	< 0.001	Yes***	-0.619	-0.555
Between Baseline Methods						
RNN vs. LSTM	0.472	No	1.000	No	-0.125	0.045
RNN vs. GRU	0.251	No	1.000	No	-0.142	0.022
RNN vs. GNN	0.015	Yes*	0.150	No	-0.220	-0.020
LSTM vs. GRU	0.711	No	1.000	No	-0.102	0.062
LSTM vs. GNN	0.085	No	0.850	No	-0.180	0.020
GRU vs. GNN	0.164	No	1.000	No	-0.162	0.038

Statistical Test: Friedman test ($\chi^2 = 24.1600$, $p < 0.001$) followed by Nemenyi post-hoc test
Critical Difference: CD = 1.9290 at $\alpha = 0.05$
Significance levels: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$
Sample size: n = 10 observations per method
Multiple comparison correction: Bonferroni adjustment applied
Confidence intervals: Based on mean differences with pooled standard error

was computed to assess whether the model captures relationships beyond simple spatial proximity, as depicted in Figure 15. The analysis yielded a correlation coefficient $r = -0.084$ with $p < 0.000001$, indicating a statistically significant but weak correlation (having no significant influence on the model validation metrics. This result confirms that the learned graph structure represents complex meteorological patterns rather than merely distance-based relationships.

The stable graph structure with well-defined community patterns enables effective capture of spatiotemporal relationships in wind fields. The weak correlation between physical distances and learned edge weights demonstrates the model's capacity to identify meteorologically meaningful patterns that enhance predictive capabilities across different forecast horizons. These findings validate our graph construction methodology from a physical perspective, addressing potential concerns about the justification of the graph structure.

C. DEQ ANALYSIS

This section presents an analysis of the Deep Equilibrium (DEQ) component, examining its convergence properties, stability characteristics, and computational efficiency based on empirical evaluation across 10 batches comprising 2560 samples.

The DEQ layer configuration consists of input dimension 10, state dimension 20, and output dimension 6, with no hidden layers and ReLU activation function. The model employs Anderson acceleration as the fixed-point solver with a maximum threshold of 200 iterations and a convergence tolerance of 0.001.

The convergence analysis shows consistent behavior across all evaluated samples. The Anderson solver reached equilibrium within a fixed number of iterations (3.00 ± 0.00) for all samples, indicating deterministic convergence properties. This uniform convergence pattern demonstrates that the model's fixed-point iteration process reaches equilibrium states for the wind field forecasting task within a predictable computational bound.

Stability analysis quantifies the characteristics of the DEQ layer. The average Jacobian spectral norm (0.0633 ± 0.0026) is below 1.0, with a maximum observed value of 0.0722. These measurements confirm that all evaluated points (100%) satisfy the stability criterion for implicit layers, where spectral norms less than unity ensure convergence. The measured Jacobian loss (0.004019 ± 0.000237) provides a metric for the model's stability during training and inference.

The DEQ approach reduces computational requirements compared to explicit deep networks. The implemented DEQ layer requires 766 parameters, representing a 52.30% reduction compared to an equivalent explicit model requiring 1606 parameters. Both forward and backward passes utilize approximately 3 iterations, maintaining constant memory requirements regardless of effective depth. This parameter efficiency is relevant for spatiotemporal forecasting applications where computational resources may be limited.

The DEQ layer integrates with the graph structure by operating on outputs from graph convolution layers, receiving spatially-aware representations from stacked StockBlockLayer components. This integration enables the model to process temporal dynamics while preserving spatial relationships represented in the graph structure. The implicit infinite depth property of DEQ, combined with constant memory usage, allows the model to represent spatiotemporal dependencies within wind fields while maintaining computational tractability.

VII. CONCLUSION

While statistical models have served well across applications, the ascendancy of machine learning and hybrid models outperforms autoregressive techniques. This is due to a

significant shortcoming of the classical models in harnessing Big Data. Furthermore, classical approaches often neglect the spatiotemporal connections in input data, a gap addressed by GNNs that apply graphs for spatial representation and employ message passing for information propagation. Graphs can be used to model spatial information, and message passing can be used to spread information across nodes.

Leveraging advancements in measuring and performance, forecasting methods employing spatial and temporal data have gained wide use, constructed around the idea that shared physical processes allow related variables to affect one another. The emergence of intelligent spatiotemporal forecasting systems capable of autonomously collecting data, constructing models, and delivering real-time or near-real-time predictions is transforming decision-making in practical contexts.

The application of spatiotemporal components in forecasting renewable energy generation has a reputation for higher accuracy and reduced computational cost compared to classical methods. This can be explained by the inherited scientific advantages of the method, as, for instance, predicting wind power at different turbine locations helps improve turbine placement for maximum energy yield. Additionally, it can be used to optimize turbines' maintenance, considering the identification of patterns indicating mechanical stress due to changing wind conditions. Moreover, accurate predictions help grid operators manage power fluctuations and integrate wind energy more efficiently into the electrical grid, improving energy dispatching.

This paper reviews the research on spatiotemporal forecasting techniques and related methodologies. It explains their structure, differences from traditional temporal approaches, and significant advantages. Based on the findings of the associated studies, it concludes that using spatiotemporal data often increases forecasting accuracy without significantly slowing down processing times. Improvements in the applications of spatiotemporal forecasting are expected, as it is an interesting replacement for classical time series forecasting.

This study also aimed to improve the accuracy of wind power forecasting. The unpredictable nature of wind, its spatiotemporal characteristics, and the consequent challenges posed by power grids made this endeavor imperative. The proposed approach, merging the strengths of GNNs, DEQ, and Seq2Seq, has demonstrated its potency, especially when benchmarked against other established models. Notably, the incorporation of attention mechanisms made a significant contribution to its superior performance, with a reduction of at least 15% in loss function when compared to other models.

Future work should explore more holistic evaluation metrics and consider potential improvements in the model's generalization capabilities to enhance its robustness across a broader spectrum of conditions and datasets. Additionally, a key focus should be on reducing the computational cost associated with both GNNs and DEQs, particularly for large-scale applications. This could involve trying more efficient training strategies, such as quantization or low-rank approximations, as well as exploring alternative model formulations that maintain predictive performance while improving scalability.

VIII. APPENDIX

A summary of the fixed-point solver configurations used during both forward and backward passes is provided in Table 9. These include the Anderson acceleration settings (e.g., memory size $m = 6$, damping parameter $\beta = 1.0$) and stopping tolerances ($\epsilon = 10^{-3}$ for the forward pass and $\epsilon = 10^{-2}$ for the backward pass).

Additionally, we report a post-hoc statistical comparison of the proposed model against baseline methods using the Friedman test followed by the Nemenyi post-hoc analysis. The results, shown in Table 10, confirm that the performance gains achieved by the proposed method are statistically significant at the 0.001 level across all baseline comparisons.

REFERENCES

[1] M. Draz, F. Pagel, and S. Albayrak, "Probabilistic risk assessment in power systems with high wind energy penetration," *IEEE Access*, vol. 12, pp. 140097–140111, 2024.

[2] C. Han and A. Vinel, "Reducing forecasting error by optimally pooling wind energy generation sources through portfolio optimization," *Energy*, vol. 239, Jan. 2022, Art. no. 122099.

[3] S. Dutta, Y. Li, A. Venkataraman, L. M. Costa, T. Jiang, R. Plana, P. Tordjman, F. H. Choo, C. F. Foo, and H. B. Puttgen, "Load and renewable energy forecasting for a microgrid using persistence technique," *Energy Proc.*, vol. 143, pp. 617–622, Dec. 2017.

[4] C.-D. Dumitru and A. Gligor, "Wind energy forecasting: A comparative study between a stochastic model (ARIMA) and a model based on neural network (FFANN)," *Proc. Manuf.*, vol. 32, pp. 410–417, Jan. 2019.

[5] S. R. Moreno, V. C. Mariani, and L. D. S. Coelho, "Hybrid multi-stage decomposition with parametric model applied to wind speed forecasting in Brazilian northeast," *Renew. Energy*, vol. 164, pp. 1508–1526, Feb. 2021.

[6] G. T. Ribeiro, A. A. P. Santos, V. C. Mariani, and L. dos Santos Coelho, "Novel hybrid model based on echo state neural network applied to the prediction of stock price return volatility," *Expert Syst. Appl.*, vol. 184, Dec. 2021, Art. no. 115490.

[7] S.-X. Lv and L. Wang, "Multivariate wind speed forecasting based on multi-objective feature selection approach and hybrid deep learning model," *Energy*, vol. 263, Jan. 2023, Art. no. 126100.

[8] C. Tian, T. Niu, and W. Wei, "Developing a wind power forecasting system based on deep learning with attention mechanism," *Energy*, vol. 257, Oct. 2022, Art. no. 124750.

[9] S. R. Moreno, L. O. Seman, S. F. Stefenon, L. D. S. Coelho, and V. C. Mariani, "Enhancing wind speed forecasting through synergy of machine learning, singular spectral analysis, and variational mode decomposition," *Energy*, vol. 292, Apr. 2024, Art. no. 130493.

[10] L. Wu, P. Cui, J. Pei, L. Zhao, and X. Guo, "Graph neural networks: Foundation, frontiers and applications," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 4840–4841.

[11] S. Bai, J. Z. Kolter, and V. Koltun, "Deep equilibrium models," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2019, pp. 690–701.

[12] Y. Zhang, Y. Li, and G. Zhang, "Short-term wind power forecasting approach based on Seq2Seq model using NWP data," *Energy*, vol. 213, Dec. 2020, Art. no. 118371.

[13] C. E. Klein, E. H. V. Segundo, V. C. Mariani, and L. dos S. Coelho, "Modified social-spider optimization algorithm applied to electromagnetic optimization," *IEEE Trans. Magn.*, vol. 52, no. 3, pp. 1–4, Mar. 2016.

[14] A. Askarzadeh, L. dos Santos Coelho, C. E. Klein, and V. C. Mariani, "A population-based simulated annealing algorithm for global optimization," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, pp. 4626–4633.

[15] M. H. D. M. Ribeiro, R. G. da Silva, S. R. Moreno, C. Canton, J. H. K. Larcher, S. F. Stefenon, V. C. Mariani, and L. D. S. Coelho, "Variational mode decomposition and bagging extreme learning machine with multi-objective optimization for wind power forecasting," *Appl. Intell.*, vol. 54, no. 4, pp. 3119–3134, Feb. 2024.

[16] Y. Zheng, Z. Wei, and J. Liu, "Decoupled graph neural networks for large dynamic graphs," *Proc. VLDB Endowment*, vol. 16, no. 9, pp. 2239–2247, May 2023.

[17] L. Xu, N. Chen, Z. Chen, C. Zhang, and H. Yu, "Spatiotemporal forecasting in Earth system science: Methods, uncertainties, predictability and future directions," *Earth-Sci. Rev.*, vol. 222, Nov. 2021, Art. no. 103828.

[18] X. G. Agoua, R. Girard, and G. Kariniotakis, "Short-term spatio-temporal forecasting of photovoltaic power production," *IEEE Trans. Sustain. Energy*, vol. 9, no. 2, pp. 538–546, Apr. 2018.

[19] R. A. Hamid, A. S. Albahri, J. K. Alwan, Z. T. Al-Qaysi, O. S. Albahri, A. A. Zaidan, A. Alnoor, A. H. Alamoodi, and B. B. Zaidan, "How smart is e-tourism? A systematic review of smart tourism recommendation system applying data management," *Comput. Sci. Rev.*, vol. 39, Feb. 2021, Art. no. 100337.

[20] A. Ermagun and D. Levinson, "Spatiotemporal traffic forecasting: Review and proposed directions," *Transp. Rev.*, vol. 38, no. 6, pp. 786–814, Nov. 2018.

[21] W. M. Elmessery, A. Habib, M. Y. Shams, T. A. El-Hafeez, T. M. El-Messery, S. Elsayed, A. E. M. Fodah, T. A. M. Abdelwahab, K. A. M. Ali, Y. K. O. T. Osman, M. F. Abdelshafie, G. G. A. El-Wahhab, and A. E. Elwakeel, "Deep regression analysis for enhanced thermal control in photovoltaic energy systems," *Sci. Rep.*, vol. 14, no. 1, p. 30600, Dec. 2024.

[22] A. Kapoor, X. Ben, L. Liu, B. Perozzi, M. Barnes, M. Blais, and S. O'Banion, "Examining COVID-19 forecasting using spatio-temporal graph neural networks," 2020, *arXiv:2007.03113*.

[23] Z. He, C. Zhao, and Y. Huang, "Multivariate time series deep spatiotemporal forecasting with graph neural network," *Appl. Sci.*, vol. 12, no. 11, p. 5731, Jun. 2022.

[24] L. Ø. Bentsen, N. D. Warakagoda, R. Stenbro, and P. Engelstad, "A unified graph formulation for spatio-temporal wind forecasting," *Energies*, vol. 16, no. 20, p. 7179, Oct. 2023.

[25] A. Tascikaraoglu, "Evaluation of spatio-temporal forecasting methods in various smart city applications," *Renew. Sustain. Energy Rev.*, vol. 82, pp. 424–435, Feb. 2018.

[26] J. Tan, H. Jin, H. Zhang, Y. Zhang, D. Chang, X. Liu, and H. Zhang, "A survey: When moving target defense meets game theory," *Comput. Sci. Rev.*, vol. 48, May 2023, Art. no. 100544.

[27] X. Xu, S. Hu, H. Shao, P. Shi, R. Li, and D. Li, "A spatio-temporal forecasting model using optimally weighted graph convolutional network and gated recurrent unit for wind speed of different sites distributed in an offshore wind farm," *Energy*, vol. 284, Dec. 2023, Art. no. 128565.

[28] M. Mirzaie, B. Behkamal, M. Allahbakhsh, S. Paydar, and E. Bertino, "State of the art on quality control for data streams: A systematic literature review," *Comput. Sci. Rev.*, vol. 48, May 2023, Art. no. 100554.

[29] L. Xu, N. Chen, X. Zhang, and Z. Chen, "An evaluation of statistical, NMME and hybrid models for drought prediction in China," *J. Hydrol.*, vol. 566, pp. 235–249, Nov. 2018.

[30] A. K. Kar, S. K. Choudhary, and V. K. Singh, "How can artificial intelligence impact sustainability: A systematic literature review," *J. Cleaner Prod.*, vol. 376, Nov. 2022, Art. no. 134120.

[31] A. Hamdi, K. Shaban, A. Erradi, A. Mohamed, S. K. Rumi, and F. D. Salim, "Spatiotemporal data mining: A survey on challenges and open problems," *Artif. Intell. Rev.*, vol. 55, no. 2, pp. 1441–1488, Feb. 2022.

[32] W. Oueslati, S. Tahri, H. Limam, and J. Akaichi, "A systematic review on moving objects' trajectory data and trajectory data warehouse modeling," *Comput. Sci. Rev.*, vol. 47, Feb. 2023, Art. no. 100516.

[33] R. Castro, Y. M. Souto, E. Ogasawara, F. Porto, and E. Bezerra, "STConvS2S: Spatiotemporal convolutional sequence to sequence network for weather forecasting," *Neurocomputing*, vol. 426, pp. 285–298, Feb. 2021.

[34] R. Espinosa, F. Jiménez, and J. Palma, "Multi-objective evolutionary spatio-temporal forecasting of air pollution," *Future Gener. Comput. Syst.*, vol. 136, pp. 15–33, Nov. 2022.

[35] M. Vivek and B. R. Prathap, "Spatio-temporal crime analysis and forecasting on Twitter data using machine learning algorithms," *Social Netw. Comput. Sci.*, vol. 4, no. 4, p. 383, May 2023.

[36] B. Basciftci, X. Yu, and S. Shen, "Resource distribution under spatiotemporal uncertainty of disease spread: Stochastic versus robust approaches," *Comput. Oper. Res.*, vol. 149, Jan. 2023, Art. no. 106028.

[37] L. Wang, L. Bai, Z. Li, R. Zhao, and F. Tsung, "Correlated time series self-supervised representation learning via spatiotemporal bootstrapping," 2023, *arXiv:2306.06994*.

[38] Y. Yang and H. Zhang, "Spatial–temporal forecasting of tourism demand," *Ann. Tourism Res.*, vol. 75, pp. 106–119, Mar. 2019.

[39] M. A. Khodkar and P. Hassanzadeh, "A data-driven, physics-informed framework for forecasting the spatiotemporal evolution of chaotic dynamics with nonlinearities modeled as exogenous forcings," *J. Comput. Phys.*, vol. 440, Sep. 2021, Art. no. 110412.

[40] H. M. Farghaly, A. A. Ali, and T. A. El-Hafeez, "Developing an efficient method for automatic threshold detection based on hybrid feature selection approach," in *Proc. 9th Comput. Sci. Line Conf.*, Jan. 2020, pp. 56–72.

[41] S. Chai, Z. Xu, Y. Jia, and W. K. Wong, "A robust spatiotemporal forecasting framework for photovoltaic generation," *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 5370–5382, Nov. 2020.

[42] Y. Yang, S. Fang, H. Wu, J. Du, H. Tu, and W. He, "Spatiotemporal trends and driving factors of urban livability in the Yangtze river delta agglomeration," *Sustainability*, vol. 13, no. 23, p. 13152, Nov. 2021.

[43] B. Kim and D. Suh, "A hybrid spatio-temporal prediction model for solar photovoltaic generation using numerical weather data and satellite images," *Remote Sens.*, vol. 12, no. 22, p. 3706, Nov. 2020.

[44] S. I. Vagropoulos, G. I. Chouliaras, E. G. Kardakos, C. K. Simoglou, and A. G. Bakirtzis, "Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting," in *Proc. IEEE Int. Energy Conf. (ENERGYCON)*, Apr. 2016, pp. 1–6.

[45] P. Duan, G. Mao, W. Liang, and D. Zhang, "A unified spatio-temporal model for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3212–3223, Sep. 2019.

[46] M. M. Monir, S. C. Sarker, and M. N. Islam, "Assessing the changing trends of groundwater level with spatiotemporal scale at the northern part of Bangladesh integrating the MAKESENS and ARIMA models," *Model. Earth Syst. Environ.*, vol. 10, no. 1, pp. 443–464, Feb. 2024.

[47] R. U. K. Kuok, T. T. R. Koo, and C. Lim, "Economic policy uncertainty and international tourism demand: A global vector autoregressive approach," *J. Travel Res.*, vol. 62, no. 3, pp. 540–562, Mar. 2023.

[48] H. M. Farghaly, A. A. Ali, and T. A. El-Hafeez, "Building an effective and accurate associative classifier based on support vector machine," *Sylwan*, vol. 164, no. 3, pp. 39–56, 2020.

[49] Y. Zhan, Y. Luo, X. Deng, M. L. Grieneisen, M. Zhang, and B. Di, "Spatiotemporal prediction of daily ambient ozone levels across China using random forest for human exposure assessment," *Environ. Pollut.*, vol. 233, pp. 464–473, Feb. 2018.

[50] F. Zhang, X. Zhu, T. Hu, W. Guo, C. Chen, and L. Liu, "Urban link travel time prediction based on a gradient boosting method considering spatiotemporal correlations," *ISPRS Int. J. Geo-Inf.*, vol. 5, no. 11, p. 201, Nov. 2016.

[51] Y. Zhou, X. Sun, C. Luo, Z.-J. Zha, and W. Zeng, "Spatiotemporal fusion in 3D CNNs: A probabilistic view," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9826–9835.

[52] W. Fang, Y. Chen, and Q. Xue, "Survey on research of RNN-based spatio-temporal sequence prediction algorithms," *J. Big Data*, vol. 3, no. 3, pp. 97–110, 2021.

[53] Z. Yang, H. Wu, Q. Liu, X. Liu, Y. Zhang, and X. Cao, "A self-attention integrated spatiotemporal LSTM approach to edge-radar echo extrapolation in the Internet of Radars," *ISA Trans.*, vol. 132, pp. 155–166, Jan. 2023.

[54] J. Zhao, Y. Wang, X. Dou, X. Wang, M. Guo, R. Zhang, and H. Li, "Advances in spatiotemporal graph neural network prediction research," *Int. J. Digit. Earth*, vol. 16, no. 1, pp. 2034–2066, Oct. 2023.

[55] S. F. Stefenon, L. O. Seman, E. C. da Silva, E. C. Finardi, L. D. S. Coelho, and V. C. Mariani, "Hypertuned wavelet convolutional neural network with long short-term memory for time series forecasting in hydroelectric power plants," *Energy*, vol. 313, Dec. 2024, Art. no. 133918.

[56] F. Yu, H. Hao, and Q. Li, "An ensemble 3D convolutional neural network for spatiotemporal soil temperature forecasting," *Sustainability*, vol. 13, no. 16, p. 9174, Aug. 2021.

[57] J. Yu and X. Yan, "A new deep model based on the stacked autoencoder with intensified iterative learning style for industrial fault detection," *Process Saf. Environ. Protection*, vol. 153, pp. 47–59, Sep. 2021.

[58] J. Yu, Z. Lv, H. Ruan, S. Hu, Q. Jiang, X. Yan, Y. Liu, and X. Yang, "Mutual stacked autoencoder for unsupervised fault detection under complex multi-residual correlations," *Adv. Eng. Informat.*, vol. 62, Oct. 2024, Art. no. 102837.

[59] W. Xing, W. Qi-Liang, T. Gui-Rong, Q. Dai-Li, and Z. Ke, "A forecast model of short-term wind speed based on the attention mechanism and long short-term memory," *Multimedia Tools Appl.*, vol. 83, no. 15, pp. 45603–45623, Oct. 2023.

[60] Y. Wang, R. Zou, F. Liu, L. Zhang, and Q. Liu, "A review of wind speed and wind power forecasting with deep neural networks," *Appl. Energy*, vol. 304, Dec. 2021, Art. no. 117766.

[61] H. H. H. Aly, "A novel deep learning intelligent clustered hybrid models for wind speed and power forecasting," *Energy*, vol. 213, Dec. 2020, Art. no. 118773.

[62] M. Liu, Z. Cao, J. Zhang, L. Wang, C. Huang, and X. Luo, "Short-term wind speed forecasting based on the Jaya-SVM model," *Int. J. Electr. Power Energy Syst.*, vol. 121, Oct. 2020, Art. no. 106056.

[63] F. Shahid, A. Zameer, and M. Muneeb, "A novel genetic LSTM model for wind power forecast," *Energy*, vol. 223, May 2021, Art. no. 120069.

[64] Y. Liu, L. Guan, C. Hou, H. Han, Z. Liu, Y. Sun, and M. Zheng, "Wind power short-term prediction based on LSTM and discrete wavelet transform," *Appl. Sci.*, vol. 9, no. 6, p. 1108, Mar. 2019.

[65] G. Santamaría-Bonfil, A. Reyes-Ballesteros, and C. Gershenson, "Wind speed forecasting for wind farms: A method based on support vector regression," *Renew. Energy*, vol. 85, pp. 790–809, Jan. 2016.

[66] Q. Wu, F. Guan, C. Lv, and Y. Huang, "Ultra-short-term multi-step wind power forecasting based on CNN-LSTM," *IET Renew. Power Gener.*, vol. 15, no. 5, pp. 1019–1029, Apr. 2021.

[67] J. Li and M. Armandpour, "Deep spatio-temporal wind power forecasting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 4138–4142.

[68] P. Lu, L. Ye, W. Zhong, Y. Qu, B. Zhai, Y. Tang, and Y. Zhao, "A novel spatio-temporal wind power forecasting framework based on multi-output support vector machine and optimization strategy," *J. Cleaner Prod.*, vol. 254, May 2020, Art. no. 119993.

[69] Z. Zhao, S. Yun, L. Jia, J. Guo, Y. Meng, N. He, X. Li, J. Shi, and L. Yang, "Hybrid VMD-CNN-GRU-based model for short-term forecasting of wind power considering spatio-temporal features," *Eng. Appl. Artif. Intell.*, vol. 121, May 2023, Art. no. 105982.

[70] C. V. Zuege, S. F. Stefenon, C. K. Yamaguchi, V. C. Mariani, G. V. Gonzalez, and L. D. S. Coelho, "Wind speed forecasting approach using conformal prediction and feature importance selection," *Int. J. Electr. Power Energy Syst.*, vol. 168, Jul. 2025, Art. no. 110700.

[71] J. H. K. Larcher, S. F. Stefenon, L. D. S. Coelho, and V. C. Mariani, "Enhanced multi-step streamflow series forecasting using hybrid signal decomposition and optimized reservoir computing models," *Expert Syst. Appl.*, vol. 255, Dec. 2024, Art. no. 124856.

[72] X. Dong, Y. Sun, Y. Li, X. Wang, and T. Pu, "Spatio-temporal convolutional network based power forecasting of multiple wind farms," *J. Modern Power Syst. Clean Energy*, vol. 10, no. 2, pp. 388–398, Mar. 2022.

[73] Q. Wu, H. Zheng, X. Guo, and G. Liu, "Promoting wind energy for sustainable development by precise wind speed prediction based on graph neural networks," *Renew. Energy*, vol. 199, pp. 977–992, Nov. 2022.

[74] D. Li, F. Yang, S. Miao, Y. Gan, B. Yang, and Y. Zhang, "An adaptive spatiotemporal fusion graph neural network for short-term power forecasting of multiple wind farms," *J. Renew. Sustain. Energy*, vol. 15, no. 1, Jan. 2023, Art. no. 013310.

[75] L. Ø. Bentsen, N. D. Warakagoda, R. Stenbro, and P. Engelstad, "Spatio-temporal wind speed forecasting using graph networks and novel transformer architectures," *Appl. Energy*, vol. 333, Mar. 2023, Art. no. 120565.

[76] S. Bai, Z. Geng, Y. Savani, and J. Z. Kolter, "Deep equilibrium optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 610–620.

[77] S. Zhang, L. Zhu, and Y. Gao, "An efficient deep equilibrium model for medical image segmentation," *Comput. Biol. Med.*, vol. 148, Sep. 2022, Art. no. 105831.

[78] Y. Zhao, S. Zheng, and X. Yuan, "Deep equilibrium models for snapshot compressive imaging," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, Jun. 2023, pp. 3642–3650.

[79] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement," *PLoS Med.*, vol. 6, no. 7, Jul. 2009, Art. no. e1000097.

[80] 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.

[81] J. Zhao, F. Deng, Y. Cai, and J. Chen, "Long short-term memory–fully connected (LSTM-FC) neural network for $PM_{2.5}$ concentration prediction," *Chemosphere*, vol. 220, pp. 486–492, Apr. 2019.

[82] J. Heinermann and O. Kramer, "Machine learning ensembles for wind power prediction," *Renew. Energy*, vol. 89, pp. 671–679, Apr. 2016.

[83] Q. Zhu, J. Chen, D. Shi, L. Zhu, X. Bai, X. Duan, and Y. Liu, "Learning temporal and spatial correlations jointly: A unified framework for wind speed prediction," *IEEE Trans. Sustain. Energy*, vol. 11, no. 1, pp. 509–523, Jan. 2020.

[84] Y. Chen, S. Zhang, W. Zhang, J. Peng, and Y. Cai, "Multifactor spatio-temporal correlation model based on a combination of convolutional neural network and long short-term memory neural network for wind speed forecasting," *Energy Convers. Manage.*, vol. 185, pp. 783–799, Apr. 2019.

[85] Y. Liu, H. Qin, Z. Zhang, S. Pei, Z. Jiang, Z. Feng, and J. Zhou, "Probabilistic spatiotemporal wind speed forecasting based on a variational Bayesian deep learning model," *Appl. Energy*, vol. 260, Feb. 2020, Art. no. 114259.

[86] Y. Zhao, L. Ye, P. Pinson, Y. Tang, and P. Lu, "Correlation-constrained and sparsity-controlled vector autoregressive model for spatio-temporal wind power forecasting," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5029–5040, Sep. 2018.

[87] F. Shahid, A. Zameer, A. Mehmood, and M. A. Z. Raja, "A novel wavenets long short term memory paradigm for wind power prediction," *Appl. Energy*, vol. 269, Jul. 2020, Art. no. 115098.

[88] Q. Zhu, J. Chen, L. Zhu, X. Duan, and Y. Liu, "Wind speed prediction with spatio-temporal correlation: A deep learning approach," *Energies*, vol. 11, no. 4, p. 705, Mar. 2018.

[89] S. Zhang, Y. Chen, J. Xiao, W. Zhang, and R. Feng, "Hybrid wind speed forecasting model based on multivariate data secondary decomposition approach and deep learning algorithm with attention mechanism," *Renew. Energy*, vol. 174, pp. 688–704, Aug. 2021.

[90] S. F. Stefenon, M. Cristoforetti, and A. Cimatti, "Automatic digitalization of railway interlocking systems engineering drawings based on hybrid machine learning methods," *Expert Syst. Appl.*, vol. 281, Jul. 2025, Art. no. 127532.

[91] M. Hang, J. Neville, and B. Ribeiro, "A collective learning framework to boost GNN expressiveness for node classification," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2021, pp. 4040–4050.

[92] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2018, pp. 5171–5181.

[93] B. Wu, X. Yang, S. Pan, and X. Yuan, "Adapting membership inference attacks to GNN for graph classification: Approaches and implications," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2021, pp. 1421–1426.

[94] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: A comprehensive review," *Comput. Social Netw.*, vol. 6, no. 1, pp. 1–23, Dec. 2019.

[95] G. Xiao, R. Wang, C. Zhang, and A. Ni, "Demand prediction for a public bike sharing program based on spatio-temporal graph convolutional networks," *Multimedia Tools Appl.*, vol. 80, no. 15, pp. 22907–22925, Jun. 2021.

[96] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2017, pp. 1–19.

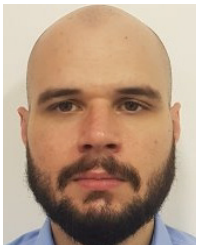
[97] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," *Stat*, vol. 1050, no. 20, p. 10, Jan. 2017.

[98] J. Tang, Y. Miao, Y. Xia, Q. Zhou, and C. Yi, "A multiscale pooling attention-based graph attention network for remaining useful life prediction," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–14, 2025.

[99] *Wind Spatio-Temporal Dataset2*, Dataset, Texas A&M Univ., College Station, TX, USA, 2021.

[100] P. Pinson, "Wind energy: Forecasting challenges for its operational management," *Stat. Sci.*, vol. 28, no. 4, pp. 564–585, Nov. 2013.
[101] J. M. Jonkman, E. S. P. Branlard, and J. P. Jasa, "Influence of wind turbine design parameters on linearized physics-based models in OpenFAST," *Wind Energy Sci.*, vol. 7, no. 2, pp. 1–27, Mar. 2022.
[102] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.
[103] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021.
[104] Z. Ding, W. Chen, T. Hu, and X. Xu, "Evolutionary double attention-based long short-term memory model for building energy prediction: Case study of a green building," *Appl. Energy*, vol. 288, Apr. 2021, Art. no. 116660.
[105] B. H. Nayef, S. N. H. S. Abdullah, R. Sulaiman, and Z. A. A. Alyasseri, "Optimized leaky ReLU for handwritten Arabic character recognition using convolution neural networks," *Multimedia Tools Appl.*, vol. 81, no. 2, pp. 2065–2094, Jan. 2022.

LUIZA SCAPINELLO AQUINO received the master's degree in electrical engineering from the Federal University of Paraná, in 2022, where she is currently pursuing the Ph.D. degree in electrical engineering. Her research interests include strategies for static and dynamic optimization, bioinformatics, control systems, and renewable energy production systems.



LAIO ORIEL SEMAN received the Ph.D. degree in electrical engineering from the Federal University of Santa Catarina, in 2017.

His research interests include strategies for static and dynamic optimization, along with applications in traffic control, cyber-physical systems, and oil and gas production systems.

VIVIANA COCCO MARIANI received the B.E. degree in mathematics from the Federal University of Santa Maria, in 1993, and the master's degree in computer science and the Ph.D. degree in mechanical engineering from the Federal University of Santa Catarina, in 1997 and 2002, respectively. She is currently a Full Professor with the Department of Electrical Engineering, Federal University of Paraná (UFPR). She has turned her research to the area of regression and classification, with applications in renewable energy problems, such as wind and solar. She is a member of the editorial board of Elsevier's journals.

LEANDRO DOS SANTOS COELHO received the B.Sc. degree in informatics and the B.Sc. in electrical engineering from the Federal University of Santa Maria, in 1994 and 1999, respectively, and the M.Sc. degree in computer science and the Ph.D. degree in electrical engineering from the Federal University of Santa Catarina, in 1997 and 2000, respectively. Currently, he is a Full Professor with the Department of Electrical Engineering, Federal University of Paraná (UFPR). He is a member of the editorial board of Elsevier's journals.

STEFANO FRIZZO STEFENON received the B.E. and M.E. degrees in electrical engineering (power systems) from the Regional University of Blumenau, Brazil, in 2012 and 2015, respectively, and the Ph.D. degree in electrical engineering from the State University of Santa Catarina, Brazil, in 2021.

During the doctoral studies, he developed a research project in deep learning applied to computer vision with the Faculty of Engineering and Applied Science, University of Regina, Canada. His research interests include fault identification in the electrical power systems, condition monitoring and the predictive maintenance of complex industrial systems, and model-based reasoning meets data science.

GABRIEL VILLARRUBIA GONZÁLEZ received the master's degree in intelligent systems, the master's degree in internet security, and the master's degree in information systems management from the University of Salamanca, Spain, in 2012, 2014, and 2015, respectively, and the Ph.D. degree from the Department of Computer Science and Automation, University of Salamanca. He is currently an Associate Professor with the University of Salamanca and a Researcher with the Expert Systems and Applications Laboratory.

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - ROR identifier: 00x0ma614